



香 港 大 學

**THE UNIVERSITY OF HONG KONG**

**Capstone Final Report**

**SF Express Local Hub Planning Problem by Group Unicorn**

**Advisor: Dr. Jingqi Wang**

**3035818262 Yishu WANG**

**3035819462 Changhang ZUO**

**3035831575 Linxiangyun YANG**

**3035831563 Chonglung CHEUNG**

**3035819917 Yuxin LI**

**3035831599 Yiming LI**

**3035809950 Renyu LIU**

**3035809261 YaLi ZONG**

## Table of Contents

0. Executive Summary.....	1
1. Company/Industry Background Information.....	2
2. Problem the company is facing.....	2
3. The Analytical Challenge.....	2
a. Why the problem cannot be solved by a simple/traditional method.....	2
b. How the problem is related to business analytics/big data/AI.....	3
4. The Analytical Solution.....	4
4.1 The Analytical Solution - Clustering Method.....	4
4.2 The Analytical Solution - Baseline Model.....	7
4.3 Sensitivity Analysis of Baseline Model.....	12
4.3.0 Motivation for Sensitivity analysis.....	12
4.3.1 Sensitivity analysis on demand scale changes.....	12
4.3.2 Sensitivity analysis on unit transportation cost changes.....	16
4.4 Modifications based on the base model.....	18
4.4.1 Two-nodes terminal delivery routes.....	18
4.4.2 Primary Distribution Center.....	20
4.4.3 The usage of Type B Vans.....	21
4.4.4 Combined modifications.....	22
5. The Results.....	23
a. Potential improvement/value gained.....	23
b. Overall Business implications.....	24
c. Comments from the client.....	25
6. Appendix.....	25
a. Technical details.....	25
b. Models and code.....	28
7. Reference.....	30

## **0. Executive Summary**

This report provides analysis and optimization methods of the intra-city delivery problem of SF Express. Methods applied include PAM clustering, optimization, and sensitivity analysis, supported by Python and Gurobi. The main codes, tables and graphs can be found in the appendices. Results of models show that through optimization, the total cost of the whole delivery process can be saved by 7.87% compared with the baseline model.

The key conclusions of the report include:

- Partition around Medoids (PAM) clustering is the most effective cluster method when dividing the customer areas into 20 clusters.
- During the modification process, when primary distribution center, type B van, 2-node-path and combined modification are applied, the total cost is gradually decreasing.
- Sensitivity analysis shows that the model is still valid when the input values fluctuate.

The report also investigates the fact that the analysis conducted has limitations. Some of the limitations include:

- The final results are local optimal instead of global optimal due to the limited capacity of the solver.
- More practical issues like geographical situations can be taken into consideration.

## **1. Company/Industry Background Information**

### **a. Company/industry background**

In recent years, with the booming of Internet e-commerce, the demand for logistics shows an increasingly growing trend. Logistic companies differentiate themselves in different aspects, such as lower prices or higher delivery quality.

### **i. Market position & competitive advantage**

SF Express is positioned in the logistics industry of a highly competitive market. It differentiates itself with reliable, high-speed and efficient distribution, with its own unique service networks.

### **b. Current business operations of the focal business unit**

For a parcel to be delivered from one city to the other, SF express has established intra-city and inter-city service networks. The parcel is firstly sent to a local hub, which is a service point that is

close to the sender's area. Afterwards, when the local hub collects a certain number of parcels, the parcel is sent to a gateway hub, which is generally near the logistic centers of a city, such as an airport and the railway station. When the parcel arrives at the gateway hub of the receiver's city, it will also be delivered to a local hub first, and then be sent to the destination. For transportation, SF Express uses Type A vans, with a capacity of 800 units, to connect local hubs and gateway hubs. Type C vans, with a capacity of 40 units, are used for the transport between customers and local hubs. Type B vans, with a capacity of 200 units, are sometimes applied to connect the local hubs.

## **2. Problem the company is facing**

### **a. The problem**

#### **i. External pain point**

The logistic companies are competing fiercely on price. While ensuring the high efficiency of distribution, SF express is also facing the challenge of deducting costs.

#### **ii. Internal pain point**

Currently, the local hub is planned in terms of the experience, which could not well balance the trade-off between the local hub fixed costs and transportation costs. A large number of local hubs will incur high local hub fixed costs and distribution costs. But if the company cuts a lot of local hubs, the terminal distribution costs between customers and local hubs would be high.

### **b. Desired outcome**

Therefore, SF Express is expecting a reasonable local hub plan that minimizes the total of transportation costs and fixed costs, while satisfying the customers' demand.

## **3. The Analytical Challenge**

### **a. Why the problem cannot be solved by a simple/traditional method**

#### **i. Unstructured problem**

The previous approach of planning local hubs was fact-based and experience-oriented. The company selected the sites of local hubs based on the real environment, considering whether there are venues with fitted size and viable transportation environments in the demand areas. To solve the problem in a more scientific way, algorithms and optimization models could be applied.

## **ii. Computational challenge**

However, the local hub planning problem cannot be directly solved by using an optimization model, since the problem is NP-hard<sup>1</sup>. There are too many demand nodes (2,021) and thus it takes a long time/unlimited time to solve the problem if considering all the nodes at one time and using one single algorithm.

## **b. How the problem is related to business analytics/big data/AI**

While combining with the unsupervised machine learning algorithms of clustering, which splits the whole planning problems into small partitions, the NP-hardness could be tackled. Subsequently, with the establishment of the optimization model that incorporates the data in different aspects, local hubs are planned intelligently with the machine, which could provide more data-based insights for the company to make business decisions.

## **4. The Analytical Solution**

### **4.1 The Analytical Solution - Clustering Method**

#### **a. The data used**

The goal for clustering is to reduce the number of nodes in each cluster and solve the problem in each cluster using the baseline model instead of doing the optimization that finally generates the results. The data available for analysis are the distances between each demand node. PAM Clustering developed by Kaufman and Rousseeuw (2009) is chosen since the distance matrix, instead of the sequence of points, can be used.

#### **b. The analytical methods**

##### **i. Description of the math/conceptual model and assumptions**

PAM Clustering is chosen to do size reduction. PAM stands for “partition around medoids”. It uses a partitional clustering rule, meaning that it breaks up the dataset into groups. With reference to Pyclustering (2021), the clustering algorithm is derived by using its data mining library (the code in the appendix). The clustering rule for PAM clustering, in this case, is to minimize the sum of squares of Euclidean distances between the center of the cluster and points labeled in the cluster.

*1. NP-hardness: In computational complexity theory, NP-hardness (non-deterministic polynomial-time hardness) is the defining property of a class of problems that are informally "at least as hard as the hardest problems in NP".*

The objective function is the following:

$$\min J = \sum_{j=1}^k \times \sum_{i=1}^n \times ||x_i^{(j)} - c_j||^2$$

In the formula above,  $k$  represents the number of clusters,  $n$  is the number of demand nodes,  $x_i^{(j)} - c_j$  shows the distance between each demand node  $x_i^{(j)}$  and its centroid for cluster  $j$ .

In the algorithm, the input variables in the clustering problem are  $k$  (the number of clusters) and the initial list of clusters to contain the cluster consisting of all demand nodes. The output variable is a set of  $k$  clusters. During the process, the algorithm will calculate the distances and classify each demand node into the nearest cluster. The process will repeat and select a new point in each cluster that minimizes the distances. The stop condition is that if the maximum value of distance change of medoids of clusters is less than tolerance (0.0001), the algorithm will stop processing. The clustering method at this stage will only consider the factor of distances between each demand node to do decomposition and make the problem easier to tackle.

## ii. A comparison of the other two clustering methods

K-means clustering uses a partitional clustering rule, meaning that it divides the dataset into groups. k-means clustering rule is based on centroids. In this case, for every point in a cluster, it has a minimized distance with its centroid (the code in the appendix). However, the k-means method is not suitable for clustering customer nodes because the total cost (¥81,298) is much greater than PAM clustering using the baseline model (¥51,144). Outliers have a strong effect on k-means clustering because the centroids are not the existing points and it may increase total distances.

