Optimization of Service Point Locations Report

Prepared for Post&L, N.V
Prepared by Team8, LLC

This report presents the results of a detailed data analysis aimed at optimizing the logistics and delivery services of Post&L, N.V. We have systematically analyzed extensive datasets provided by the client, including demographic data and service point efficiencies, using sophisticated routing algorithms and statistical tools. Our findings deliver crucial insights into areas such as service point allocation, paving the way for enhanced operational efficiency and customer service.

7. June 2024, Maastricht

Table of Contents

Introduction

Our company has been tasked with the job of optimizing Post&L's Maastricht delivery route. Our main goal is to minimize the costs, by finding optimal routes of delivery, and locations for service points.

Optimization is important both for the business and the customers. Smooth running helps the business to keep its costs low and deliver its services in the best way, which results in a more pleasant customer experience. Post&L delivers packages daily, hundreds and thousands of them. Operating the business on the most efficient level with such a huge customer base is therefore of the utmost importance. We as a company would like to help Post&L out by presenting our approach.

We approached the task by first looking at the customer behavior. We tried to model and simulate how each customer interacts with the service point based on demographic data. Then we constructed a map of the existing service points and assigned the squares to them as efficiently as possible using Dijkstrats algorithm. After assigning each square to their nearest service points, we moved on to estimating the daily deliveries and pickups each service point has to do. We did this by training random forest and XGBoost models to predict the demand of these service points and later on implemented simulated annealing to optimize the service point locations. These algorithms will be explained in detail later.

Initial steps

In our initial steps we tried to understand our dataset. The provided data had to be cleaned first and foremost. There were many entries that had missing values, due to low population and privacy reasons, mostly in the demographic information section. We decided to remove any information about the squares that had more than 22-24 missing values. Since these squares had low population it should not affect the predictions. After cleaning the data we constructed initial EDA's such as the following:

Customer Analysis

We constructed a population heatmap to see more densely populated squares. Seeing the population distribution helped us identify squares where the service point would have higher

demands, which means that more packages would be delivered and picked up from, hence resulting in higher costs.

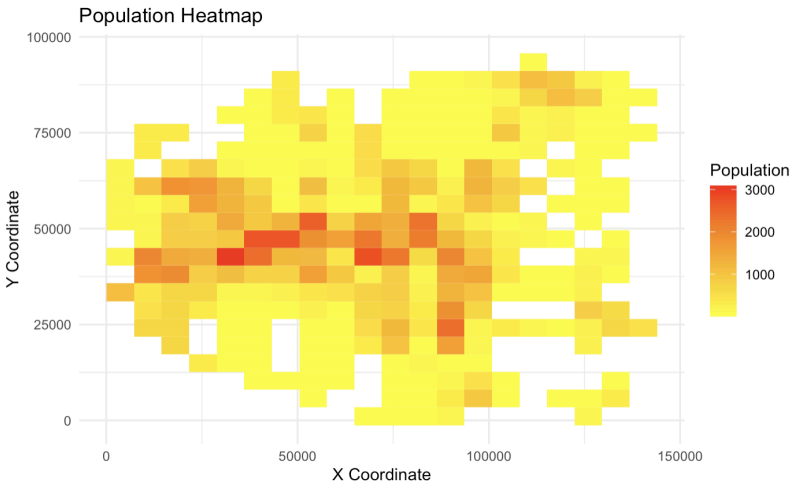

Figure 1: Population heatmap with X and Y coordinates of squares

After identifying the densely populated areas we turned our interests toward the demographic of these areas (Fig. 2). We looked at the age group of the total count of people in the city. We see that the population has the highest Age distribution of between 45-62. We can see that the graph is left skewed hence the population has a higher age on average. With this initial information we deduced that the number of deliveries could be higher than pickups, because higher age can point towards higher financial stability, as well as with higher age people would be less likely to pick up since they don't want to make the trip due to health complications. I would like to point out that these findings are initial and cannot be taken to face value.

