

After the Catastrophe: The Resilience of Baltimore

Summary

The transportation network is closely related to the development of a city, influencing various aspects such as people's daily lives, the transportation efficiency of goods and commodities, and the city's economic development. The transportation network in Baltimore is severely affected by the aging infrastructure, posing potential safety hazards to the smooth operation of the network. In this paper, we construct a transportation network analysis model and propose spatially - adaptable optimization strategies.

Firstly, our transportation network analyzes traffic by computing the importance of nodes. Since traditional algorithms often overlook the actual network conditions and merely focus on the topological structure, we have innovatively improved the traditional **K - Shell node importance evaluation algorithm**. We substitute the "degree" of nodes in the K - Shell algorithm with a combination of the weighted degree of nodes and the node's own weight. The node weight is mainly determined by the net passenger flow at the node, and the edge weight is determined by the average travel time of the road deduced from the **BPR** function. Considering that different stakeholders have diverse demands for the transportation network, we categorize them into three groups and investigate the GDP shares of different stakeholder groups, which renders the model we establish more comprehensive.

For Problem 1, we apply the model to analyze the situation before and after the collapse of the Francis Scott Key Bridge. By comparing the changes in node importance before and after the bridge collapse, we find that the importance of the I - 895 tunnel, the I - 95 tunnel, and the inland detour routes has increased significantly. Additionally, we assess the impact of the bridge collapse on various stakeholders.

For Problem 2, we first use the **K-Prototypes clustering algorithm** to cluster the data related to bus stops, and then we select one of the clusters for analysis. With the help of the node importance in the transportation network, the number of bus routes, and the passenger flow at bus stops, we believe that Baltimore should add the route shown as the red line in Figure 9.

For Problem 3, we first analyze the potential safety hazards of traffic congestion in the Baltimore transportation network. Then, based on the **LWR model** and the node importance scores, we determine the exponential growth model of traffic congestion. Finally, according to the characteristic that traffic congestion "grows exponentially in the early stage and slowly in the later stage", we formulate a congestion management strategy.

Finally, we conduct a sensitivity analysis of the parameter α in the K - Shell algorithm to illustrate the robustness of the model.

Keywords: Transportation Network; K - Shell; BPR; K - Prototypes; Baltimore

Contents

1	Introduction	3
1.1	Problem Background	3
1.2	Problem Restatement	3
1.3	Literature Review	4
1.4	Our Work	5
2	Assumptions	5
3	Notation	6
4	Modeling of the Baltimore Transportation Network	6
4.1	Representation of the Network Model	6
4.2	Improved K - Shell Decomposition Algorithm Based on Node and Edge Weights	6
4.2.1	Classification of Known Data	7
4.2.2	Improved K - Shell Decomposition Algorithm	8
4.3	Determination of Node and Edge Weights	8
4.3.1	Determination of Node Weight	8
4.3.2	Determination of Edge Weight	9
4.4	Determination of α by Stakeholders	10
4.5	Data Preprocessing	11
4.5.1	Data Screening	11
4.5.2	Missing Value Filling: Fast - filling Algorithm to Determine AADT .	11
4.6	Visualization of the Baltimore Transportation Network	12
5	Problem 1: Application of the Transportation Network Model to the Collapse of the Francis Scott Key Bridge	13
6	Problem 2: Application of the Transportation Network Model to Optimize Bus Routes	14
6.1	Feature Weight Analysis Based on PCA	14

6.1.1	Data Selection and Processing	14
6.1.2	Principal Component Analysis	14
6.1.3	Feature Weight Calculation	15
6.2	Bus Stop Clustering Based on K-Prototypes	16
6.3	Bus Route Optimization Based on Clustering Results	17
6.4	Impact on Stakeholders in the Surrounding Areas	18
7	Problem 3: Congestion Management Strategy for the Baltimore Transportation Network	18
7.1	Traffic Hazards in Baltimore	18
7.2	Exponential Model of Traffic Congestion and Congestion Management Strategy	18
7.3	Impact of Traffic Congestion Management Strategy on the Interests of All Parties	20
8	Sensitivity Analysis	20
9	Advantages and Disadvantages	21
9.1	Advantages	21
9.2	Disadvantages and Improvements	21
9.2.1	Disadvantages	21
9.2.2	Improvements	21
Letter		21
Reference		23
10	Appendix	24
10.1	Stakeholders and Their GDP Data Sources	24
10.2	Improved K-Shell algorithm based on node weights and edge weights . . .	24

1 Introduction

1.1 Problem Background

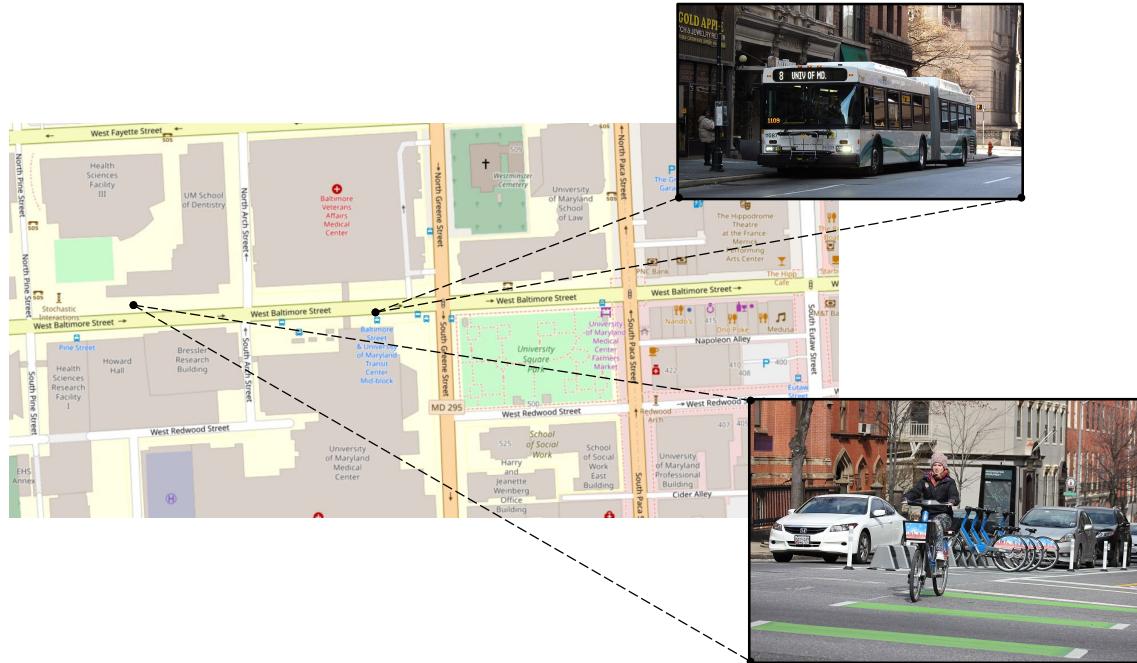


Figure 1: Various Transportation in Baltimore: Bicycles and Buses

The transportation network is akin to the arteries of a city, which is crucial for a city's development and the living standards of its residents. This is also the case for Baltimore, Maryland, USA. Affected by the aging original infrastructure and limited transportation options, the Baltimore City Government is currently focusing on redesigning and optimizing its transportation system. The aim is to boost urban development, enhance the living standards and satisfaction of all stakeholders, while also adapting to future demands and ensuring sustainable development.