Furthermore, SNN clustering means sharing nearest neighbor clustering. According to He (2014), the clustering rule for SNN clustering is based on the relative distance rule, which measures whether a point is surrounded by its nearest neighbor. We got the neighbor list for all points and updated the clustering center according to the score of occurrence (the code in the appendix). The problem with SNN clustering is that there is a great difference in the number of points within a cluster. It takes a much longer time and its result is similar to the final result derived by PAM Clustering.

### iii. The number of clusters and its justification

In terms of the optimal number of clusters, the goal of using clustering methods is to do size reduction and have a small number of clusters that can be solved by the baseline model that will be discussed later. Traditional clustering evaluation such as the elbow method cannot be applied when determining the number of clusters. If using the elbow method, the number of clustering should be around 6. However, in that case, the number of nodes in each cluster would be too large and cannot be solved by the model since it is NP-hardness. On the other hand, choosing too many clusters, such as 60 clusters, may increase the total costs due to more unnecessary fixed costs and distribution costs. For example, some clusters may not need extra local hubs since they are very close to other clusters in distance. Finally, the number of 20 clusters is chosen first as the balance between optimization and overfitting.

To justify the number of clusters that have been selected, the number of 18-25, 35, 60 clusters are selected and compared through total costs using the baseline model that will be introduced later. As shown in the table below, when choosing 35 and 60 clusters, the total costs are much larger than the costs of choosing 20 clusters. There are more than 1,000 RMB differences every shift, and this is not acceptable because the cost difference would be too much for one year. However, when choosing between 18 and 25 clusters, except for 19 clusters, the cost differences are small and acceptable. Thus, choosing either one in the range would be reasonable. Our team chooses 20 clusters since the cost is the smallest in the trails.

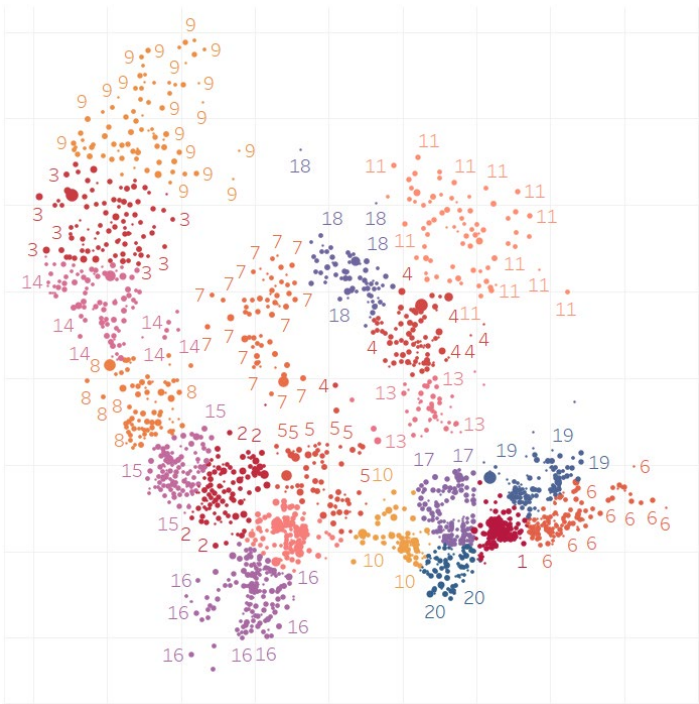
**Table 4.1-1** Cost comparison for different number of clusters based on the baseline model

Number Clusters	of 18	19	20	21	22	23	24	25	35	60
Total cost (RMB)	51173.38	52117.41	<b>51143.80</b>	51309.77	51332.71	51429.89	51234.78	51294.64	52439.75	53171.89

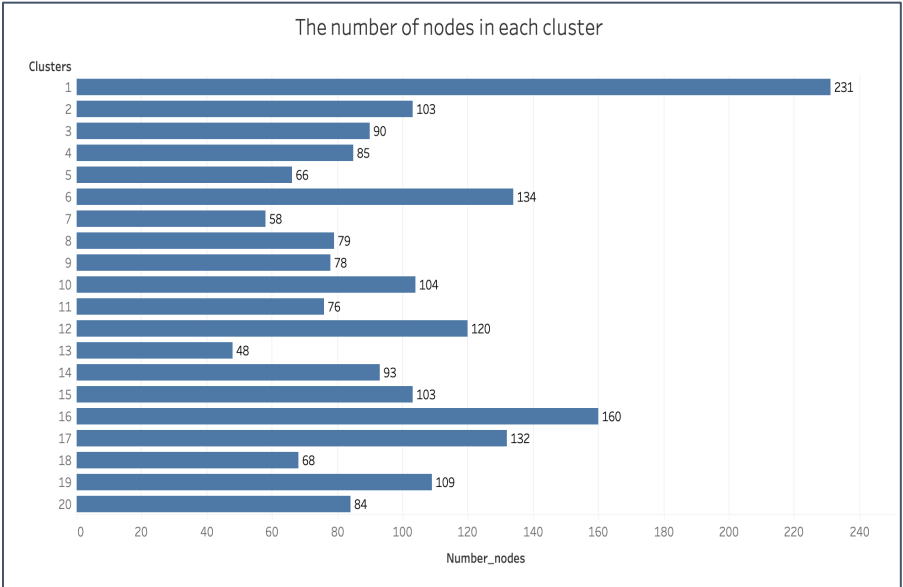
### iv. The result of clustering (tables, graphs)

The graph one below shows the visualization that can tell the clusters our team just mentioned. Take cluster one as an example, it is located on the right corner of the map and there are more demand nodes due to closer distance in the cluster.

After deciding the number of clusters and choosing the PAM algorithm, the distribution of 20 clusters can be derived. As shown in the graph two below, there are 20 clusters with around 50-200 demand nodes in each one. Each cluster has demand nodes with possibly short distances in it based on the PAM Algorithm. Most clusters have the number of demand nodes around or below 100. However, cluster one has more than 200 nodes, since the demand nodes in that cluster are very close to each other in distance.



**Graph 4.1-1** PAM Clustering with 20 clusters



**Graph 4.1-2** The number of demand nodes in each cluster



## **4.2 The Analytical Solution - Baseline Model**

### **a. The data used**

In order to get the minimized cost of delivery, we try to use the input data to build a model with several variables and to get the objective function. Then we consider using a commercial software solver to get the aiming goal.

In the step of optimization, the reason why we choose to use an existing commercial software solver instead of writing the algorithms by ourselves is based on the simplicity of input data, decision variables and calculating logistics. The input data we have are simply the node codes, their relative position and their respective demands of about 2,000 nodes. Among various types of commercial solvers, we choose to use Gurobi to do the optimization under the environment Python. We learned it in the MSBA class and the model we are going to build is relatively simple which could be solved by Gurobi precisely and conveniently.

Besides, we do not have the actual data of the node's position like longitude or latitude, although we try to show the result of the optimization in visualization in the later part with the relative position of all the nodes.

### **b. The analytical methods**

#### **i. Simplification**

By the knowledge and the tools we have at present, the problem we are going to deal with is quite complex and too difficult in reality. Also, our existing data and computer cannot support such a large amount of calculation and analysis. We have to do appropriate simplification to make the question be possible to solve.

Firstly, the great amounts of nodes make it hard for our computers to do the optimization in limited time and resources. We do clustering and divide the 2,021 nodes into 20 clusters, which has been introduced in detail before.

Secondly, to consider these problems in reality, too many variables and elements will make differences for our choice of suitable local hubs. Rent, salary of workers, location, traffic, environment and even political elements will affect our election of the local hubs. We do not know how much effect these elements will make respectively, and the collection of data is complex.

Therefore, at this step, we will first give up considering these elements and mainly focus on the most important elements, the nodes' demands and their relative position. As a result of the simplification, we will not consider other elements and only the nodes' demands and relative position become the final input data.

Thirdly, we have two types of delivery methods, small vans type C and big vans type A. Type C has a load of 40 which will be usually used in the delivery from nodes to local hubs, namely terminal delivery. Type A has a load of 800 which will be usually used in the delivery from local hubs to gateway hubs, namely distribution. Although the demand of each node will usually be small which means the type C could fully fulfill the need, it doesn't mean that we couldn't use type A in terminal delivery. Also, type C could also be used in the distribution of great amounts of goods. Besides, a van could also pass by more than one node and take goods from them. The situation mentioned above will make the optimization become too complex to solve. We will discuss the more complex situations and try to get a better optimization later in the modification part. After all, in the first step of basic optimization, we only consider using type C in terminal delivery and use type A in distribution, as well as every van going between only two nodes.

