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

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 getaway 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[1]. 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.

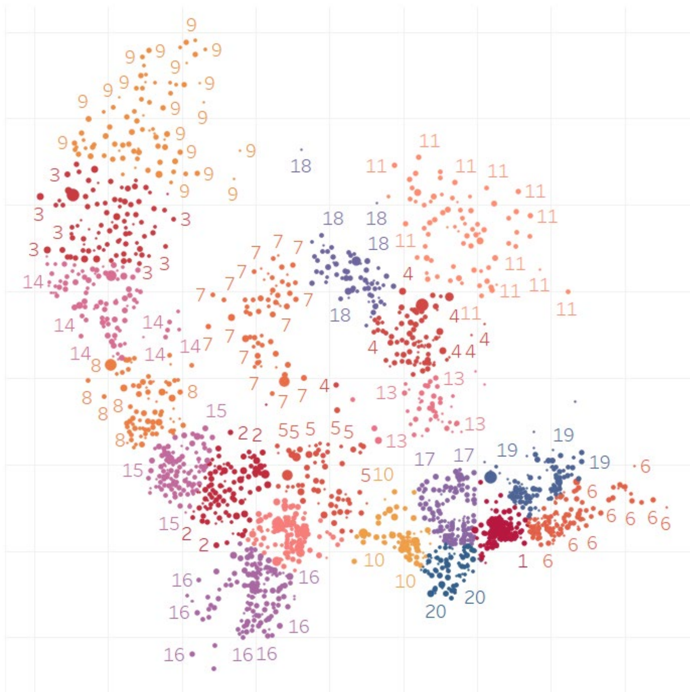
Table 4.1-1 Cost comparison for different number of clusters based on the baseline model

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

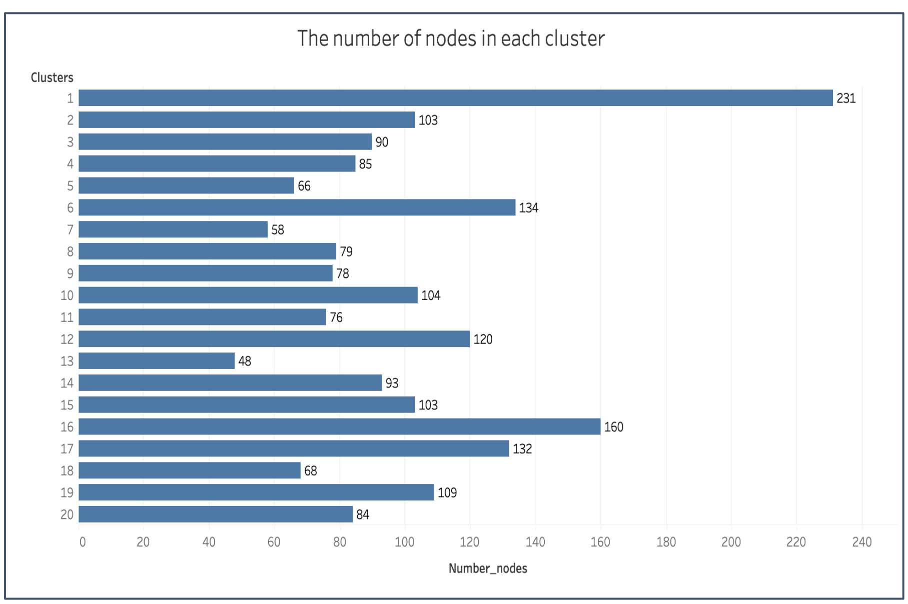
iv. The result of clustering (tables, graphs)

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: MAX[70, 70+4.5*(distance-5)]