1.2 Problem Restatement

Based on the problem background information, we need to model the transportation network of Baltimore to improve the commuting experience of stakeholders. The main tasks are as follows:

1. Based on the data provided in the problem, model and visualize the transportation network of Baltimore.
2. Analyze the impact of the collapse of the Francis Scott Key Bridge on the transportation system in Baltimore and the impact on each stakeholder.

3. Select a project that affects the bus system or the sidewalk system, analyze the impact of our Baltimore transportation network model on it, and analyze the impact on each stakeholder.
4. Recommend a transportation network project that can best improve the lives of Baltimore residents and analyze it.

1.3 Literature Review

According to the requirements put forward in section 1.2, we have searched a large number of literatures. The methods for analyzing key nodes in the network structure mainly include centrality methods based on the physical structure of the network¹, degree - centrality algorithm², betweenness - centrality algorithm³, K - Shell algorithm⁴, PageRank algorithm⁵, etc. However, the above - mentioned methods only search for factors reflecting the importance of nodes from a single aspect, without considering the influence of various indicators in practical problems. They cannot fully explore the hidden characteristics of the network, so the accuracy of key - node identification results is relatively low. Based on these methods, this paper proposes an improved K - Shell decomposition algorithm based on node weights and edge weights and has achieved exciting results.

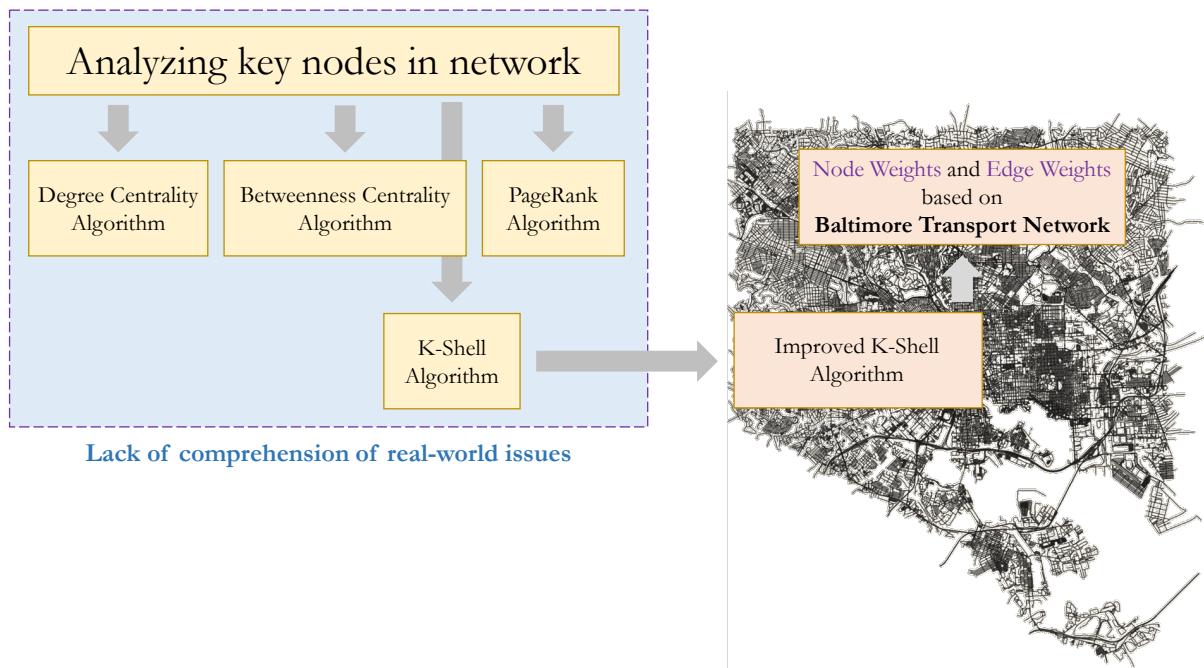


Figure 2: Thought Map of Literature Review

1.4 Our Work

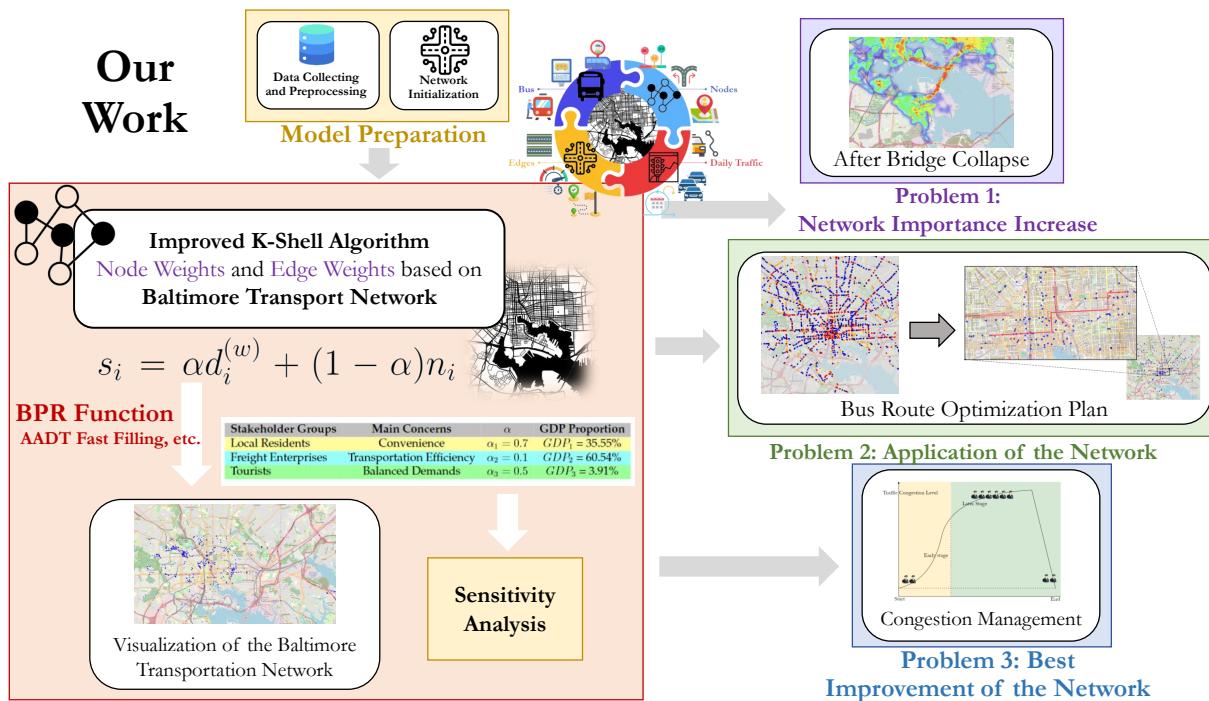


Figure 3: Our Work

2 Assumptions

1. Assume that the passenger flow at bus stops can, to a certain extent, reflect the nearby human flow, and after passengers get off, the passenger flow is evenly distributed on both sides of the road.
2. Assume that the private road segments with `access=private` in the city of Baltimore do not belong to the urban transportation network.
3. Assume that when considering the bus routes in Baltimore, `rider_total` and `Stop_Rider` in the dataset can fully represent the passenger flow.