## ii. Input data

We start the optimization by setting input data. Input data contain four parts.

$N$ : List of nodes

$D_{ij}$ : Distance between node  $i$  and its corresponding hub  $j$

$g_j$ : Distance between hub  $j$  and its corresponding gateway hub SFA

$d_i$ : Customer demand of node  $i$

## iii. Decision Variables

Decision variables also contain four parts. Because in the simplification step we suppose that we will only use type C in terminal delivery and type A in distribution, we only consider the number of type C for nodes and the number of type A for hubs.

$X_{ij} = 1$  {if node  $i$  is assigned to node  $j$ }

$Y_j = 1$  {if node  $i$  is assigned to node  $j$ }

$C_i$ : Number of Type C Vans needed for Node  $i$

$A_j$ : Number of Type A Vans needed for Hub  $j$

#### iv. Constraints

We suppose that each node is assigned to only one local hub in order to simplify the problem. Not every node could be a local hub and only the local hub will assign respective nodes.

**α.** Node  $i$  will only be assigned to node  $j$  if  $j$  is a local hub.

$$x_{ij} \leq y_j \text{ for all } i, j \in N$$

**β.** Each node is only assigned to one local hub.

$$\sum_{j \in N} x_{ij} = 1 \text{ for each node } i \in N$$

Two types of vans should fulfill the respective customer demand for both distribution and terminal delivery. Same as before, type C only serves for normal nodes and type A only serves for local hubs.

**γ.** The number of vehicles can fulfill the demand.

$$40C_i \geq d_i \text{ for each node } i \in N$$

$$800A_j \geq \sum_{i \in N} x_{ij} * d_i \text{ for each hub } j \in N$$

#### v. Cost Decomposition

Total fixed cost equals the number of hubs times the unit fixed cost 20 for each hub.

**α.** Fixed cost

Unit cost: 20

$$F * \sum_{j \in N} y_j, F = 20$$

Total variable cost equals the cost of type A vans, namely the distribution cost, and the cost of type C vans, namely the terminal delivery cost.

**β.** Cost of Type C Vans

Unit cost: 6\*distance

$$\sum x_{ij} * 6D_{ij} * C_i$$

γ. Cost of Type A Vans

Unit cost:  $\text{MAX}[70, 70+4.5*(\text{distance}-5)]$

$$y_j * \text{MAX}[70, 70 + 4.5(g_j - 5)] * A_j$$

$$g'_j = \text{MAX}(5, g_j)$$

$$\sum y_j * [70 + 4.5(g'_j - 5)] * A_j$$

We add them together and finally, we get the objective function showing below. We are aiming to use the commercial software solver to minimize it.

δ. Objective Function

$$\text{minimize } 20 \sum_{j \in N} y_j + \sum_{i,j \in N} x_{ij} * 6D_{ij} * C_i + \sum_{j \in N} y_j * [70 + 4.5(g'_j - 5)] * A_j$$

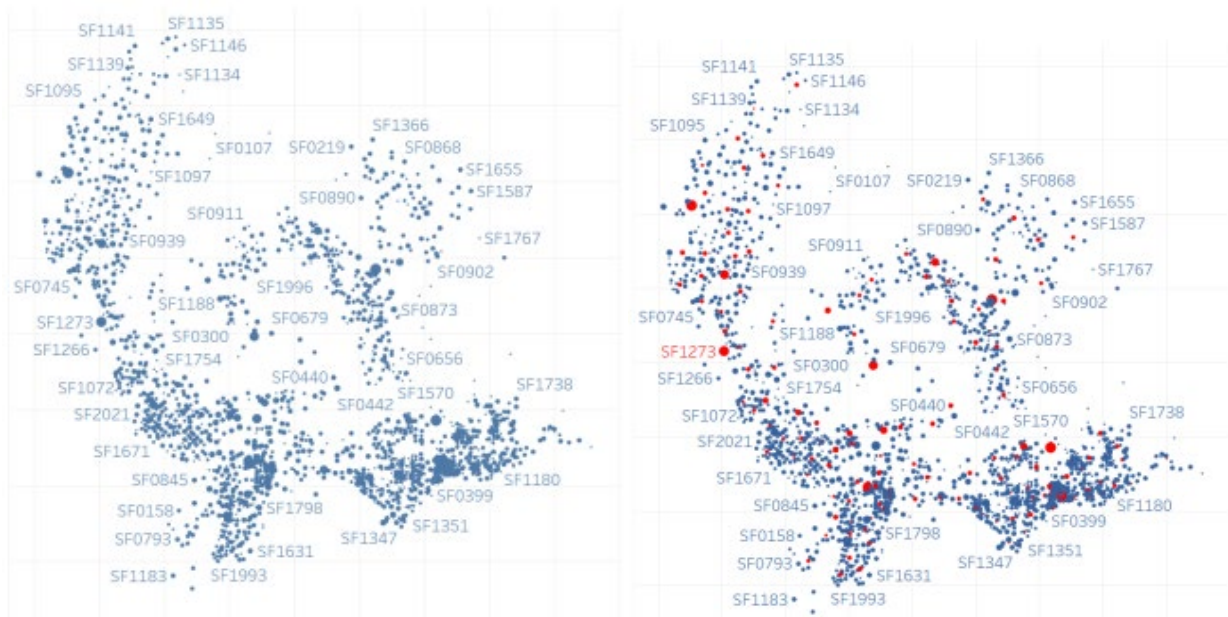
## vi. Optimization Result

Finally, we do the optimization for each cluster and get the result as Graph 4.2.1. The average number of local hubs of every cluster is 7.6 and the average cost of every cluster is 2557.19. Because the clusters haven't been divided into the same size groups, some clusters may have more nodes with more costs and more hubs.

Clusters	Costs	Nodes	Hubs	Clusters	Costs	Nodes	Hubs
1	4685.20	231	14	11	3138.14	76	6
2	2624.67	103	9	12	3134.21	120	11
3	2523.90	90	9	13	1397.52	48	3
4	2654.19	85	7	14	1801.72	93	8
5	2155.87	66	6	15	2006.98	103	8
6	3326.61	134	8	16	3738.28	160	11
7	1727.54	58	5	17	3181.74	132	9
8	1654.53	79	7	18	1715.64	68	5
9	2453.38	78	7	19	2907.47	109	7
10	2341.12	104	7	20	1975.09	84	5

**Graph 4.2-1**

We also try to show the result in visualization. For example, Graph 4.2.2 shows the relative position of every node of all clusters and the red points represent the positions of the elected local hubs.



**Graph 4.2-2**

Graph 4.2.3 shows the details of cluster 7. In the right graph, those big points represent the relative position of 5 local hubs in this cluster. We could find that each local hub is surrounded by several nodes and the nodes' distances to the local hub are also similar. The relative position map shows a reasonable and clear result in visualization.