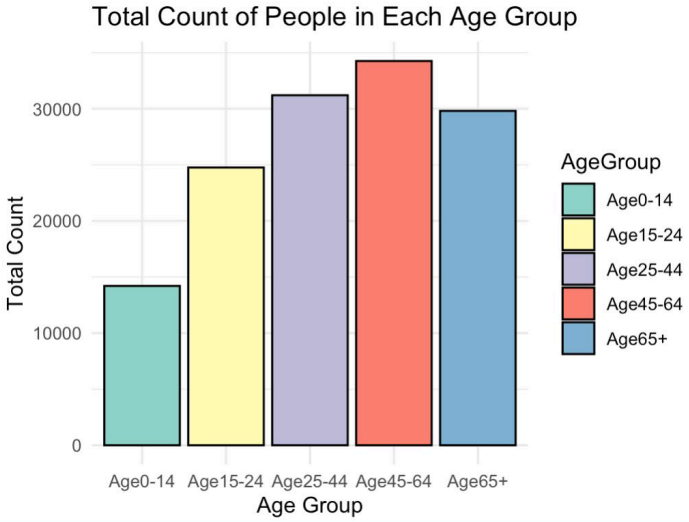

Figure 2: Age distribution of the population in Maastricht

After the age distribution we looked at total deliveries over the map of Maastricht. We entered the locations of each service point and allocated the right number of deliveries to each of them. With this we can see which service points are handling the most deliveries, which can be valuable information for closing the service point or opening an additional one for high demand. We can see from Figure 3 that most of the service points are handling between 30 and 60 thousand deliveries per year, while a couple have less hence they are more likely to be closed down. We can also see that at the top right corner there is a service point that handles more than 90 thousand deliveries, which may be too much and might require opening an additional service point nearby. Overall our initial data exploration and EDA provided us with a better overview of the dataset we got, and from here on we can focus on building an algorithm that will help us with minimizing the costs and rework the service point network.

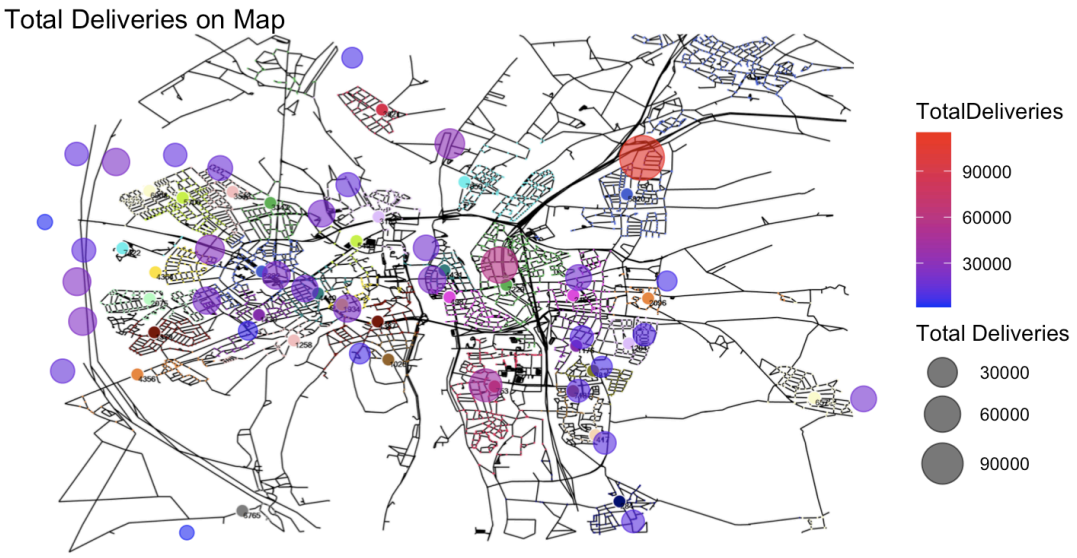

Figure 3: Total deliveries on the map.

Needless to say, findings with regards to the population of Maastricht and client's existing customer base are not restricted solely to optimal network optimization. We believe that our insights and analysis has uncovered rather interesting information which could be utilized by other segments of clients operations such as the marketing department.

Delivery and Pickup Estimation

Dijkstra's Algorithm

In the rapidly evolving landscape of delivery services, optimizing the logistics network is crucial for enhancing operational efficiency and customer satisfaction. Our company aimed to refine the assignment of geographic squares to service points, ensuring that both deliveries and pickups are conducted in the most time-effective manner possible to separate your company from its competitors. We began by employing routing algorithms such as Dijkstra's to calculate the shortest paths within a directed graph framework.

Dijkstra's algorithm which was conceived by a Dutch computer scientist E. W. Dijkstra in 1956. This algorithm finds the shortest path between nodes in a weighted graph which, as in our case, can represent a road network. The structure behind this method is fairly simple as it essentially maintains a set of visited and unvisited vertices. Next, one node, which in our case were service point locations, is selected as a source node and the algorithm finds the shortest path between the source node and all other nodes by using the weights of the edges to find a path that minimizes the total distance between the source node and all other nodes.

We've used the Location ID as the node for service points. The approximate node ID for squares was found based on the similarity of X and Y coordinates between the middle point of a square and all the nodes in the dataset. We've decided that calculating the shortest path in terms of distance would be the most efficient approach as long as we use a directed graph specifying whether a given street is one-way or two-way.