3 Notation

Table 1: Symbols and Definitions

Symbol	Definition
$d_i^{(w)}$	Weighted degree of node
n_i	Node weight
s_i	Node score
α	Proportional coefficient of $d_i^{(w)}$
score	Evaluation score of the optimized bus route model
$\rho(x, t)$	Traffic density

4 Modeling of the Baltimore Transportation Network

4.1 Representation of the Network Model

According to section 1.3, when traditional methods are used to build a transportation network model, most of them only rely on the topological structure of the network to analyze the importance of nodes, while ignoring important factors of the city such as passenger flow, climate, and terrain. However, we propose a K - Shell model based on node weights and edge weights. **The ratio of node weights and edge weights can represent the interests of all parties**, as shown in the table. This model not only takes into account the topological structure of the graph network but also the actual situation of the city. We abstract the urban transportation as a complex network $G = (V, E)$, where $|V| = N$, $|E| = M$, and $E \subseteq V \times V$.

The intersections or breakpoints of roads in the city are defined as nodes in the network G , denoted as $V = \{v_1, v_2, \dots, v_N\}$, and the number of nodes is denoted as N . All the roads in the city are defined as edges in the network G , denoted as $E = \{e_1, e_2, \dots, e_M\}$, and the number of edges is denoted as M .

4.2 Improved K - Shell Decomposition Algorithm Based on Node and Edge Weights

Given a directed graph $G = (V, E)$ with node and edge weights, where V is the set of nodes and E is the set of edges. Each edge $e_{ij} \in E$ has a weight w_{ij} , and each node $v_i \in V$ has a weight n_i .

4.2.1 Classification of Known Data



Figure 4: Types of Known Data

We can divide the Baltimore data items into two categories. One category describes the road network and intersections, and the other describes traffic flow and its fluctuations. Some of these items are related to node weights, and some are related to edge weights. The division is as follows:

Data for calculating node weights comes, on the one hand, from nodes, such as their longitude and latitude positions X and Y , the number of streets connected to the node `street_count`, the intersection type `junction`, etc. On the other hand, it comes from data of bus stops, such as the position of the stop X and Y , the total number of passengers getting on and off at the stop `Rider_Tota`, the total cumulative number of passengers at the stop `Stop_Rider`, etc.

Data for calculating edge weights mainly comes from edges, such as their start and end points u and v , the number of lanes `lanes`, the maximum speed `maxspeed`, the length `length`, the service type `service`, etc. There is also some data from various traffic volumes, such as the ridership of bus routes `Distributi`, `AAWDT`, `AVMT`, `AADT`, etc. Finally, the flow direction of vehicles is also very important, such as the directional flow adjustment coefficient $D - Factor$, the peak - hour traffic volume adjustment coefficient $K - Factor$, etc.

4.2.2 Improved K - Shell Decomposition Algorithm

Algorithm 1: The Improved K - Shell Decomposition Algorithm for Key Node Identification

Input: Graph network $G = (V, E)$

Output: Node importance ranking result

Step 1: Calculate the initial weighted degree of each node $d_i^{(w)} = \sum_{j \in N(i)} w_{ij}$, where $N(i)$ represents the set of adjacent nodes of node i .

Step 2: For each node i , calculate the score s_i given by a certain stakeholder using the formula $s_i = \alpha d_i^{(w)} + (1 - \alpha)n_i$. Here, α is a regulatory parameter, which can reflect the priority requirements of different stakeholders for node weights and edge weights in the transportation network. See section 4.4. Both $d_i^{(w)}$ and n_i are normalized using the min - max normalization method.

Step 3: Iteratively remove nodes with low scores

- Find the set S of nodes with the lowest comprehensive scores in the current graph.
- Assign these nodes to the current K - Shell layer and remove them from the graph.
- Update the weighted degrees of the remaining nodes:

$$d_j^{(w)} = d_j^{(w)} - w_{ij} \quad \text{for all } j \in N(i) \quad (1)$$

Step 4: Repeat the above steps for the remaining graph until all nodes are assigned to a certain K - Shell layer.

4.3 Determination of Node and Edge Weights

4.3.1 Determination of Node Weight

The node weight mainly considers the government's perspective. When the human flow at a certain node increases, it can promote local economic development, boost tourism, and increase the vitality of enterprises. Therefore, we use the net flow of the node as the standard for the node weight.

According to Assumption 1, the directions of passengers getting on and off are evenly distributed. We have:

- Passengers getting on the bus enter the road from nodes u and v at a ratio of $\frac{U_{uv}}{2}$, where U_{uv} represents the total number of passengers entering the road at this stop.
- Passengers getting off the bus head towards nodes u and v at a ratio of $\frac{D_{uv}}{2}$, where D_{uv} represents the total number of passengers leaving the road at this stop.

We define the set of adjacent edges of a certain node i as \mathcal{E} . First, calculate the outflow (passengers leaving the node) and inflow (passengers arriving at the node) of the node:

$$\begin{cases} \text{Outflow}(i) = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} U_{ij} \\ \text{Inflow}(i) = \frac{1}{2} \sum_{(j,i) \in \mathcal{E}} D_{ji} \end{cases} \quad (2)$$

Finally, we calculate the net flow of the node to represent the node weight:

$$\text{Net}(i) = \text{Inflow}(i) - \text{Outflow}(i) = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} U_{ij} - \frac{1}{2} \sum_{(j,i) \in \mathcal{E}} D_{ji} \quad (3)$$

4.3.2 Determination of Edge Weight

The edge weight is mainly measured by the travel time of local residents, tourists, and transit passengers. We collect indicators related to the edge weight from the data file, such as road type, the number of lanes on the road segment, the maximum speed limit on the road, whether the road is one - way, road length, road width, and the number of lanes. Based on these, we establish the following model:

The BPR function describes the relationship between travel time and traffic flow:

$$t = t_0 \left(1 + \alpha \left(\frac{V}{C} \right)^\beta \right) \quad (4)$$

where t represents the travel time, t_0 is the free - flow time (travel time without congestion), V represents the traffic flow (e.g., AADT, Annual Average Daily Traffic), C represents the road capacity, and α, β are empirical parameters (usually $\alpha = 0.15$ and $\beta = 4$)⁶⁷.

According to the BPR function, we can obtain the actual speed v :

$$v = \frac{L}{t} = \frac{L}{t_0 \left(1 + \alpha \left(\frac{V}{C} \right)^\beta \right)}, \quad (5)$$

where L is the road length. Since the free - flow speed $v_0 = \frac{L}{t_0}$, we have:

$$v = \frac{v_0}{1 + \alpha \left(\frac{V}{C} \right)^\beta}. \quad (6)$$

In addition, the road capacity C is affected by the number of lanes lanes and the road functional classification Functional Class (interstate highways, arterial roads, local roads). Finally, we introduce other adjustment factors:

- Road feature correction

- Tunnel (tunnel = yes): The speed is reduced by 15%, i.e., $v_{final} = v \times 0.85$.
- Bridge (bridge = yes): The speed is reduced by 10%, i.e., $v_{final} = v \times 0.9$.
- One - way road (oneway = True): The speed is increased by 5%, i.e., $v_{final} = v \times 1.05$.
- Influence of the number of lanes
 - The more lanes, the higher the traffic efficiency: $C = C_{base} \times \text{lanes}$.