Graph 4.2-3

### 4.3 Sensitivity Analysis of Baseline Model

#### 4.3.0 Motivation for Sensitivity analysis

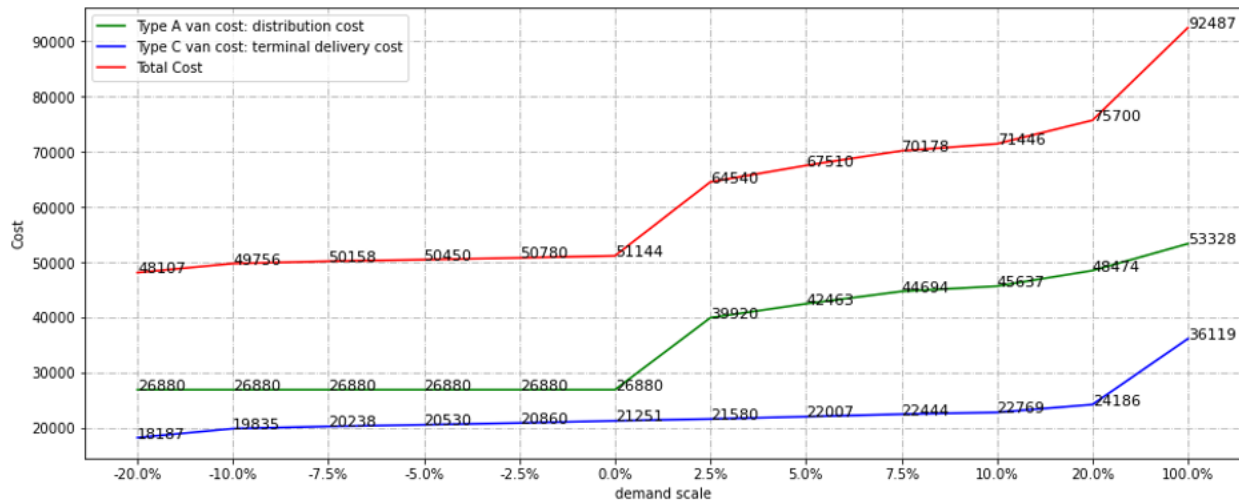
There is only one shift demand data provided by the client, and both the rent cost and unit transportation cost are fixed. However, these factors should fluctuate among shifts. Therefore, the goal of conducting sensitivity analysis is to check if our model is still valid when the input values change. We conduct two sensitivity analyses, focusing on demand and unit transportation cost separately.

##### 4.3.1 Sensitivity analysis on demand scale changes

We changed the demand scale by decreasing and increasing the demand of each customer node by 20%, 10%, 7.5%, 5%, 2.5% respectively.

### a. Cost decomposition

In the beginning, we observe how the cost will respond to the demand changes without re-running the model, and using the 152 originally selected local hubs. The following graph displays the cost decomposition under each demand circumstance.

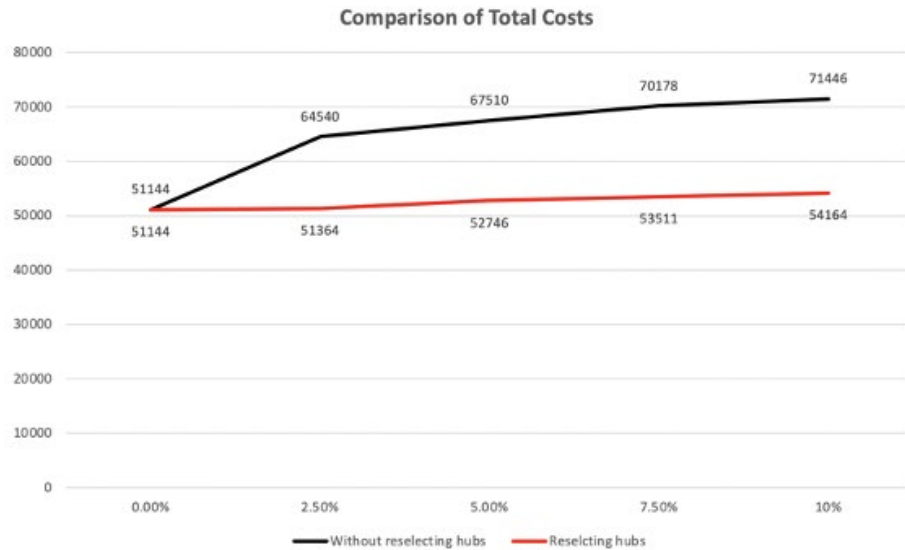


**Graph 4.3-1** Cost decomposition while demand scales changes

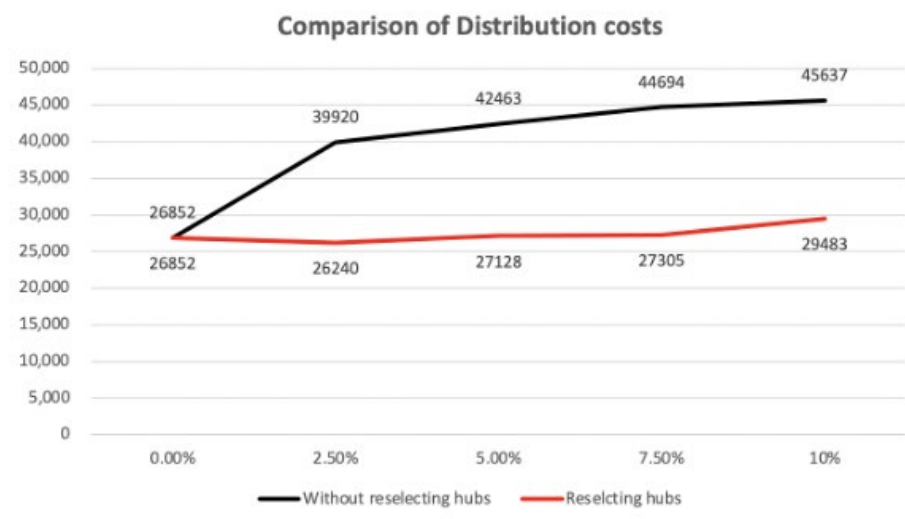
The upper red line represents the total cost. The middle green line represents distribution cost, which is Type A van cost. And the bottom blue line represents terminal delivery cost, which is the Type C van cost.

We can observe that the total cost increases rapidly when the demand increases. Given that we do not reselect the hubs, when the demand increases 2.5%, the total cost could even increase by 26%. And the total cost line is pretty much parallel to the distribution cost, which is Type A van cost. This is mostly because when the demand increases, once the hub demand exceeds the former multiplier of 800, which is the capacity of type A van, the model needs to assign one more type A van, even if there are just several parcels in this van.

Since the total cost increments can be mostly attributed to distribution cost increase, we compare the above cost decomposition with the costs after running our model and reselecting the hubs. The following graphs display the total cost comparison between reselecting the hubs and without reselecting the hubs.



**Graph 4.3-2 Comparison of Total Cost**



**Graph 4.3-3 Comparison of Distribution Cost**

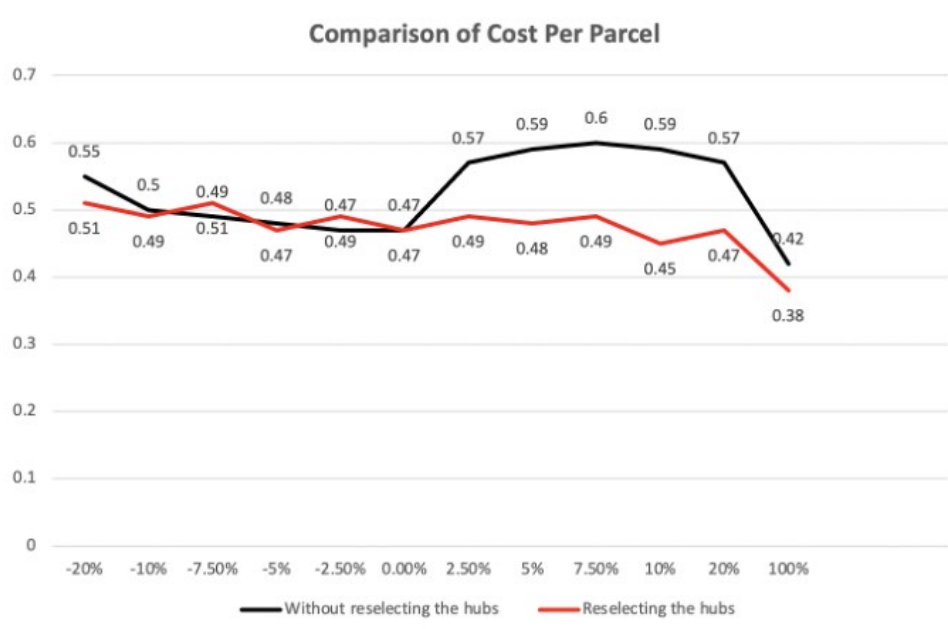
We can observe that in terms of both total cost and distribution cost, re-running the model and reselecting the hubs help to reduce the cost increments, which manifests that our model is valid regarding the demand changes.

And the practical implication might be when there is seasonal demand fluctuation, especially when there is an expected demand increase, the SF Express could try short-term contracts with potential hubs to reduce costs increments.