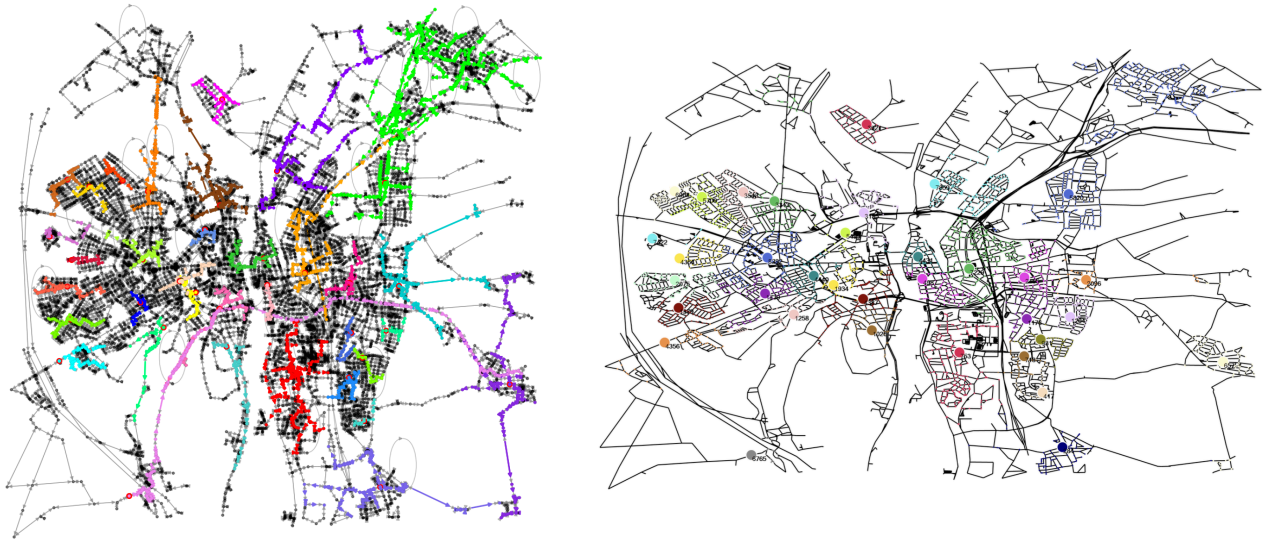

Figure 4: Output of Dijkstrat's algorithm (left)

Figure 4 presents the output of Dijsktra's algorithm. It essentially displays a rough estimate of which service points (marked by red circles) service which areas. This visualization provides a clear, graphical representation of how each service point is optimally connected to various segments of the delivery network within the city. As evidenced, our output is a realistic facsimile of the provided map of Maastricht.

This schema can be used by the logistics department of Post&L in route planning processes to ensure efficiency and timeliness or for resource allocation by showing where more vehicles or resources might be needed based on the concentration of routes and their frequency.

Finally, we've calculated probabilities for pickups and deliveries based on the average distance from the nodes within a square to the assigned service point. If the distance is less than 200 meters, the pickup probability is high (80%), and the delivery probability is low (20%). If the distance is 2000 meters or more, it assumes no pickups will occur (probability of 0% for pickups, 100% for deliveries). For distances in between, the probabilities are interpolated linearly.

$$Pickup\ Probability\ = 0.8 \times \frac{2000 - distance}{1800}$$
$$Delivery\ probability\ = 1 - Pickup\ Probability$$

The formulas above are used to calculate pickup probability (*pickup_prob*) and delivery probability (delivery_prob) based on distance given a customer lives either further than 200 meters or closer than 2000 meters. The pickup probability formula calculates the remaining distance *(2000 - distance)* within the specified range which is then normalized and scaled ensuring that probability linearly decreases from 0.8 to 0 as distance increases from 200 meters to 2000 meters. Delivery probability formula simply ensures that the sum of probabilities equals to 1 at any given distance due to the simple fact that as the pickup probability decreases, the delivery probability increases.

Feature Engineering

Subsequently we've continued with feature engineering. Feature engineering is the process of manipulating and transforming raw data into features that can be used in supervised learning. This step of the analytical process plays a crucial role in enhancing the performance of predictive models by providing them with informative, relevant, and independent variables derived from raw data.

Based on the initial problem description provided by Post&L, we've decided to focus on the following variables: Weighted *mean_age* of a square, *Income_midpoint* of a square, *Average_Distance_to_Service_Point* per square.