After considering all factors, we can obtain:

$$v_{final} = \frac{v_0}{1 + \alpha \left(\frac{\text{AADT}}{C} \right)^\beta} \times f_{\text{tunnel}} \times f_{\text{bridge}} \times f_{\text{oneway}}, \quad (7)$$

where:

- $f_{\text{tunnel}} = 1 - 0.15 \times \text{is_tunnel}$, $\text{is_tunnel} \in \{0, 1\}$
- $f_{\text{bridge}} = 1 - 0.1 \times \text{is_bridge}$, $\text{is_bridge} \in \{0, 1\}$
- $f_{\text{oneway}} = 1 + 0.05 \times \text{is_oneway}$, $\text{is_oneway} \in \{0, 1\}$

Finally, the edge weight is determined by the travel time:

$$t = \frac{L}{v_{final}}, \quad (8)$$

where L represents the road length.

4.4 Determination of α by Stakeholders

After analyzing the situation in Baltimore, we divide the stakeholders into the following three main groups with different needs, and determine their α values according to their work or life characteristics in section 10.1, as shown in Table 2:

Table 2: Stakeholder Data

Stakeholder Groups	Main Concerns	α	GDP Proportion
Local Residents	Convenience	$\alpha_1 = 0.7$	$GDP_1 = 35.55\%$
Freight Enterprises	Transportation Efficiency	$\alpha_2 = 0.1$	$GDP_2 = 60.54\%$
Tourists	Balanced Demands	$\alpha_3 = 0.5$	$GDP_3 = 3.91\%$

Thus, we can establish the comprehensive network importance proportional coefficient for Baltimore:

$$\alpha_{\text{total}} = \sum_{i=1}^n \omega_i \cdot \alpha_i, \quad i = 1, 2, 3, \quad (9)$$

where ω_i is the weight of the stakeholder group, determined by the GDP proportion, i.e., $\omega_i = GDP_i$.

After calculation, the final weight is:

$$\alpha_{\text{total}} = 0.32534 \quad (10)$$

4.5 Data Preprocessing

4.5.1 Data Screening

Since the geographical scope of the data provided in the problem is the Baltimore Metropolitan Area, which is much larger than the scope of Baltimore City, we first screen out the data within the scope of Baltimore City according to longitude and latitude. The screening results are shown in the following table:

Table 3: Data Statistics Before and After Filtering

Data Table	Before Filtering	After Filtering
edge_all.csv	565,495 entries	201,003 entries
edge_drive.csv	91,227 entries	29,309 entries

4.5.2 Missing Value Filling: Fast - filling Algorithm to Determine AADT

AADT data is only collected on roads with bus stops, while other roads lack such data. However, the time complexity of directly using the shortest - distance method to fill the missing AADT values is too high. Therefore, we use the following fast - filling algorithm to determine the AADT values of all roads.

Algorithm 2: Fast Filling Algorithm for AADT Values of All Roads

Step 1: Assign the AADT value of all edges to 0, and update the AADT values of edges with bus stops.

Step 2: Traverse the edges that have been assigned AADT values. For each such edge e_i , traverse the edges e_j connected to it, and add the AADT value of e_i to the AADT list of e_j .

Step 3: Calculate the average value of the AADT list of e_j obtained in Step 2, and use it as the AADT value of e_j .

Step 4: Repeat Step 2 and Step 3 until there is no edge with an AADT value of 0 in the entire network.

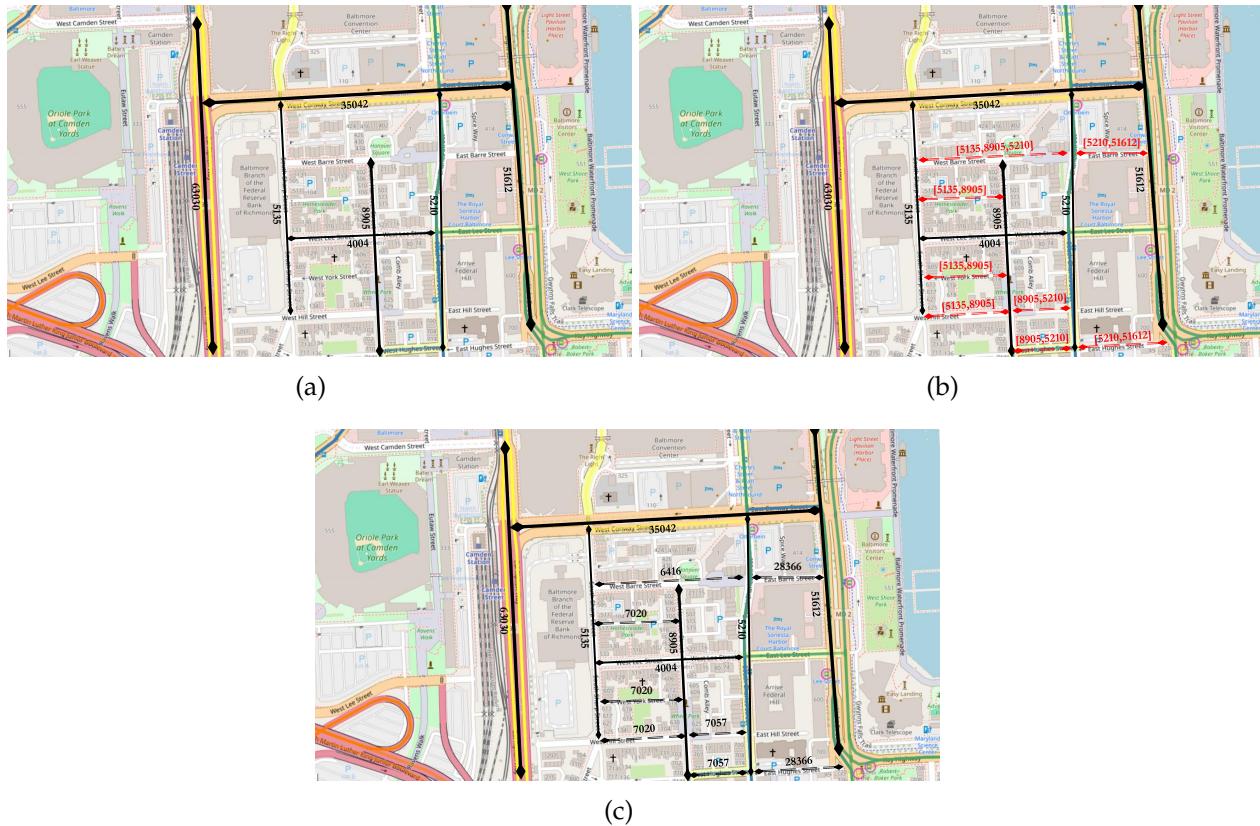


Figure 5: Example of AADT Filling

4.6 Visualization of the Baltimore Transportation Network

Based on the Baltimore urban transportation network model, we calculate the top 1000 nodes in terms of importance and visualize them. The results are shown in the figure below:

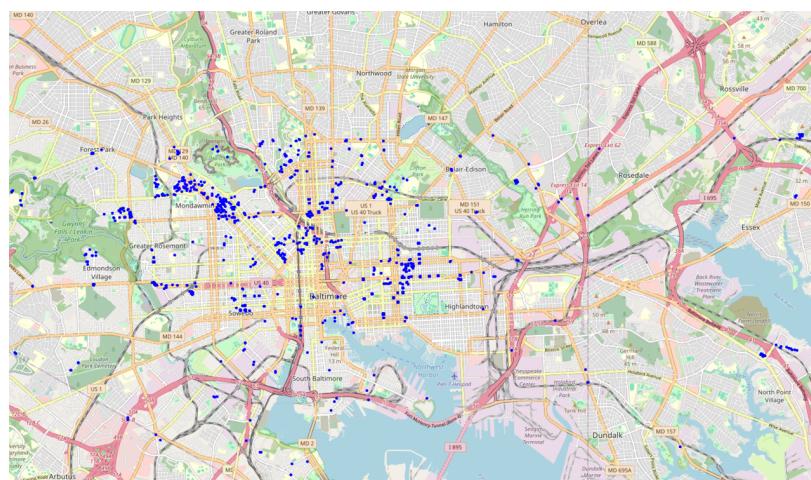


Figure 6: Visualization of the Top 1000 Important Nodes

5 Problem 1: Application of the Transportation Network Model to the Collapse of the Francis Scott Key Bridge

On March 26, 2024, a traffic accident caused the collapse of the Francis Scott Key Bridge, resulting in traffic paralysis. Based on the above - mentioned Baltimore transportation network model, our team analyzed the changes in node importance before and after the bridge collapse. We found that after the bridge collapse, the importance scores of adjacent road nodes (such as the I - 95 Tunnel and the I - 895 Tunnel) increased, and they became congestion hotspots on alternative routes. We calculated the node importance scores of the Baltimore transportation network before and after the bridge collapse respectively, and then drew a heatmap of the percentage increase in node importance after the bridge collapse. The results show that the importance of the I - 895 Tunnel increased most significantly, by about 63.701%, followed by the I - 95 Tunnel, about 47.258%, and the importance of the detour roads in the city also increased by an average of 10% - 20%. This also reveals the problems in the Baltimore urban transportation network: insufficient path redundancy and broken multi - mode connections.

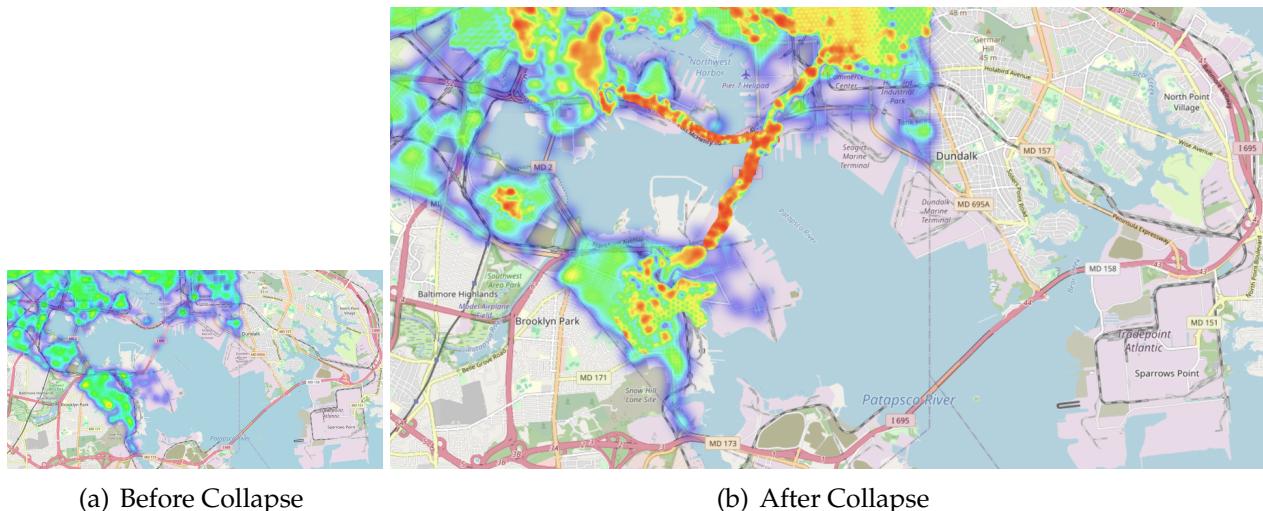


Figure 7: Heatmap of Importance Increase

Then, we calculated the increase in the importance of nodes in the Baltimore urban transportation network according to the α values of three different stakeholders: local residents, freight enterprises, and tourists. The results are shown in the following table. We found that before and after the collapse of the Francis Scott Key Bridge, the change in the importance of nodes in the Baltimore transportation network for local residents was minimal. This is because the activity scope of local residents is relatively fixed, and they generally do not go to the other side of the river. The change for tourists was slightly greater, as they are more likely to go to the other side than local residents. The change for freight enterprises was the highest, as they need to travel between the two sides frequently. In addition, it is worth noting that due to possible restrictions on passing vehicles

in the I - 895 Tunnel and the I - 95 Tunnel, vehicles have to detour in the city, so the percentage increase in the importance of detour roads in the city is relatively high.

Table 4: Importance Increase of Different Nodes for Stakeholder Groups

Stakeholder Groups	I - 895 Tunnel	I - 95 Tunnel	Detours within the City
Local Residents	62.375%	46.046%	18.901%
Tourists	64.528%	47.568%	15.679%
Freight Enterprises	64.796%	48.586%	12.179%

6 Problem 2: Application of the Transportation Network Model to Optimize Bus Routes

6.1 Feature Weight Analysis Based on PCA

6.1.1 Data Selection and Processing

In the bus stop dataset `Bus_Stops.csv`, we need to analyze the features of bus stops, determine the contribution weight of each feature to the clustering result, and perform clustering based on these weights. After screening, we removed some columns that were meaningless, had repeated meanings, or had single - valued data, and selected the following data indicators: Categorical variables `Mode`, `Shelter`, and numerical variables `X`, `Y`, `Rider_Total`, `Stop_Rider`.

Since categorical features cannot be directly used in PCA⁸, we first perform One - Hot encoding on them, and then determine the weight of each feature through PCA analysis. PCA is sensitive to the scale of data, so we need to standardize the numerical features.

6.1.2 Principal Component Analysis

The standardized data matrix is $\mathbf{X} \in \mathbb{R}^{n \times p}$, where n is the number of samples and p is the number of features. The formula for calculating the covariance matrix \mathbf{C} is:

$$\mathbf{C} = \frac{1}{n-1} \mathbf{X}^T \mathbf{X} \quad (11)$$

Perform eigenvalue decomposition on the covariance matrix \mathbf{C} :

$$\mathbf{C} = \mathbf{V} \Lambda \mathbf{V}^T \quad (12)$$

where \mathbf{V} is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues.

The principal components \mathbf{P} are the projections of the original data in the direction of the eigenvectors:

$$\mathbf{P} = \mathbf{X} \mathbf{V} \quad (13)$$

The variance contribution rate of each principal component is:

$$\text{Explained Variance Ratio}_i = \frac{\lambda_i}{\sum_{j=1}^p \lambda_j} \quad (14)$$

where λ_i is the i -th eigenvalue.