## b. Cost per parcel

Apart from the model performance regarding the total cost, we also check how the cost per parcel will change as the demand amount changes. And the following chart displays how the cost per parcel changes as the demand scale changes when the hubs are reselected and when keeping the original 152 hubs.



**Graph 4.3-4** Comparison of Cost Per Parcel

After applying the model, the unit cost becomes lower when demand increases. And the highest cost occurs when the demand decreases by around 7.5%, which might be a demand situation worth paying attention to for the SF Express.

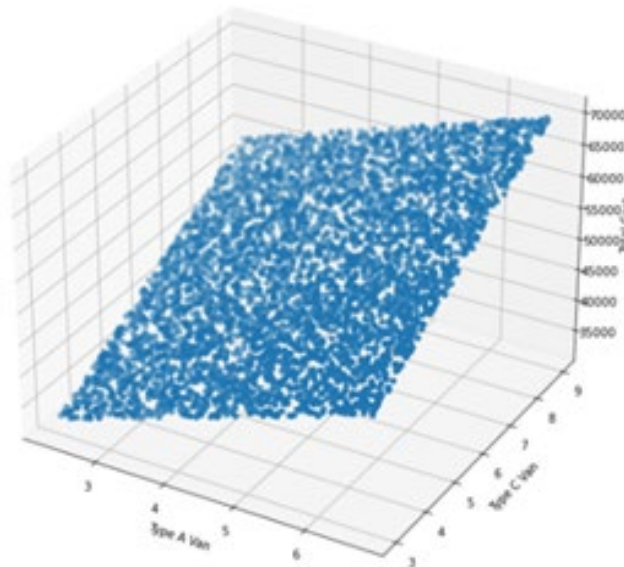
Beyond subtle and modest demand changes, we also consider the situation of demand surging, for example, during e-commerce campaigns. We found that the number of parcels was doubled during the double 11 campaign last year for SF express. So, we also check how the model performs in this demand surging circumstance. We found that reselecting the hubs could help to reduce the cost per parcel from 0.42 to 0.38. The implication here is that, the SF Express should consider improving capacity for demand increase and demand surging, as in these situations, the SF Express can take advantage of economies of scale to reduce unit cost. And the SF Express could also take short-term hub renting contracts into consideration under demand surging situations.

#### 4.3.2 Sensitivity analysis on unit transportation cost changes

After exploring the cost on different demand scale, we were also concerned about the sensitivity and marginal transportation cost of the current hub distribution plan when the unit cost of Type A and Type C van operations fluctuated according to traffic conditions. So we generated a series of alternative prices, varying from 50% to 150%, to study the relation between total cost and mathematical effect of changes of these factors. And we gain three conclusions from this process. Firstly, there is a strong linear trend between total cost and unit transportation cost. Secondly, the unit cost of type C vans is more influential than that of type A vans. Thirdly, if an unexpectable cost fluctuation occurred, our model is still valid for hub-selection optimization.

##### a. Perfectly Linear relation between factors and response

Initially, we compute the cost only based on a current distribution plan, the 3D scatter plot displays the simple relation between vans unit cost and total cost, the X-axis is the unit cost of type A van, the y-axis is the unit cost of type C van, Z-axis is the total cost. The X-axis ranges from 2.25 to 6.75 and the y-axis ranges from 3 to 9. By generating a series of random values (around 5000 data points) as input, the total costs of different input combinations can be computed and marked on the plot.



**Graph 4.3-5** Unit Transportation Cost's Effect on Total Cost

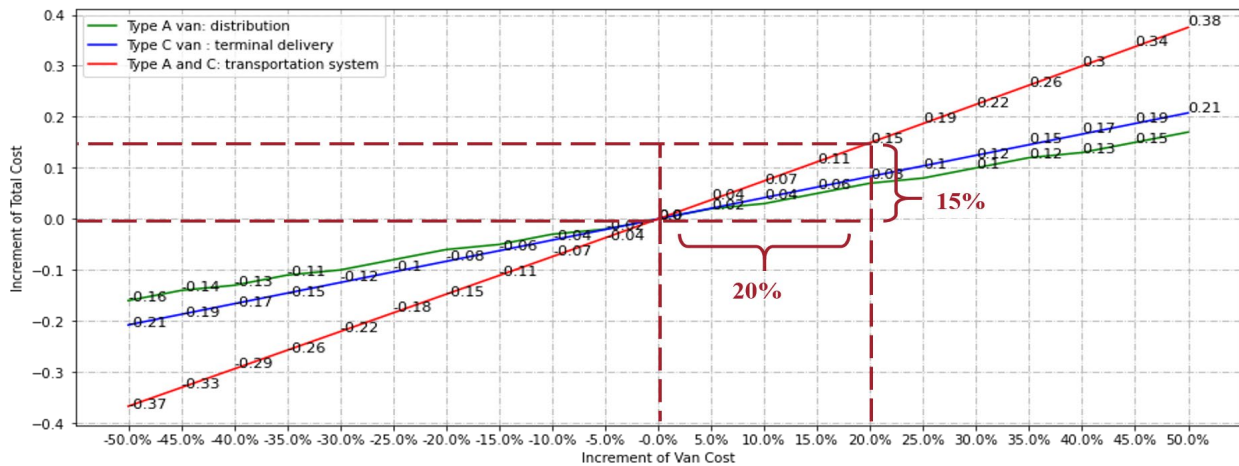
In the picture, the data point consists to a slanted plane. Total cost soars at a stable speed as the unit cost of type A vans and type C vans increase. And each unit's increment in type C and type A vans will give rise to a total cost increase by around 3,000 and 4,000 respectively.

#### b. The current model is more sensitive to changes in the unit cost of type C van

In addition to checking the effect of factors having on response when absolute values change, we also manually increase the prices from -50% to +50% with 5% step length and compute each result. In the line chart, the red line represents type A and type C van, the blue line represents type c van and the green line represents type A van. From the chart describing features and response, we can draw the conclusion that the relation between the total cost and unit transportation costs (type A, type C and type A&C) is perfectly linear. This is the first conclusion that the line chart tells us along with the 3D scatter plot. This Linear trend is determined by the linear objective function we built in the previous optimization step:

$$20 * \sum_{j \in N} y_j + \sum_{i,j \in N} x_{ij} * 6 * D_{ij} * C_i + \sum_{j \in N} y_j * [70 + 4.5 * (g_j - 5)] * A_j$$

Furthermore, if the unit cost of type A vans or type C vans change by 1%, we could expect 0.32% and 0.42% increases in total cost respectively, which indicates that our current model is more sensitive to changes in the unit cost of type C van. We can also infer this conclusion by comparing the slopes of lines in the picture.



Graph 4.3-6 Relationship Between Unit Transportation Cost and Total Cost

### c. Variance-based sensitivity indexes

Variance-based sensitivity analysis is a form of global sensitivity analysis. Working within a probabilistic framework, it decomposes the variance of the output of the model or system into fractions that can be attributed to inputs or sets of inputs.

In this project, we used this method to calculate the sensitivity indexes of each variable (unit cost of vans). The formula of sensitivity index  $S_i$  is:

$$S_i = \frac{V_i}{Var(Y)}$$

Where  $V_i = Var_{x_i}(E_{xi}(Y|x_i))$ ,  $i$  represents the order of the variable.

By generating 1000 random samples, the sensitivity index of two factors and their 95% confidence intervals are shown in the result table. The sensitivity index of the unit cost of type C vans is larger than that of type A vans, which means that changes in the unit cost of type C vans are more influential to the total budget compared with others. This conclusion can match with the one we gain from the line chart.

**Table 4.3-1** Sensitivity Indexes of Variables

Name	sensitivity index	S_conf (95%)	Total sensitivity index	ST_conf (95%)
cost_a	0.366	0.048	0.369	0.032
cost_c	0.629	0.067	0.631	0.052

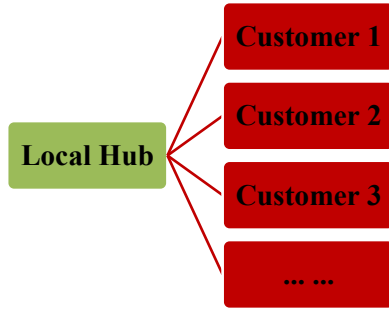
## 4.4 Modifications based on the base model

As the base model is under a simplified condition, three modifications are applied to the optimization model to make the result more practical as well as deducting the total cost.