Average_Distance_to_Service_Point, for example, calculates the average distance to a service point by looking at the square which was assigned to a given service point, and it finds all the node IDs which are in that square. Next, paths created by Dijktra's algorithms are used to find the distances from each of the nodes to the assigned service point. These distances are then summed up and divided by the total node count of the corresponding square to obtain the average distance by square. Furthermore, we sum up the distances of all squares assigned to a given service point which is later divided by the number of squares.

The weighted variable *mean_age* is calculated for each square using predefined age group midpoints. First, we sum all individuals in different age brackets. The total of each age group's population was then multiplied by its respective midpoint age to ensure that more weight is given to age groups with more individuals. Lastly the total weighted age sum was divided by the total population. Calculating the weighted *mean_age* allows us to take into account the actual distribution of the population across different age groups which results in a more accurate representation.

Variable *Income_Midpoint* adjusts different income brackets to a single midpoint value. The function takes an income range and calculates the average of the lower and upper bounds to estimate the midpoint. Conversion of ranges into a single midpoint value allows us to use data as a continuous variable that is further used in our predictions.

Additionally, historical data was transformed to long format and a feature is_non_work_days was created to identify the days on which service points were open or closed. The two datasets were merged together to make it possible to analyze operational trends over time and assess the impact of non-working days on service delivery.

These methods were chosen to transform raw data into more analyzable formats, enabling deeper insights and more effective decision-making based on demographic, economic, and geographic variables.

Predicting Demand

In order to move forward we had to predict demand for deliveries and pickups. We explored two algorithms, the Random Forest and XGBoost, and later on compared them to see which one produced better results.

Random Forest

We decided to try out the Random Forest for predicting, as it is useful to model feature interactions, and the XGBoost because it is great with complex non-linear relationships and has a high accuracy even if there are missing data or outliers. In the previous paragraph we mentioned these features and now we merged them with the historical data. In order to merge we had to find the averages of these variables. In order to do this we looked at the variables on a service point basis. We identified which squares were assigned to which service points and extracted the values of these squares. After doing this we sum up all these square values of a variable and divide them by the amount of squares the service point services. This preparation was done for both the Random Forest model and the XGBoost.

After training the random forest model we looked at the feature importance for both pickups and deliveries (Fig.5).

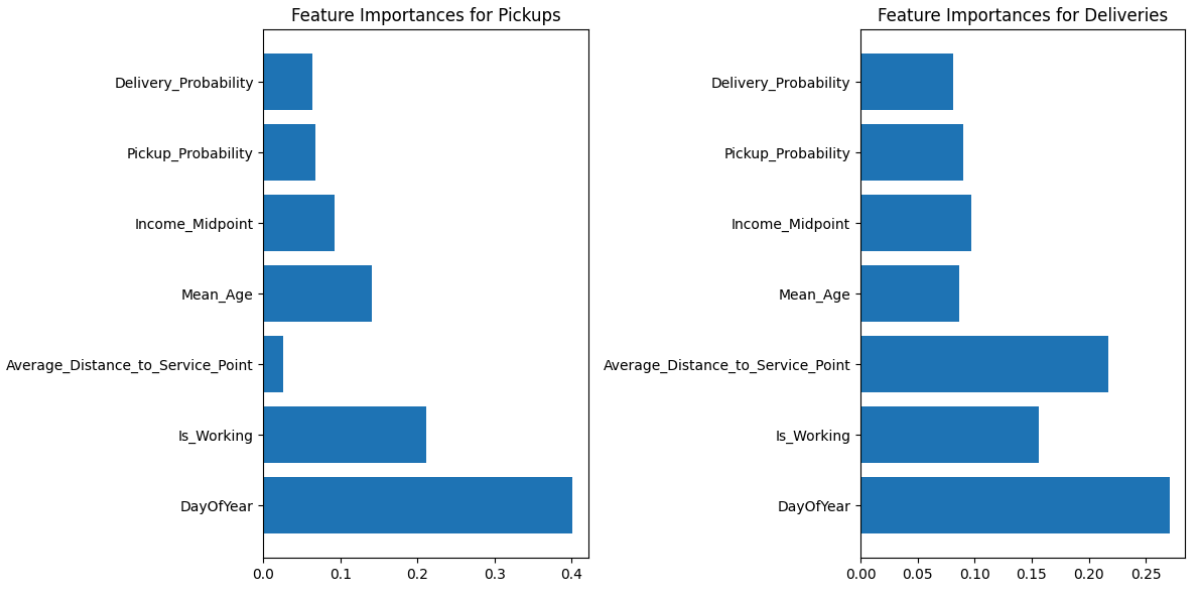

Figure 5: Feature Importance for the Random forest model, both for pickups and deliveries.