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

$$g_j' = MAX\left(5, g_j\right)$$

$$\sum y_j * \left[70 + 4.5\left(g_j' - 5\right)\right] * A_j$$

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

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

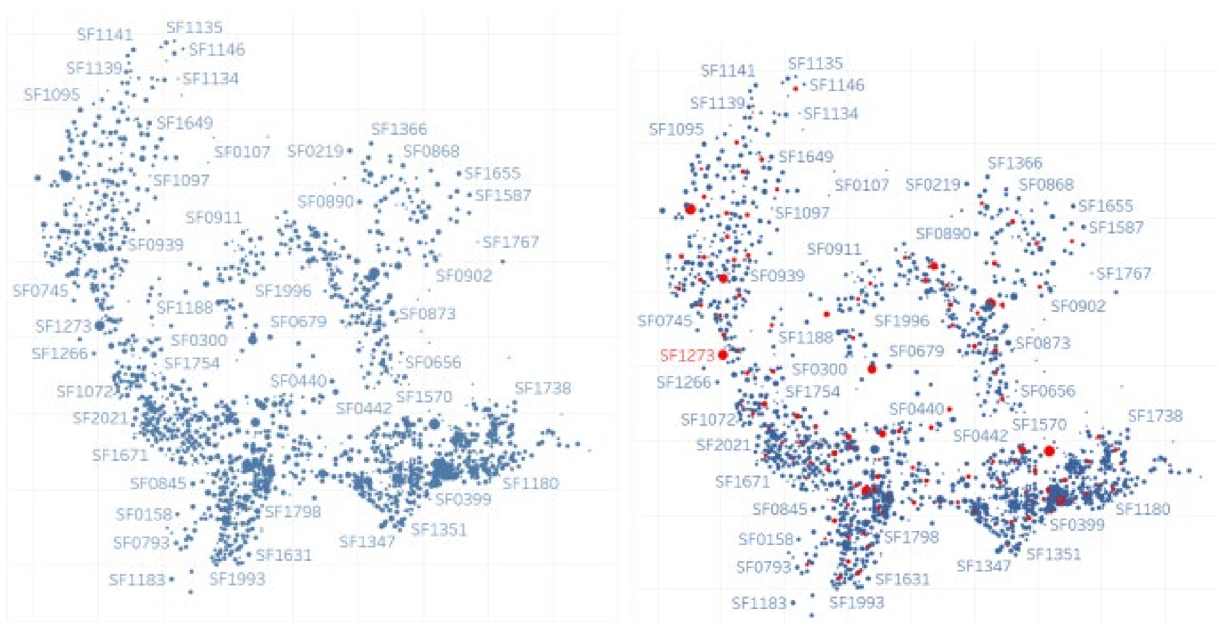
vi. Optimization Result

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

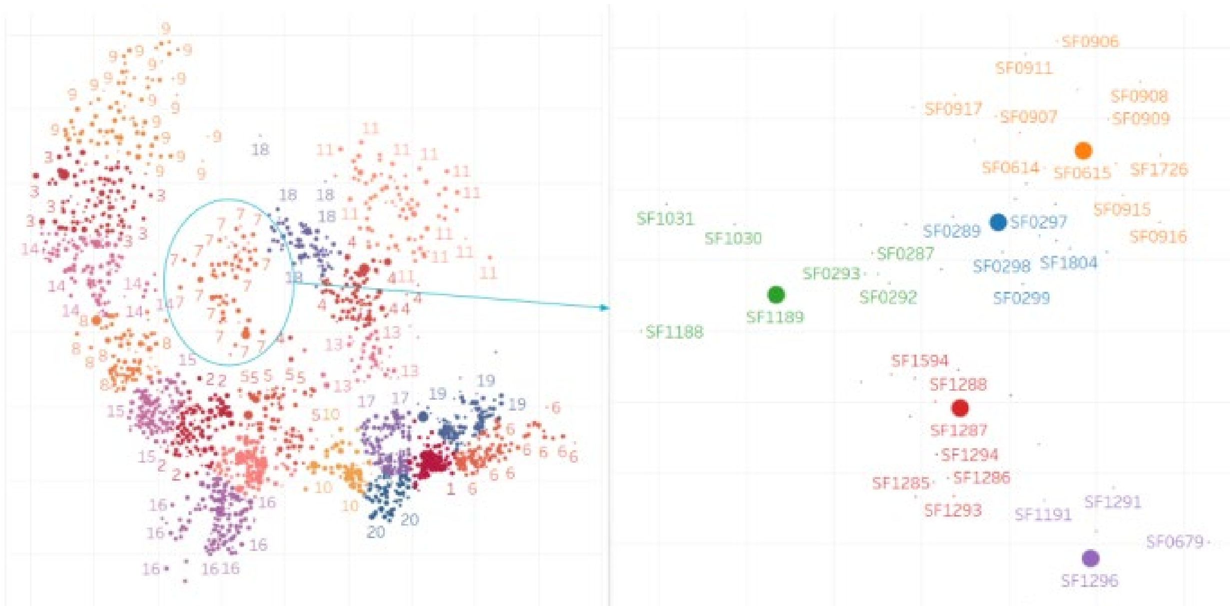
Graph 4.2-1