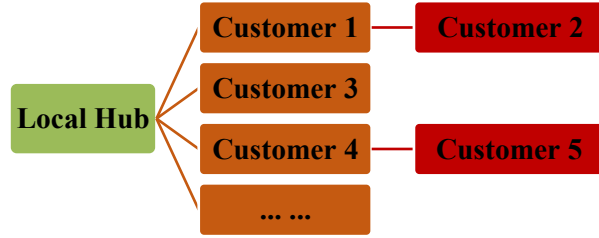
### 4.4.1 Two-nodes terminal delivery routes

The first modification focuses on the terminal delivery part, where parcels are delivered from local nodes to customer areas. The base model supposes that all the parcels are delivered directly from local hubs to customer areas, which does not involve routing problems. However, in real business, a few customer areas (especially those with low demands) may be visited on the same route in sequence to reduce the total travel distance. The differences in delivery routes are shown in the

following graphs.



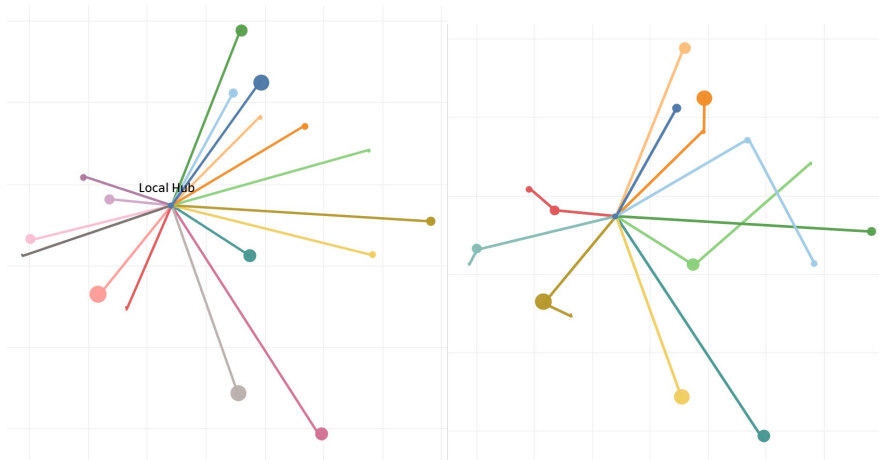
**Graph 4.4-1** Base Model



**Graph 4.4-2** Two-nodes Terminal Delivery Routes

Due to the constraints of operation capacity, this modification is implemented based on the local hub groups of the base model optimization result. A new optimization model which minimizes the total delivery cost is built for this modification. The model is a MIP that shares similar characteristics with the base model. The main difference is that the optimization process is performed on a local hub level. Also, the number of customer nodes on a route is limited to be no more than 2 for timeliness requirements to guarantee service quality. Please refer to the Appendix for the details of the optimization model.

The optimization model is performed for each of the 152 local hub groups. We take Hub SF1564 as an example. 6 of the 17 customer areas are arranged to be a second destination on a route, most of them are close to the first destination, except for the 2 on the east, which are arranged in this way to avoid using an extra van.



**Graph 4.4-3** Base Model Result of SF1564

**Graph 4.4-4** Two-nodes route result of SF1564

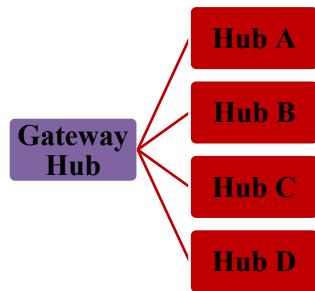
By reducing the total travel distance and the number of type C vans in usage, this modification improved the terminal delivery cost by 9.55% and the total cost by 3.97%.

**Table 4.4-1** Cost Comparison of 2-nodes routes modification

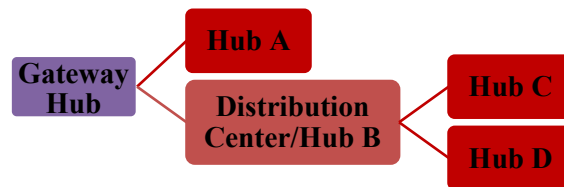
	<b>Base Model</b>	<b>2-nodes Route</b>	<b>Improvement</b>
Terminal Delivery Cost	21,251	19,222	9.55%
Total Cost	51,144	49,115	3.97%

#### 4.4.2 Primary Distribution Center

The second modification focuses on the distribution segment – delivery from gateway hubs to local hubs. Parcels may not be delivered directly from gateway hubs but are allowed to be delivered to a few primary distribution centers first and then distributed to their subordinate local hubs. Compared to a secondary local hub, a primary distribution center may have larger space and more labor. The differences are shown in the following graphs.



**Graph 4.4-5** Base Model



**Graph 4.4-6** Primary Distribution Centers

Since the fixed cost and terminal delivery cost remain the same, we build a new MIP model which minimizes the transportation cost between Gateway hub and local hubs. The optimization is performed based on the 152 local hubs chosen by the base model. What is more, clustering is unnecessary as the number of hubs is limited, so hubs from different clusters are allowed to be assigned to the same distribution center. Please refer to the Appendix for the details of the optimization model.

The optimization process chose 6 local hubs to be distribution centers and 12 local hubs to be their subordinate hubs. Other local hubs remain direct delivery from Gateway hub. This modification decreased the distribution cost by 1.1% and total cost by 0.6%, not as much as the modification on

routes, the reason may be that the base model result has already tried to make the best use of vehicle capacity (800), making it not always necessary to build a higher-level distribution center.

**Table 4.4-2** Cost Comparison of primary distribution center modification

	<b>Base Model</b>	<b>Distribution Center</b>	<b>Improvement</b>
Distribution Cost	26,852	26,560	1.09%
Total Cost	51,144	50,852	0.57%

#### 4.4.3 The usage of Type B Vans

The third modification is the usage of Type B vans. With a capacity of 200 pieces and a unit cost of ¥4/km, which are in the middle of Type A and C vans, Type B vans may perform better for medium demand deliveries. In the base model, only one type of vans is allowed for each kind of delivery. This modification allowed the usage of Type B vans for both distribution and terminal delivery.



The optimization model is a MIP that is quite similar to the base model, which minimizes the total fixed cost, distribution costs and terminal delivery costs. The variables of the number of Type B vans and relative costs in the objective function are added. For the constraints, the total capacity of 2 kinds of vans can fulfill the demand of each hub or node. Please refer to the Appendix for the details of the optimization model.

Initialization is crucial for operation efficiency. The model could only return a feasible solution in an acceptable period when initialized with the result of the base model as a warm start. The result shows that nearly 15% of routes are using Type B vans. This modification improved all of the 3 components of cost, and the total cost has decreased by 3.2%. The terminal delivery segment has contributed most to cost-saving.

**Table 4.4-3** Cost Comparison of using type B vans modification

	Base Model	Using Type B Vans	Improvement
Fixed Cost	3,040	3,020	0.66%
Gateway - Hub	26,852	26,550	1.13%
Hub - Customer	21,251	19,942	6.16%
Total Cost	51,144	49,511	3.19%

#### 4.4.4 Combined modifications

After performing the 3 modifications independently, a combined modification is conducted. Based on the optimization result of using type B vans, we implemented the modifications of 2 nodes terminal delivery routes and primary distribution centers, both allowing the usage of Type B vans.