We found that the day of the year is the most relevant for both pickups and deliveries. This means that the demand is heavily reliant on the day of the year. Further looking into this metric we identified that around holidays the number of parcels both for pickups and deliveries are higher, which makes sense, as people are preparing for the holiday.

We can also see from the figure that the average distance from the service point feature is one of the dominant features for deliveries, meanwhile it is of the lowest importance for pickups. This can be explained by the fact that customers are considering distance from their house to the service point as one of the main factors when they are choosing whether to pick it up themselves or get it delivered to their house. Customers do not want to travel a long distance from their house to the service point for their package.

The next prominent feature is the Is_Working feature. This represents whether the service point is open or not. This makes sense because if the service point is not open on a particular day when the customer could pick up their package, they might switch to another day and choose delivery, as the customer might not have free time the next available day.

Overall the most prominent features for deciding demand were the Average distance to service point, Is working, and Day Of the year features. These are all important features for the customer and therefore our Random Forest model puts high importance on it too.

The Random Forest model had the following evaluation metrics:

Initial RF Pickups Model Evaluation:	**Initial RF Deliveries Model Evaluation:**
MAE: 15.85983953033268 MSE: 642.5691817221135 R2: 0.7885296146686134	MAE: 18.022352250489238 MSE: 1694.3355083757338 R2: 0.7023348576824919

MAE represents the mean average error, MSE is the mean square error and R2 is the R squared metric. The mean average error metric represents the average magnitude of errors whether it might be negative or positive. This applies to the measure of the unit of the target variable, which in our case would be the demand; the number of parcels picked up and delivered. This means that compared to the prediction of the model ±16 parcels are picked up and ±18 parcels are delivered.

The mean squared error measures discrepancies between the predicted and the actual values. This metric also represents ± parcels but before we can interpret them we need to square root the numbers in order for it to make sense for us. After doing so we can see that the mean squared error suggests about ±25 package difference from the actual values for pickups and ±41 packages for deliveries.

The R squared metric represents the goodness of fit of our model. This gives us an idea of how well unseen cases can be predicted with our model. The higher the R squared value is the more ideal it is. We can see that it is 78% correct for pickups and 70% correct for deliveries. Overall our model provides a good overview of how the demand is in Maastricht and can predict a decently for unseen cases.

XGBoost

Moving onto the other model we tested out: the XGBoost. This model can work with high accuracy and speed. It is ideal for predicting variables, hence we choose to predict demand with it. We prepared the same feature importance diagram and model evaluation for this model too.

As we can see from Figure 6 it is quite different from the Random Forest model. XGBoost does not place that heavy importance for the Day of the year feature. It is still not disregarding it, hence we can conclude that it is somewhat useful in predicting demand.

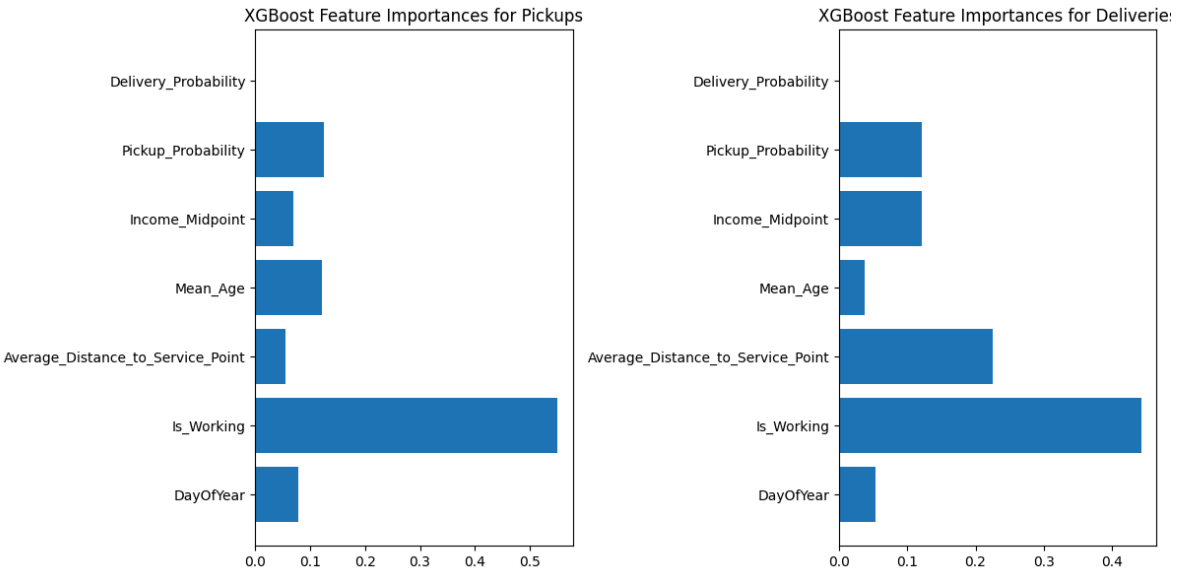

Figure 6: Feature importance diagram for the XGBoost delivery and pickup demands

The most important value for this model was the Is_Working feature. We can see that both for pickups and deliveries it is by far the highest importance. This makes sense, as I explained earlier, whether the service point is open or not. We can also observe that the average distance to service point, similarly to the RandomForest model, plays an important role in deciding for the delivery.