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

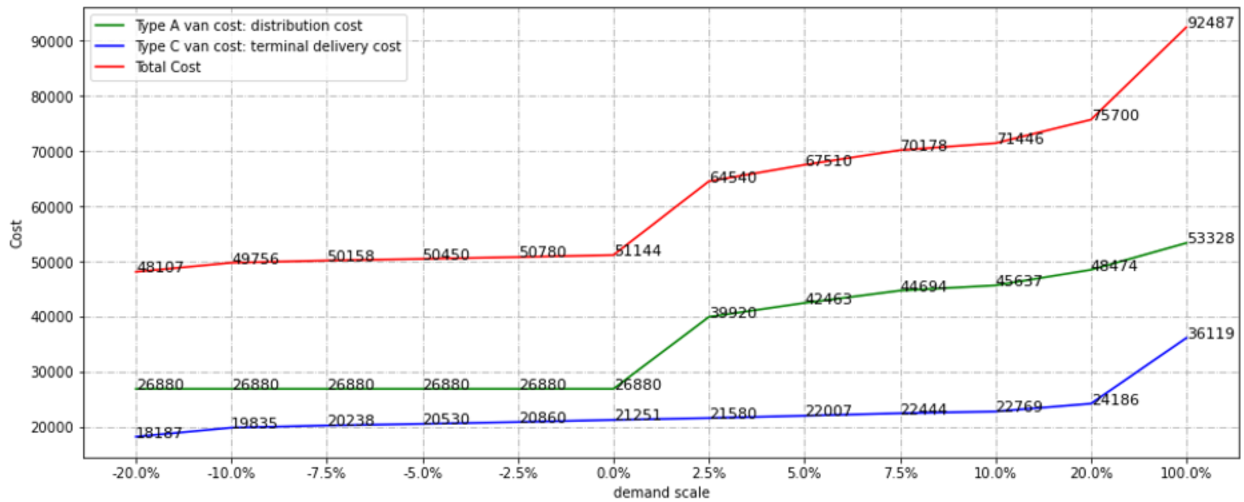
There is only one shift demand data provided by the client, and both the rent cost and unit transportation cost are fixed. However, these factors should fluctuate among shifts. Therefore, the goal of conducting sensitivity analysis is to check if our model is still valid when the input values change.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