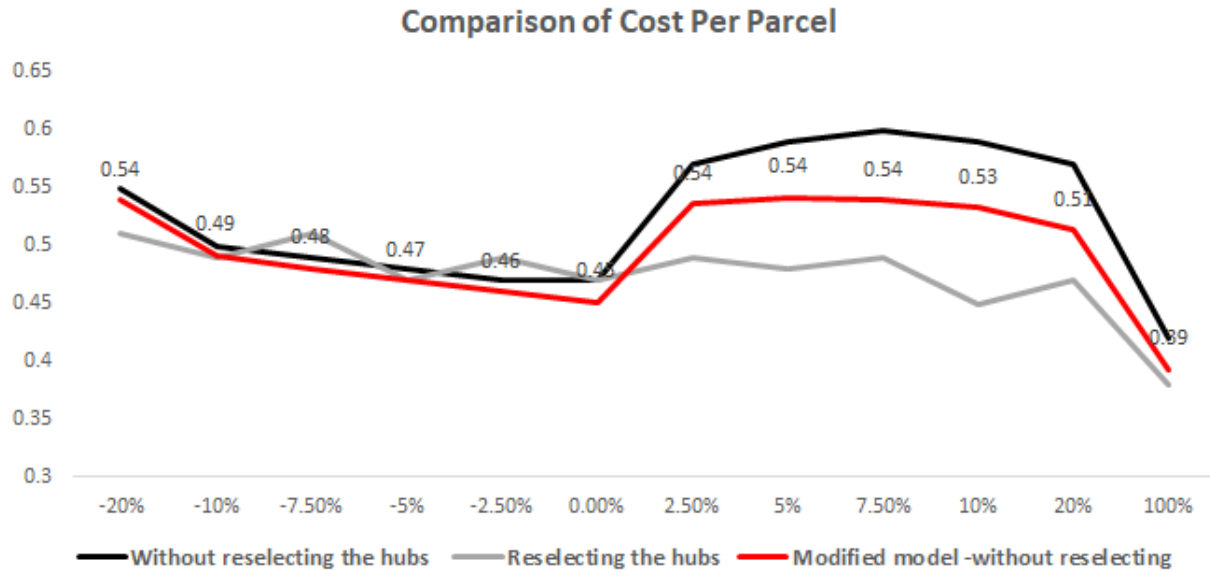
The final result has a total cost of ¥47,118, which is 4.8% lower than the Type B Van model and 7.9% lower than the baseline model.

**Table 4.4-4** Cost Comparison of combined modifications

	Base Model	Using Type B Vans	Combined Modifications	Improvement
Fixed Cost	3,040	3,020	3,020	
Gateway - Hub	26,852	26,550	26,267	1.07%
Hub - Customer	21,251	19,942	17,831	10.59%
Total Cost	51,144	49,511	47,118	4.84%

We also compared the cost per parcel without re-selecting the hubs when demand changes. As the baseline model, it caused a jump in unit cost facing small demand increments, but to a lower degree compared to the base model and it became steady much earlier.



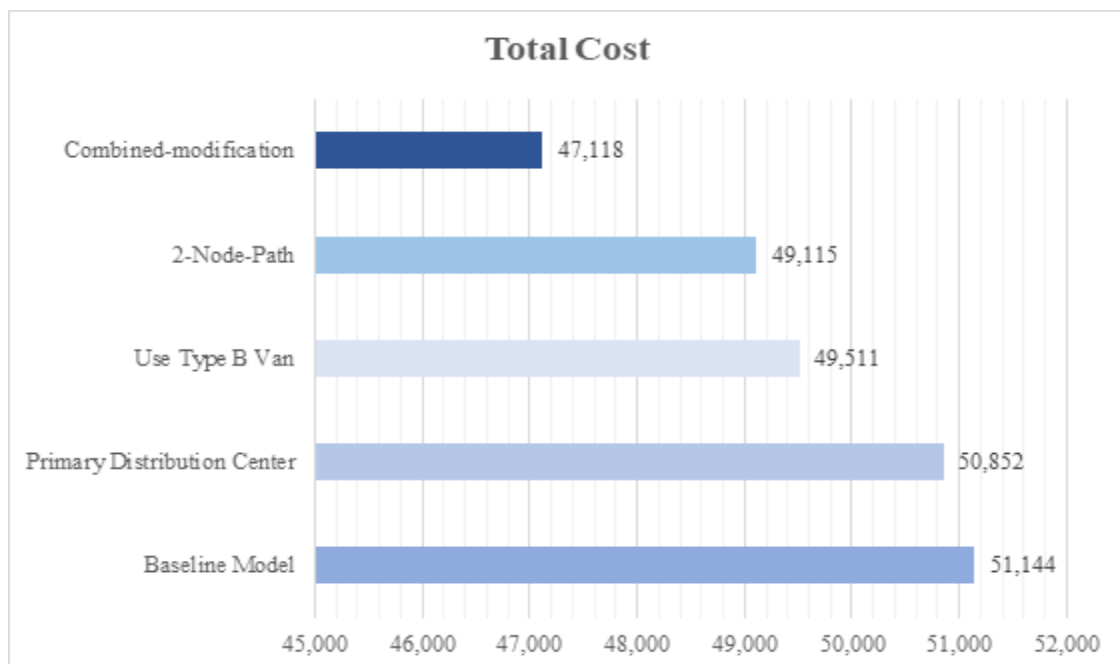


Graph 4.4-7 Comparison of Cost Per Parcel

## 5. The Results

### a. Potential improvement/value gained

In conclusion, we compared total costs using different modification methods. The corresponding total costs are as follow:



Graph 5.1

**Table 5.1**

	<b>Direct</b>	<b>Baseline Model</b>	<b>Primary Distribution Center</b>	<b>Use Type B Van</b>	<b>2-Node-Path</b>	<b>Combined modification</b>
<b>Total Cost</b>	676,309	51,144	50,852	49,511	49,115	47,118
<b>Proportion of Direct Cost</b>	-	100.00%	99.43%	96.81%	96.03%	92.13%

As shown in Graph 5.1 above, the total cost is gradually decreasing during the modification process. In the beginning, we clustered the customer areas and designed a baseline model, and the total cost is ¥51,144. Then the primary distribution center was added, reducing the total cost to ¥50,852. As type B van and 2-node-path are used, total costs declined to ¥49,511 and ¥49,115 respectively. When we used the combined-modification method, the total cost reached ¥47,118 at last.

On the first column of Table 5.1, without clustering, which means parcels are directly transported from customer areas to SFA, and there is no fixed cost. The original direct cost is ¥676,309, more than ten times the baseline model cost. But with clustering, the total cost is greatly reduced to ¥47,118 at most using the combined modification method, 92.13% of the original baseline model cost.

#### **b. Overall Business implications**

With the optimized local hub plan, SF Express could achieve a large deduction of local hub fixed costs. Combined with further modifications on the routing and distribution methods, the transportation costs could also be reduced. Compared with the original experience-based plan, the optimized planning of local hubs is more intelligent and cost-efficient. However, as we applied the optimization model on each of the clusters, the result we had is only local optimum. To further improve this, we may check the demand nodes on the edge of each cluster. This may help to see whether there are demand areas to be re-assigned. Alternatively, if there are solvers with a larger capacity to solve the problems, we could also apply the optimization model on the whole data set to obtain the globally optimal result. In addition, we may adjust our model to capture more real

issues or information, such as the demand fluctuations, cost variations, rent or other practical information.

### c. Comments from the client

SF Express is overall satisfied with the results. The thought and method of doing clustering before the optimization is creative and helpful. For future research, we may consider improving the clustering algorithm by incorporating the factor of demands and focus more on designing the routes of distributions.

## 6. Appendix

### a. Technical details

#### 1) Optimization model for 2-nodes terminal delivery routes

##### Parameters

$N$	List of nodes (customer areas) within a local hub group
$D_{ij}$	Distance between node $i$ and node $j$
$g_j$	Distance between node $j$ and local hub
$d_i$	Customer demand of node $i$

##### Decision Variables

$x_{ij}$	$\mathbb{1}$ {if node $i$ is on the same route with node $j$ }
$y_j$	$\mathbb{1}$ {if node $j$ is the first node on a route}
$C1_j$	Number of Type C Vans needed between local hub and first node $j$
$C2_i$	Number of Type C Vans needed between first node and second node $i$

##### Objective Function

$$\text{Minimize Terminal Delivery Cost} = \sum_{i,j \in N} x_{ij} \cdot 6D_{ij} \cdot C2_i + \sum_{j \in N} y_j \cdot 6g_j \cdot C1_j \quad (1)$$

S.t.

$$x_{ij} \leq y_j \quad \forall i, j \in N \quad (2)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N \quad (3)$$

$$\sum_{i \in N} x_{ij} \leq 2 \quad \forall j \in N \quad (4)$$

$$40C1_j \geq \sum_{i \in N} x_{ij} \cdot d_i \quad \forall j \in N \quad (5)$$

$$40C2_i \geq d_i \quad \forall i \in N \quad (6)$$

The objective function (1) minimizes the total terminal delivery cost between a local hub and its subordinate customer areas. Constraints (2) and (3) ensures the assignments and constraints (5) and (6) ensures the fulfillment of demand. Constraint (4) restricts the number of customer nodes on a route to be no more than 2 for timeliness requirements to guarantee service quality.