Summarizing the feature importance for XGBoost we can see that Is_Working and Average distance to service point are the most important features that XGBoost pays attention to.

Moving onto the model evaluation we get the following results:

Initial XGBoost Pickups Model Evaluation:	Initial XGBoost Deliveries Model Evaluation:
• MAE: 14.64983856203636 • MSE: 539.9592013301688 • R2: 0.8222987055778503	• MAE: 14.313275583651201 • MSE: 1047.1076980729097 • R2: 0.8160414695739746

We can see that this model produces lower values for MAE and MSE, and produces higher values for the R squared. Since the accuracy of the XGBoost model is higher and the error in prediction is lower, we decided to use this model for predicting demand and use it further for our purpose.

Optimizing Strategy

For our optimization strategy, we decided to use a simulated annealing algorithm to help us find a more optimal layout. We decided to use the algorithm for a few reasons.

Simulated Annealing deals well with complex optimization problems, while being relatively simple and easy to implement, without complex mathematical structures. It allows a large range of flexibility, being able to interplay between the discrete, continuous and combinatorial problems. In our case, it helps combine the problem of layout/location of service point optimization, with optimizing the cost function, which are, respectively, discrete and continuous problems in nature. Lastly, simulated annealing deals well with local optima. It explores a large variety of solutions by allowing itself to periodically accept worse

solutions and through that reduces the risk of getting stuck in a local optima. The presence of local optima becomes more pronounced the more complex the problem and the larger the search space becomes.

We've implemented the Simulated Annealing algorithm in our model as follows.

First we estimate the service point capacities and costs using the pick-up/delivery possibilities we've predicted previously.

It's at this stage there are some assumptions and restrictions we need to make. Firstly, we want to ensure that the city-wide and service point based bounce-back rates of 1% and 2% respectively, are respected. What does this mean? On certain peak days, it might occur that too many packages were ordered for pickup and the capacity has been exceeded. They then need to be sent back to source, and re-sent to the particular service point. As this is a major cause of customer dissatisfaction, we want to limit it so that at maximum 2% of the time, the capacity of each service point is exceeded, and the same for the network as a whole. We do so by imposing confidence intervals when calculating capacities. An interval of 0.98 is imposed on each service points capacity (ensuring that our determined pick-up capacity can handle the load at least 98% of the time) and an interval of 0.99 is imposed on the whole net of service points (ensuring the entire system's determined capacity can handle the entire pick-up workload 99% of the time).

We also make some assumptions regarding distance when calculating our costs.. We base the delivery costs of each service point, on average distances. First, an average distance per square is found, by dividing the sum of distances from each node belonging to the square, from the service point, by the amount of nodes belonging to the square. Secondly, we find a final average distance per service point, by averaging the average distances per square for all of the squares assigned to a service point. This final average distance per service point is used when calculating the predicted costs.

Once the service point capacities and costs have been generated, and the total network cost based on current layout pinpointed, the algorithm proceeds with generating a second, "neighbor" solution. In the new solution, the algorithm either adds a new service point (by picking a random node id from our dataset, excluding the current service point nodes) or removes an existing service point from the network.

The algorithm then recalculates the shortest distances based on the new service point locations, reassigns the squares, calculates the new probabilities, capacities and costs for the new layout and compares the total costs of our new, "neighbor" solution with the cost of our

initial solution. If the new solution has an improved cost, it accepts it and sets it as the new "initial" solution (in essence, our current solution). The algorithm still has a chance to accept a "neighbor" solution, even if its cost is worse than the current.

This is where we use temperature and cooling to optimize the model further. We want the probability of accepting worse solutions to decrease as we further iterate the algorithm. To do so we set an initial temperature, and a cooling rate which decreases the temperature with each iteration of the algorithm. A probability function for accepting worse solutions is then created, using the temperature to ensure the probability of accepting worse solutions decreases. In our model to determine the probability whether the worse solution is accepted we generate a random number between 0 and 1, that has to be lower than $e^{(\text{current cost - neighbor cost}) / \text{temperature}}$, also known as the Metropolis criterion, in order to be accepted. The initial temperature of our model was set to 1000, we applied a cooling rate of 0.035. We iterated the simulated annealing algorithm 130 times, allowing it to make 15 changes per iteration, in order to obtain our results which we will discuss in the following section.