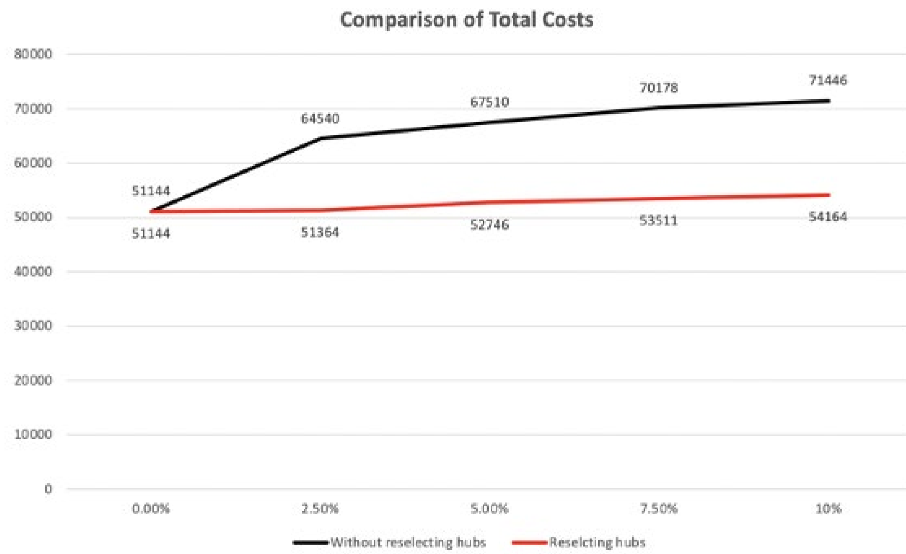


Graph 4.3-1 Cost decomposition while demand scales changes

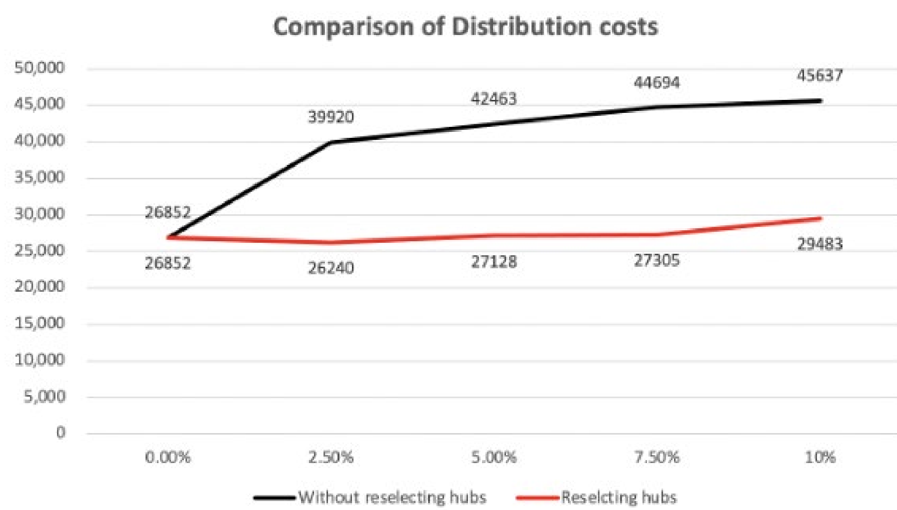
The upper red line represents the total cost. The middle green line represents distribution cost, which is Type A van cost. And the bottom blue line represents terminal delivery cost, which is the Type C van cost.

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 re-selecting the hubs.



Graph 4.3-2 Comparison of Total Cost



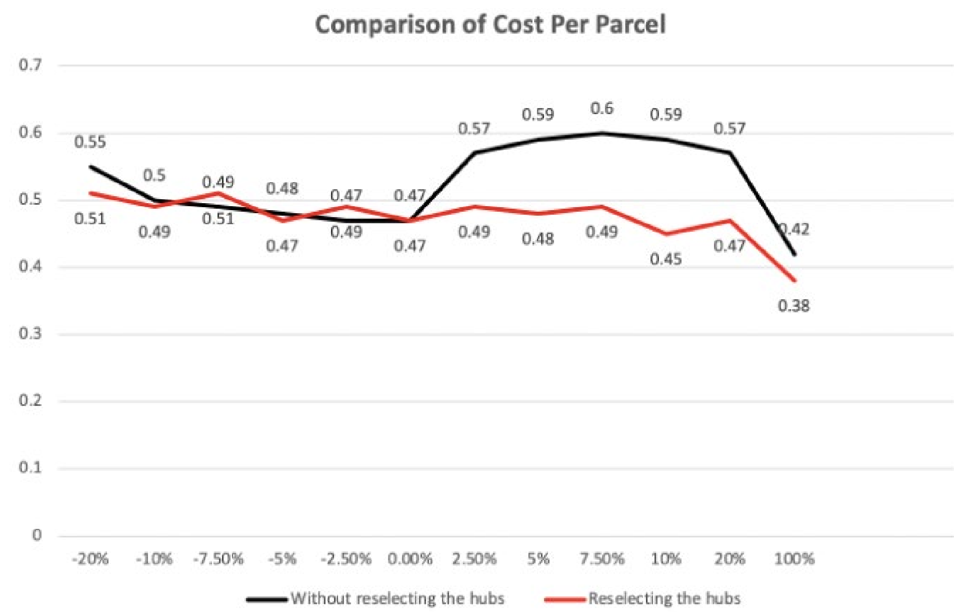
Graph 4.3-3 Comparison of Distribution Cost

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.