6.1.3 Feature Weight Calculation

For categorical features, such as Mode and Shelter, we use One - Hot encoding to convert them into binary vectors. Suppose the categorical feature C has k values, then the feature matrix after One - Hot encoding is:

$$\mathbf{C}_{\text{encoded}} \in \mathbb{R}^{n \times k} \quad (15)$$

Combine the standardized numerical features $\mathbf{X}_{\text{scaled}}$ and the One - Hot encoded categorical features $\mathbf{C}_{\text{encoded}}$:

$$\mathbf{X}_{\text{processed}} = [\mathbf{X}_{\text{scaled}}, \mathbf{C}_{\text{encoded}}] \quad (16)$$

Perform PCA analysis on the combined data $\mathbf{X}_{\text{processed}}$ to obtain the variance contribution rate Explained Variance Ratio _{i} of each principal component. Combine the variance contributions of the One - Hot encoded features back to the original categories. Suppose the variance contributions corresponding to the One - Hot encoded columns of the original categorical feature C are Explained Variance Ratio _{j} , then the weight of the original category is:

$$\text{Weight}_i = \sum_{j \in \text{Category}_i} \text{Explained Variance Ratio}_j \quad (17)$$

Normalize the weights so that their sum is 1:

$$\text{Normalized Weight}_i = \frac{\text{Weight}_i}{\sum_{j=1}^p \text{Weight}_j} \quad (18)$$

So far, we have calculated the weights of the shown data, and the results are as follows:

Table 5: Data Meanings and Their Weights

Data Meaning	Variable	Weight
Total Number of Riders	Rider_Tota	0.62830
Total Cumulative Number of Riders at Stops	Stop_Rider	0.21777
Transportation Mode Type (e.g., bus)	Mode	0.03646
Whether There is a Passenger Shelter	Shelter	0.00093
Service Type of Bus	Routes_Ser	0.11650

6.2 Bus Stop Clustering Based on K-Prototypes

The K-Prototypes algorithm⁹ is a clustering method capable of handling both numerical and categorical data simultaneously. Our dataset contains both numerical and categorical features, which fully meets the requirements of the K - Prototypes algorithm. In addition, as calculated in Table 5 through PCA analysis, we have obtained the weight contributions of each feature, and these weights will be used to optimize the clustering process.

Unlike the K-means algorithm, the K-Prototypes algorithm can calculate the similarity of numerical types not only through the Euclidean distance but also calculate the similarity of categorical types through the Hamming distance.

We performed clustering on the bus stop data in Baltimore. After selecting different values of k multiple times, we found that the effect was relatively ideal when $k = 5$. The visualization result of the clustering is shown in the figure below:

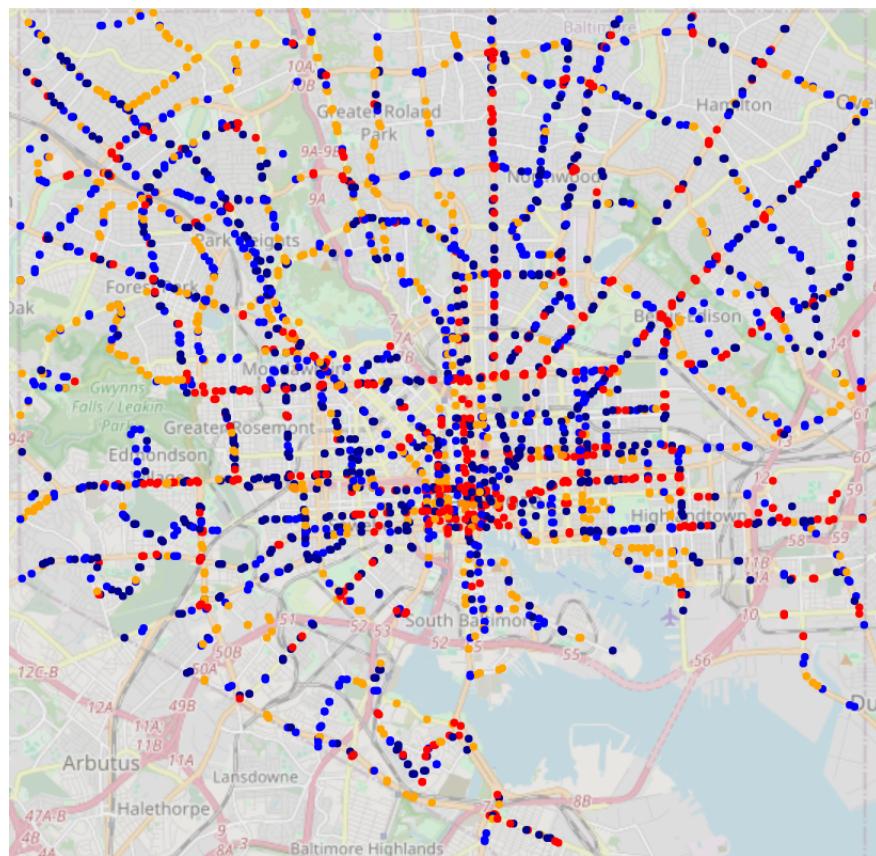


Figure 8: Visualization of Bus Stop Clustering

By observing the clustering result graph, we noticed that the stations with the highest importance are largely concentrated in the city center, distributed along busy traffic routes. While the stations with low importance are mostly concentrated in areas far from the city center, distributed along traffic routes with low human flow. Therefore, we

consider selecting the area where the stations with high importance are concentrated for further analysis.

6.3 Bus Route Optimization Based on Clustering Results

We extracted one part of the clustering results for analysis and attempted to plan the addition of bus routes for this area. To this end, we calculated the corresponding score for each station and added bus routes in places with high scores. Besides the node weights in Section 4, for bus routes, we should also pay attention to the influence of passenger flow and the number of bus routes on the score: the larger the passenger flow, the more bus routes should be added; and the fewer the number of bus routes, the more bus routes should be added. The specific formula for calculating the score is as follows¹:

$$score = \frac{\text{Passenger Flow} \times \text{Average Importance of Nodes on Both Sides of the Station}}{\text{Number of Bus Routes}} \quad (19)$$

Due to the geographical distribution of multiple hospitals and universities, we believe that adding bus routes to the red - line part is the optimal choice, as shown in Figure 9.