Results and Analysis

As per the historical data we received, the cost of running a service point is estimated to be €75,000 this year. This is the most dominant cost we would have to incur by far compared to delivery cost which comes second and finally the storage cost. Initially, we have 35 operating service centers which alone accounts for a whopping €2.625 million. Followed by almost €500,000 for package delivery and lastly, €100,000 for storage costs. Summing up to nearly €3.2 million, this is our initial value.

On the other hand, with factual analysis, we conclude that we can save a substantial amount of money by closing the majority of current service points and opening new ones in more centralized areas in order to cover the vast majority of Maastricht districts. As stated earlier, we use simulated annealing to find a very feasible solution for the minimization of our total cost in a realistic amount of computing time. The optimal solution of that particular algorithm stated a spectacular total cost of €672,000, with 7 service centers located at the following service points ID: 99, 386, 1030, 4142, 5897, 6171, 8485. Those changes will save us around €2.5 million.

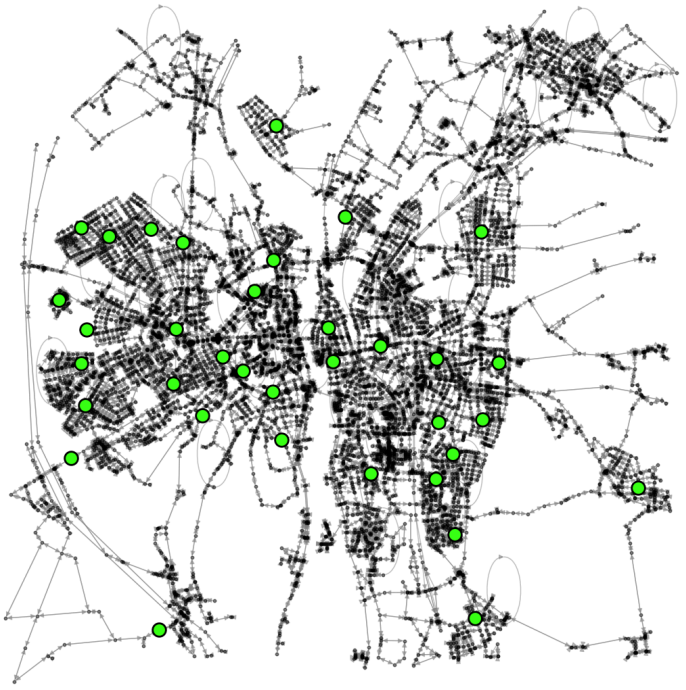

Figure 7: The initial location of service points

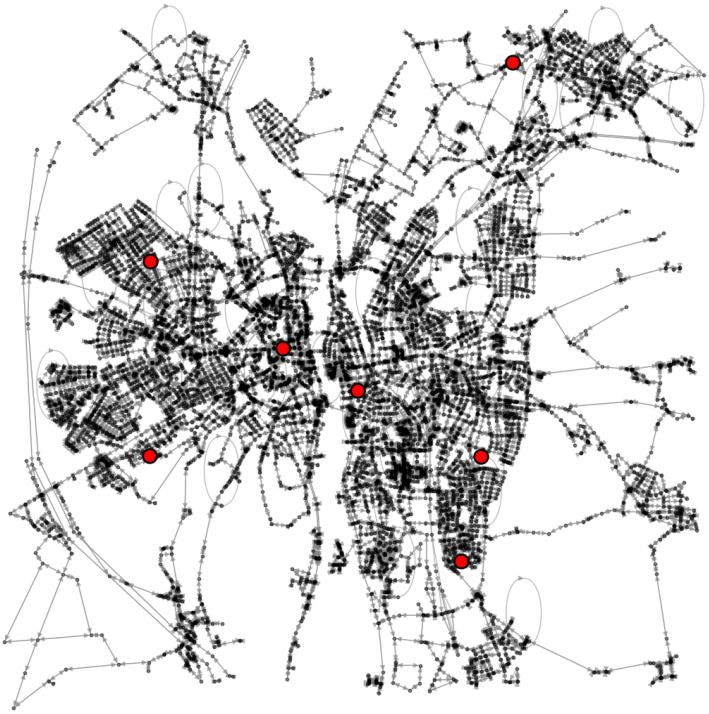

Figure 8: The optimized locations of service points

Diagrams above compare the initial locations against the newly created optimal locations. As stated before, the optimal solution involves building new service point locations in such a manner that all major areas of Maastricht are covered to ensure customer satisfaction while saving more than €2.5 million in costs. Below is the table of the optimal location