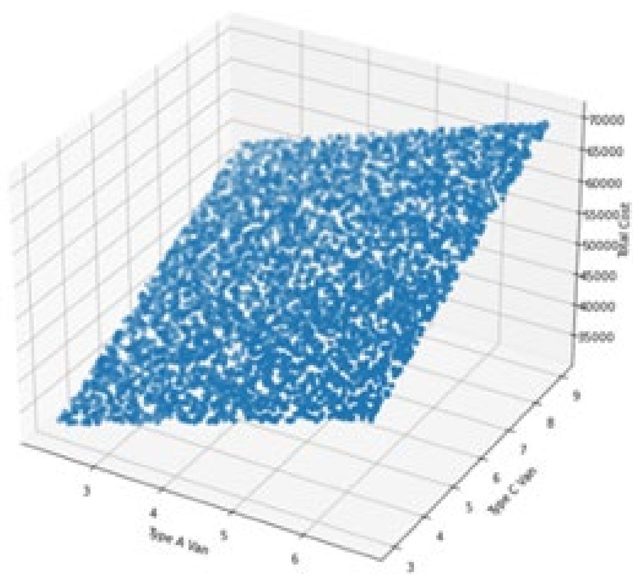
Beyond subtle and modest demand changes, we also consider the situation of demand surging, for example, during e-commerce campaigns. We found that the number of parcels was doubled during the double 11 campaign last year for SF express. So, we also check how the model performs in this demand surging 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

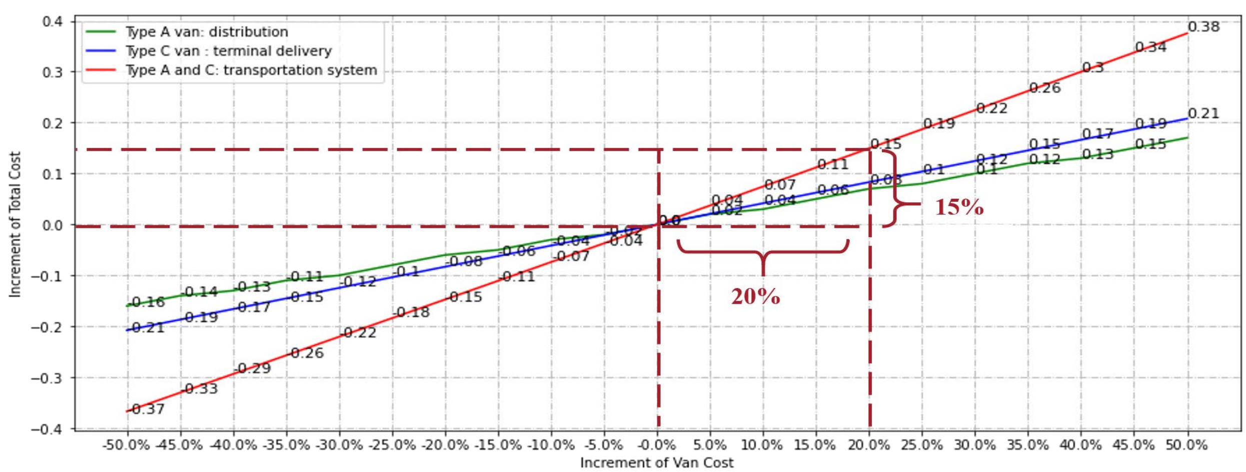
In the picture, the data point consists to a slanted plane. Total cost soars at a stable speed as the unit cost of type A vans and type C vans increase. And each unit's increment in type C and type A vans will give rise to a total cost increase by around 3,000 and 4,000 respectively.