## 2) Optimization model for primary distribution center

### Parameters

$N$  List of local hubs

$D_{ij}$  Distance between hub  $i$  and hub  $j$

$$D'_{ij} = \begin{cases} \max(5, D_{ij}), & i \neq j \\ 5 - \frac{70}{4.5}, & i = j \end{cases}$$

$g_j$  Distance between hub  $j$  and gateway hub

$$g'_j = \max(5, g_j)$$

$d_i$  Total customer demand of local hub  $i$  and its customer areas

### Decision Variables

$x_{ij}$   $\mathbb{1}$  {if hub  $i$  is subordinate to distribution center  $j$ }

$y_j$   $\mathbb{1}$  {if hub  $j$  is a primary distribution center}

$A1_j$  Number of Type A Vans needed between local hub and first node  $j$

$A2_i$  Number of Type C Vans needed between first node and second node  $i$

### Objective Function

Minimize Distribution Cost =

$$\sum_{i,j \in N} x_{ij} \cdot (70 + 4.5(D'_{ij} - 5)) \cdot A2_i + \sum_{j \in N} y_j \cdot (70 + 4.5(g'_j - 5)) \cdot A1_j \quad (7)$$

S.t.

$$x_{ij} \leq y_j \quad \forall i, j \in N \quad (8)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N \quad (9)$$

$$800A1_j \geq \sum_{i \in N} x_{ij} \cdot d_i \quad \forall j \in N \quad (10)$$

$$800A2_i \geq d_i \quad \forall i \in N \quad (11)$$

The objective function (7) minimizes the total distribution cost between the gateway hub and all the local hubs. Constraints (8) and (9) ensures the assignments and constraints (10) and (11) ensures the fulfillment of demand using type A vans.

### 3) Optimization model for using type B vans

#### Parameters

$N$  List of nodes

$D_{ij}$  Distance between node  $i$  and hub  $j$

$$D'_{ij} = \begin{cases} \max(5, D_{ij}), & i \neq j \\ 5 - \frac{30}{4}, & i = j \end{cases}$$

$g_j$  Distance between hub  $j$  and gateway hub SFA

$$g'_j = \max(5, g_j)$$

$d_i$  Customer demand of node  $i$

#### Decision Variables

$x_{ij}$   $\mathbb{1}$  {if node  $i$  is assigned to hub  $j$ }

$y_j$   $\mathbb{1}$  {if node  $j$  is a local hub}

$A_j$  Number of Type A Vans needed between local hub  $j$  and gateway hub

$B1_j$  Number of Type B Vans needed between local hub  $j$  and gateway hub

$C_i$  Number of Type C Vans needed between node  $i$  and local hub

$B2_i$  Number of Type B Vans needed between node  $i$  and local hub

#### Objective Function

*Minimize* Total Cost =

$$20 \sum_{j \in N} y_j + \sum_{i,j \in N} x_{ij} \cdot 6D_{ij} \cdot C_i + \sum_{i,j \in N} x_{ij} \cdot (30 + 4(D'_{ij} - 5)) \cdot B1_i + \sum_{j \in N} y_j \cdot (70 + 4.5(g'_j - 5)) \cdot A_j + \sum_{j \in N} y_j \cdot (30 + 4(g'_j - 5)) \cdot B2_j \quad (12)$$

S.t.

$$x_{ij} \leq y_j \quad \forall i, j \in N \quad (13)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N \quad (14)$$

$$800A_j + 200B2_j \geq \sum_{i \in N} x_{ij} \cdot d_i \quad \forall j \in N \quad (15)$$

$$40C_i + 200B1_i \geq d_i \quad \forall i \in N \quad (16)$$

## b. Models and code

### i. PAM Clustering

```

1 # K-Medoids algorithm - PAM
2 # Running about 30 mins
3 from pyclustering.cluster.kmedoids import kmedoids
4 from pyclustering.cluster.center_initializer import kmeans_plusplus_initializer
5 from pyclustering.cluster import cluster_visualizer
6 from pyclustering.samples.definitions import FCPS_SAMPLES
7 from pyclustering.utils import read_sample, calculate_distance_matrix
8
9
10 initial_medoids = [102] + [101] * 19
11
12 # create K-Medoids algorithm for processing distance matrix instead of points
13 kmedoids_instance = kmedoids(matrix, initial_medoids, data_type='distance_matrix')
14
15 # run cluster analysis and obtain results
16 kmedoids_instance.process()
17
18 clusters = kmedoids_instance.get_clusters()
19 medoids = kmedoids_instance.get_medoids()

```

## ii. K-Means Clustering

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 loaction_info = pd.read_excel('./Coding and data/loaction_info.xlsx')
6 loaction_info = loaction_info[["customer_code", "longitude", "latitude"]]
7
8 from pyproj import Transformer
9 transformer = Transformer.from_crs('epsg:4269', 'epsg:4326', always_xy=True)
10 points = list(zip(loaction_info.longitude, loaction_info.latitude))
11 coordsWgs = np.array(list(transformer.itransform(points)))
12
13 loaction_info['lonWgs'] = coordsWgs[:, 0]
14 loaction_info['latWgs'] = coordsWgs[:, 1]
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
265
```

## assign nodes and compute centroids

```
def assignment(mdist,centroids):
    #assign each instance to its closest cluster
    assign = dict()
    for n in list(mdist.index):
        min_value = 9999
        node = ''
        for c in centroids:
            if min_value > mdist[c].loc[n]:
                min_value = mdist[c].loc[n]
                node = c
        assign[n] = node

    cluster = dict()
    for c in centroids:
        temp = list()
        for i in range(len(assign)):
            if list(assign.values())[i] == c:
                temp.append(list(assign.keys())[i])
        cluster[c] = temp
    return assign,cluster

def update(centroids,cluster):
    #update centroids
    new_centroids = list()
    for c in centroids:
        occurence = list()
        freq = dict()
        max_value = 0
        max_key = ''
        for n in cluster[c]:
            occurence.extend([n])
        for node in occurence:
            if node not in freq:
                freq[node] = occurence.count(node)

        for node in freq:
            if freq[node] > max_value:
                max_value = freq[node]
                max_key = node
        new_centroids.append(max_key)
    return new_centroids
```

## Clustering

```
for i in range(1000):
    assign,cluster = assignment(mdist,centroids)
    centroids = update(centroids,cluster)
```

```
df_1 = pd.DataFrame()
for l in list(cluster.keys()):
    df_2 = pd.DataFrame(columns = ['node','centroids'])
    df_2['node'] = cluster[l]
    df_2['centroids'] = l
    df_1 = pd.concat([df_1,df_2], axis=0)
df_1.to_csv('SnnCluster_result.csv',index = False)
df_1.groupby('centroids').size()
```

## 7. References

He, Z. (2014, August). Hub selection for hub-based clustering algorithms. In 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (pp. 479-484). IEEE.

Kaufman, L., & Rousseeuw, P. J. (2009). Finding groups in data: an introduction to cluster analysis (Vol. 344). John Wiley & Sons.

Pyclustering. (2021). K-Medoids Class Reference. Retrieved from [https://pyclustering.github.io/docs/0.10.1/html/d0/dd3/classpyclustering\\_1\\_1cluster\\_1\\_1kmedoids\\_1\\_1kmedoids.html#a27155cba825aeaec306d59588e11bcfa](https://pyclustering.github.io/docs/0.10.1/html/d0/dd3/classpyclustering_1_1cluster_1_1kmedoids_1_1kmedoids.html#a27155cba825aeaec306d59588e11bcfa)