While it can be seen as "merely a prediction", our model's accuracy does not share the same opinion as its uncertainty is around 25 packages every day which adds up to, in the context of your business, a negligible uncertainty of 9000 packages being delivered or stored by each location. A rough overestimation of this uncertainty can cost us at most €100,000 more than predicted. Yet, amidst these calculations, we still manage to retain a net positive of €2 million. The optimal locations can be seen to have a good combination of several factors that give a nice logical idea of why they might be optimal. First of all, all of the newly created locations are around 1 km from a supermarket, suggesting why they do not opt for deliveries which is more costly for Post&L than storing the package as people usually go to the supermarket and fetch their parcels on the way. The service location points are a mix of urbanization index areas, which means that they get to cover all deliveries to both the quiet places as well as places with a lot of hustle and bustle.

When optimizing for small changes from our current solution, simply closing service point 6765, for example, will save us around €50,000 with very negligible prediction uncertainty. On the other hand, if we talk about closing and opening service points, but in the end having around 30 service points, the total price dropped to €2.55 million already. Yet all these are so much worse and undesirable from the optimal values stated above.

Limitations and Improvements

Nonetheless, to make an even better optimal solution, we need to know the cost of closing or opening a new service point. We would also need more complete information regarding the areas' data. Lastly, we will need to be provided with more powerful computers in order to obtain the most optimal value in a realistic amount of time. As with the ones we have right now, we can make 1000 combinations of service points in half an hour which is not enough as Post&L will surely be willing to apply our optimisation methods with other areas bigger than Maastricht. It also goes without saying that concrete demand figures would improve the accuracy of our models. Although we were able to predict demand using Xgboost and random forest, the usage of these or other algorithms will never provide us with the exact demand, which at times can be volatile and unpredictable. Using the exact distances from every household in a square to its assigned servicepoint, rather than averaging the distances of all the households would also improve our models accuracy.

Another interesting tool we could look into would be to improve our current real time and traffic data. This could be done by using data that could be provided by the Handhaving (the dutch traffic enforcers), which could include insights into traffic jams, closures, and updated street coverage. Going a step further, we could hire automotive engineers and vehicle experts to deduce which models of which trucks function better regarding overall speed and fuel consumption on certain routes to further elevate our model. For example, for tighter routes in the city center, a smaller truck with a better turning circle would be more ideal whereas in an opposite scenario on wider and longer streets, a fuel saving aerodynamic truck would be more beneficial. It also goes without mention that the transition to greener energy such as electric delivery trucks would save us money while abiding by future Dutch climate control standards and laws.

It is important to note that our findings are not impeccable, but rather a valiant attempt at this problem. Using the data that was provided, we managed to devise a plan that helps us save an excess of 78% of the initial costs, which is no small feat. With excess time and the suggested improvements and ideas mentioned previously, we at team 8 LLC truly believe we can improve that figure to exceed 80 percent at the least.

Conclusion

In conclusion, we strongly suggest having a significant decrease in the number of operating service centers due to the simple fact that operating expenses for running a single service point has a drastic annual cost of €75,000. The service centers should be located in easily accessible locations with an uncomplicated access to the city center whilst remaining a reasonable distance to less urban areas. As stated before, a very feasible solution involves opening 7 service centers located at the following service points ID: 99, 386, 1030, 4142, 5897, 6171, 8485 saving us more than €2.5 million. As well as saving costs, the downsizing from 35 to 7 service points will only prove beneficial to the operations of Post&L, by reducing the need for staff, ultimately minimizing operating costs.

Final Remarks

As we conclude this comprehensive report, we at Team8, LLC would like to extend our gratitude to Post&L, N.V for the opportunity to delve into the intricate dynamics of service point optimization. The progress we have achieved together in this initial phase, despite the constraints of limited resources and time, underscores the potential for significant enhancements with more targeted investments and data enrichment.

Looking ahead, we are enthusiastic about the possibility of deepening our collaboration. We believe that with the strategic allocation of additional resources—specifically more detailed operational data, enhanced computational power, and expert insights into traffic management and vehicle efficiency—we can refine our models to deliver even more robust solutions. This would not only optimize service point configurations across broader regions but also integrate cutting-edge technologies and sustainable practices that align with future industry standards.

We invite Post&L to consider the next steps in our partnership, where we can tackle these challenges together, ensuring that the solutions we develop not only meet but exceed expectations.