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:

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

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

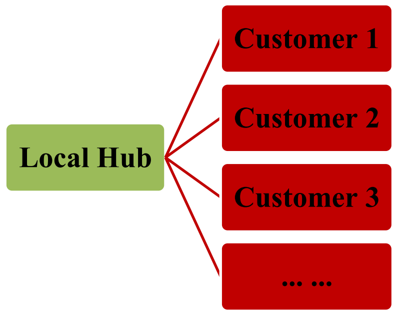
Name	sensitivity index	S_conf (95%)	Total sensitivity index	ST_conf (95%)
cost_a	0.366	0.048	0.369	0.032
cost_c	0.629	0.067	0.631	0.052

4.4 Modifications based on the base model

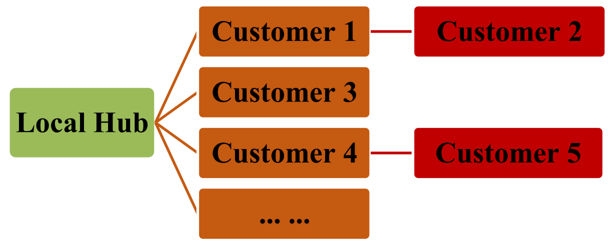
As the base model is under a simplified condition, three modifications are applied to the optimization model to make the result more practical as well as deducting the total cost.

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.



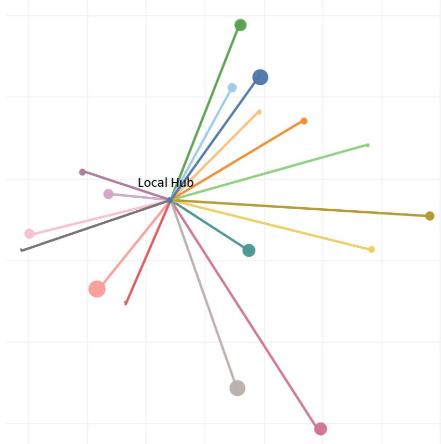
Graph 4.4-1 Base Model



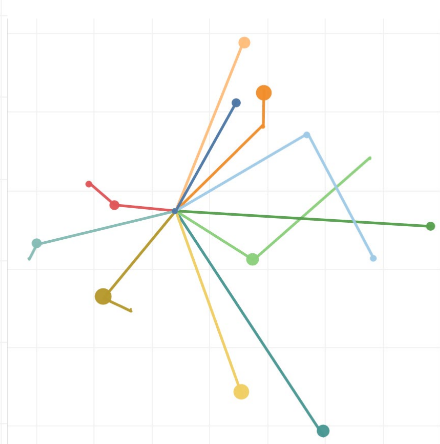
Graph 4.4-2 Two-nodes Terminal Delivery Routes

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

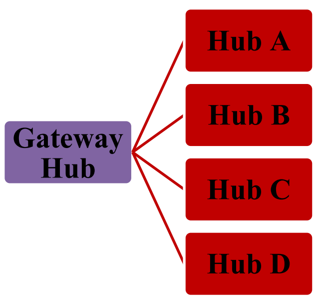
By reducing the total travel distance and the number of type C vans in usage, this modification improved the terminal delivery cost by 9.55% and the total cost by 3.97%.

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

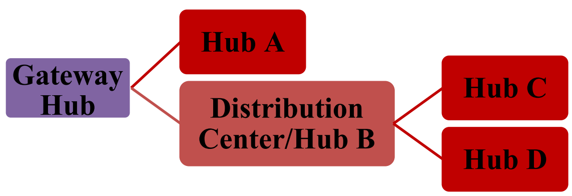
	Base Model	2-nodes Route	Improvement
Terminal Delivery Cost	21,251	19,222	9.55%
Total Cost	51,144	49,115	3.97%

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

	Base Model	Distribution Center	Improvement
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