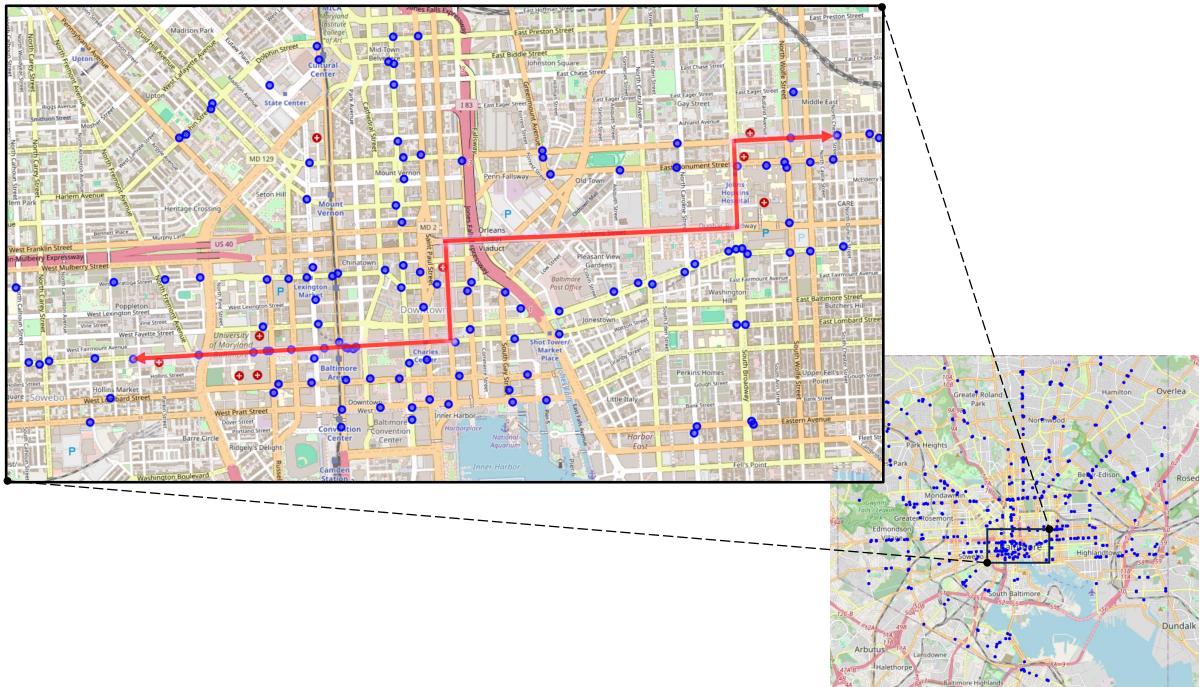


Figure 9: Bus Route Optimization Plan

¹According to Assumption 3, the passenger flow is Rider_Tota, and the number of bus routes is the length of the Routes_Ser list.

6.4 Impact on Stakeholders in the Surrounding Areas

After adding new bus routes at the places marked with the red line on the map, it has greatly facilitated the daily travel of the surrounding residents and significantly improved the travel experience of tourists. At the same time, it has reduced the willingness of residents such as university students to travel by car, providing a more convenient logistics channel for freight companies. This improvement not only strengthens the connection of the transportation network within the region but also promotes the local economic development, enabling residents, tourists, and enterprises to benefit from it.

7 Problem 3: Congestion Management Strategy for the Baltimore Transportation Network

7.1 Traffic Hazards in Baltimore

Traffic network congestion in the city has always been a safety hazard in Baltimore. It may not only increase the probability of traffic accidents but also lead to an increase in the crime rate in areas with inconvenient transportation due to the difficulty of the police arriving quickly. In addition, stampede accidents may occur in densely populated areas, and crimes such as theft and robbery are also likely to happen. Therefore, our team has designed a congestion management strategy for the Baltimore transportation network. By analogy with the congestion control algorithm in the field of computer networks, we adjust the signal light cycle and the number of traffic police dynamically in stages.

7.2 Exponential Model of Traffic Congestion and Congestion Management Strategy

According to the classical LWR model to describe the traffic density $\rho(x, t)$, we have:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho v(\rho)) = 0 \Rightarrow v(\rho) = v_{\text{free}} \left(1 - \frac{\rho}{\rho_{\max}} \right) \quad (20)$$

where $\rho(x, t)$ is the traffic flow density, $v(\rho)$ is the traffic flow speed, and ρ_{\max} is the road - carrying capacity.

Considering that the importance of the node w_i weakens the local traffic capacity:

$$\rho_{\max,i} = \rho_{\text{base}} \cdot (1 - \alpha w_i) \quad (21)$$

In the local road section (near node i), ignoring the spatial derivative term, the LWR equation degenerates into the Logistic equation:

$$\frac{d\rho_i}{dt} = k_i \rho_i \left(1 - \frac{\rho_i}{\rho_{\max,i}} \right) \quad (22)$$

Considering that the higher the importance of the node, the faster the congestion may spread. Define the modified growth rate:

$$k_i = k_{\text{base}} \cdot (1 + \beta w_i) \quad (23)$$

Substituting it into the above equation, we get:

$$\frac{d\rho_i}{dt} = k_{\text{base}}(1 + \beta w_i)\rho_i \left(1 - \frac{\rho_i}{\rho_{\text{base}}(1 - \alpha w_i)}\right) \quad (24)$$

Solving the above equation by the method of separation of variables:

$$\rho_i(t) = \frac{\rho_{\max,i}}{1 + e^{-k_{\text{base}}(1+\beta w_i)t} \cdot \left(\frac{\rho_{\max,i}}{\rho_{i,0}} - 1\right)} \quad (25)$$

When $\rho_i \ll \rho_{\max,i}$:

$$\rho_i(t) \approx \rho_{i,0} \cdot e^{k_{\text{base}}(1+\beta w_i)t} \quad (26)$$

In the early stage of congestion, the degree of congestion will increase exponentially with time. After reaching a certain critical value, the growth of the congestion degree begins to slow down, which is very similar to the congestion control in the TCP protocol¹⁰. Therefore, we propose the following congestion management strategies:

- Stage 1: In the initial stage of congestion, the degree of congestion increases exponentially with time. At this time, the signal light cycle should be adjusted at major intersections, and the number of traffic police should be increased rapidly.
- Stage 2: In the middle and late stages of congestion, the growth of the congestion degree slows down, and so does the number of additional traffic police.
- Stage 3: After the congestion ends, the signal light cycle and the number of traffic police return to normal levels.

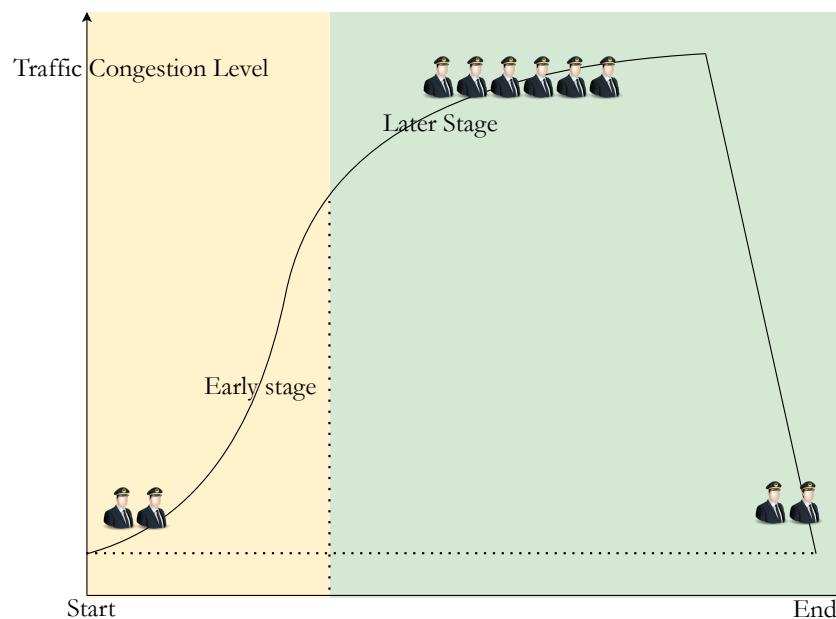


Figure 10: Schematic Diagram of Congestion Management Strategy

7.3 Impact of Traffic Congestion Management Strategy on the Interests of All Parties

Table 6: Stakeholders and Their Concerns

Stakeholders	Convenience of Travel	Safety of Travel
Local Residents	Shorter travel time, more travel options, and an improved quality of life	Fewer traffic accidents and the rapid arrival of police forces
Tourists	Enhanced travel convenience and an improved tourism experience	Reduced stampede accidents and a lower crime rate
Freight Enterprises	Higher transportation efficiency and lower transportation costs	Fewer traffic accidents and improved cargo safety