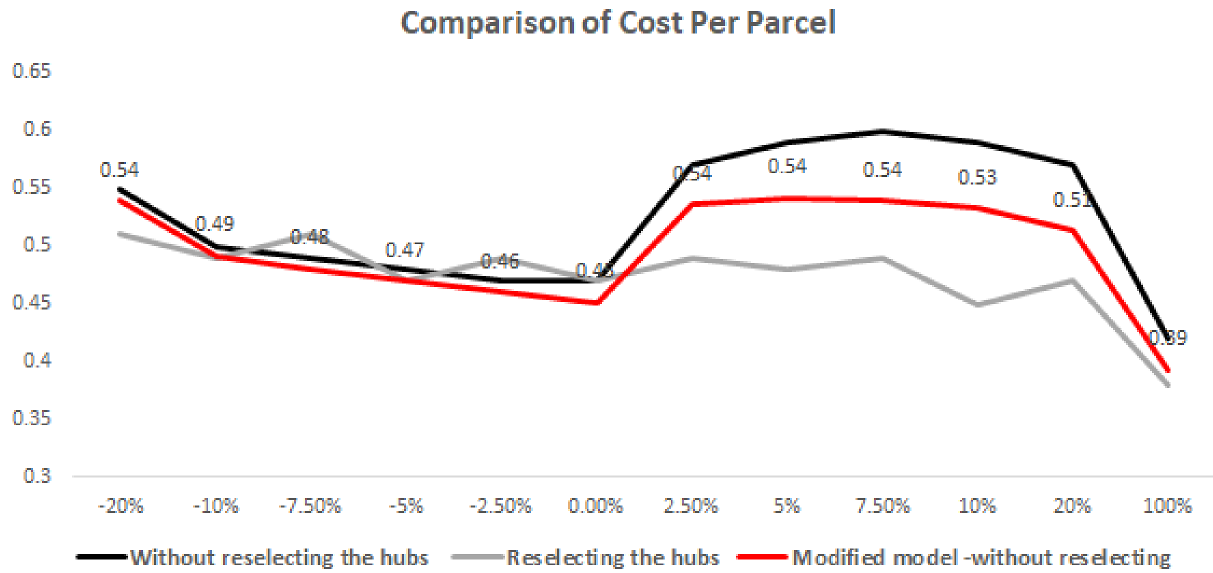
After performing the 3 modifications independently, a combined modification is conducted. Based on the optimization result of using type B vans, we implemented the modifications of 2 nodes terminal delivery routes and primary distribution centers, both allowing the usage of Type B vans.

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.

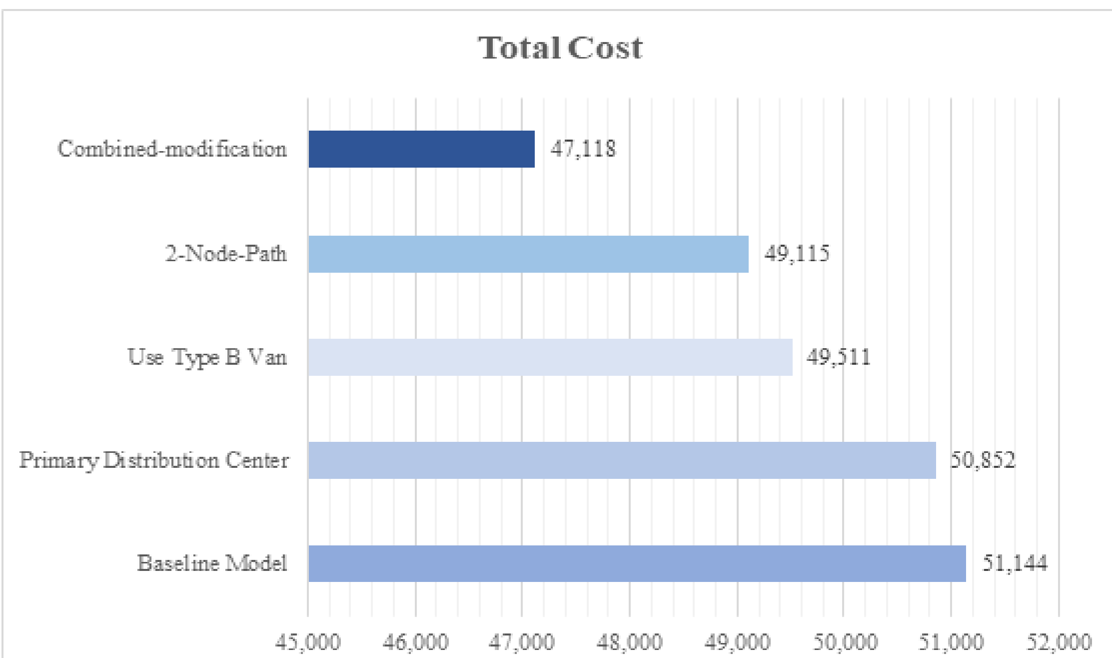


Graph 4.4-7 Comparison of Cost Per Parcel

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

	Direct	Baseline Model	Primary Distribution Center	Use Type B Van	2-Node-Path	Combined modification
Total Cost	676,309	51,144	50,852	49,511	49,115	47,118
Proportion of Direct Cost	-	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

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

S.t.

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

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

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

$$40C1_j \geq \sum_{i \in N} x_{ij} \cdot d_i \qquad \forall j \in N \tag{5}$$

$$40C2_i \geq d_i \qquad \forall i \in N \tag{6}$$

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

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

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 =

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

S.t.

$$x_{ij} \leq y_j \qquad \forall i, j \in N \tag{8}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad \forall i \in N \tag{9}$$

$$800A1_j \geq \sum_{i\in N} x_{ij} \cdot d_i \qquad \forall j \in N \tag{10}$$

$$800A2_i \geq d_i \qquad \forall i \in N \tag{11}$$

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 =

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

S.t.

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

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

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

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

b. Models and code

i. PAM Clustering

```python
# K-Medoids algorithm - PAM
# Running about 30 mins
from pyclustering.cluster.kmedoids import kmedoids
from pyclustering.cluster.center_initializer import kmeans_plusplus_initializer
from pyclustering.cluster import cluster_visualizer
from pyclustering.samples.definitions import FCPS_SAMPLES
from pyclustering.utils import read_sample, calculate_distance_matrix


initial_medoids = [102] + [101] * 19

# create K-Medoids algorithm for processing distance matrix instead of points
kmedoids_instance = kmedoids(matrix, initial_medoids, data_type='distance_matrix')

# run cluster analysis and obtain results
kmedoids_instance.process()

clusters = kmedoids_instance.get_clusters()
medoids = kmedoids_instance.get_medoids()
```

ii. K-Means Clustering

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

loaction_info = pd.read_excel('../Coding and data/loaction_info.xlsx')
loaction_info = loaction_info[["customer_code", "longitude", "latitude"]]

from pyproj import Transformer
transformer = Transformer.from_crs('epsg:4269','epsg:4326',always_xy=True)
points = list(zip(loaction_info.longitude,loaction_info.latitude))
coordsWgs = np.array(list(transformer.itransform(points)))

loaction_info['lonWgs']=coordsWgs[:,0]
loaction_info['latWgs']=coordsWgs[:,1]
```

```python
# K-Means algorithm
from pyclustering.cluster.kmedoids import kmedoids
from pyclustering.cluster.center_initializer import kmeans_plusplus_initializer
from pyclustering.cluster import cluster_visualizer
from pyclustering.samples.definitions import FCPS_SAMPLES
from pyclustering.utils import read_sample, calculate_distance_matrix

from pyclustering.cluster.kmeans import kmeans, kmeans_visualizer
from pyclustering.cluster.center_initializer import kmeans_plusplus_initializer
from pyclustering.samples.definitions import FCPS_SAMPLES
from pyclustering.utils import read_sample

loaction_info = loaction_info[["longitude", "latitude"]]
loaction_info_list = loaction_info.values.tolist()

# Prepare initial centers using K-Means++ method.
initial_centers = kmeans_plusplus_initializer(loaction_info_list, 20).initialize()

# Create instance of K-Means algorithm with prepared centers.
kmeans_instance = kmeans(loaction_info_list, initial_centers)

# Run cluster analysis and obtain results.
kmeans_instance.process()
clusters = kmeans_instance.get_clusters()
final_centers = kmeans_instance.get_centers()

# Print allocated clusters.
print("Clusters:", clusters)
```

ii. SNN Clustering

Initialization

```python
#initialization for centroids
node_list = dist['node_1'].unique()
np.random.seed(3)
initial_idx = np.random.randint(0,len(node_list)-1,20)
centroids = node_list[initial_idx]
centroids
```

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(nn_list[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