We summarize the impact of traffic congestion management on various stakeholders in Baltimore as shown in Table 6. The traffic congestion management strategy in Baltimore not only facilitates the travel of local residents, tourists, and freight enterprises but also has a positive impact on the safety of the city. By dynamically adjusting the signal light cycle and increasing the number of traffic police, traffic congestion can be effectively reduced, travel efficiency and safety can be improved, the quality of life of residents and the tourism experience of tourists can be enhanced, the transportation costs of freight enterprises can be lowered, and the economic development of the city can be promoted.

8 Sensitivity Analysis

We increased and decreased the α values calculated in Section 4.4 by 5% respectively, and calculated the mean squared errors between the results after the increase or decrease and the original α values. The results are shown in Table 7:

Table 7: Mean Squared Errors of α Values after Perturbation

	α Value	Mean Squared Error
Original Value	0.32534	\varnothing
Increased by 5%	0.341607	14.209293
Decreased by 5%	0.309073	8.608151

The results show that after making small perturbations to the parameter α , the resulting differences are within an acceptable range, which proves the robustness of our model.

9 Advantages and Disadvantages

9.1 Advantages

1. Our improved K-Shell decomposition algorithm with node weights and edge weights can comprehensively construct the traffic network graph model of the city from aspects such as the actual traffic situation and the topological structure of the traffic network, which is relatively comprehensive.
2. By introducing proportional coefficients for node weights and edge weights, the influence of various stakeholders is analyzed and deeply considered.
3. When clustering with the K-Prototypes algorithm, the weights of each type are determined by the PCA algorithm, which is relatively accurate.

9.2 Disadvantages and Improvements

9.2.1 Disadvantages

1. When calculating edge weights, the AADT value of each road is required. However, not every road has a bus stop. If the shortest - distance algorithm is directly used to fill the data, the time complexity is too high.
2. The k - means clustering algorithm calculates similarity based on Euclidean distance and cannot handle categorical data.

9.2.2 Improvements

1. Similar to the watershed algorithm in the field of computer vision¹¹, our team has constructed a fast algorithm for filling the missing AADT values. See section 4.2. This algorithm can complete the filling task with almost linear time complexity, meeting the filling requirements for large - scale graph network data.
2. K - Prototype is an extended version of K - means, which can handle both numerical and categorical data and can well complete the clustering task in Section 6.2.



Robust Baltimore

Dear Mayor of Baltimore,

Our team is greatly honored to present to you an innovative transport network model designed specifically for Baltimore. This model aims to offer a scientific and data-driven traffic governance solution. Below is an overview of our model and specific implementation suggestions.

Traditional models mostly rely on network topological structures and struggle to reflect real travel demands. Our model achieves precise analysis through the following breakthroughs:

Dynamic Weighting of Node Importance

- ❖ Replace the node degree value with a composite index of the weighted degree (traffic flow intensity) and the node's own weight (net passenger flow).
- ❖ Calculate the actual road travel time for edge weights using the BPR function, which reflects the congestion differences between morning and evening rush hours.

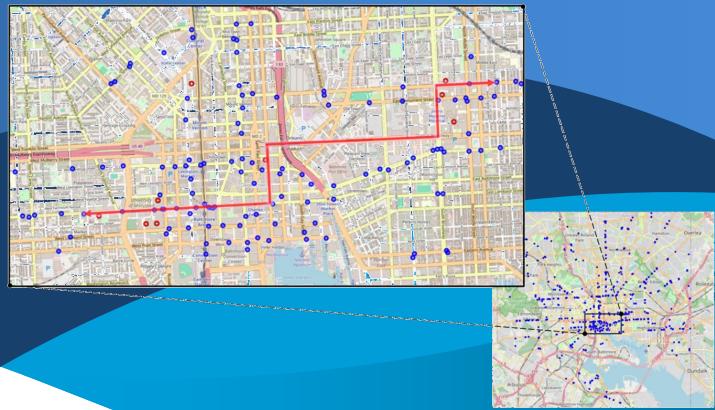
Balancing Mechanism for Stakeholders

- ❖ Introduce the GDP contribution weights of various groups to ensure the balanced demands of different functional nodes such as commercial and industrial areas, residential areas, and logistics hubs.

Projects

► Precise Expansion of the Bus Network

We analyzed all the bus stops in the city using the KP clustering algorithm. By comprehensively considering indicators such as passenger flow,

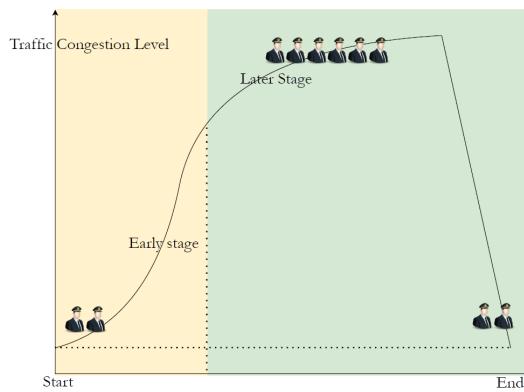


the number of bus routes, and the importance of nodes around the routes, we identified a bus route. This route passes by hospitals, schools, etc. Adding this bus route will greatly facilitate the travel of the surrounding residents.

► Index-based Control of Road Congestion

We constructed a traffic congestion propagation model based on the LWR fluid dynamics model and discovered the law that "traffic congestion grows exponentially in the early stage and slowly in the later stage".

- ⚠ In the early congestion stage, adjust signal light cycles at major intersections and quickly add traffic police.
- ⚠ In the mid - late congestion stages, keep adjusting signal light cycles and gradually increase the number of traffic police.
- ⚠ After congestion ends, return signal light cycles and the number of traffic police to normal.



We believe that by implementing the above - mentioned projects, we can not only significantly improve the traffic conditions in Baltimore but also further enhance the quality of life of the citizens. We look forward to having in - depth discussions with your team about these proposals and jointly advancing Baltimore towards a smarter and more efficient future.

Best regards,
Team #2500387

References

- [1] X. Ren and L. Lyu. A review of methods for ranking important nodes in networks. *Chinese Science Bulletin*, 2014(13):23, 2014.
- [2] J. Zhang and Y. Luo. Degree centrality, betweenness centrality, and closeness centrality in social network. In *2017 2nd international conference on modelling, simulation and applied mathematics (MSAM2017)*, pages 300–303. Atlantis press, 2017.
- [3] E. Bergamini, H. Meyerhenke, and C. L. Staudt. Approximating betweenness centrality in large evolving networks. In *2015 Proceedings of the Seventeenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 133–146. Society for Industrial and Applied Mathematics, 2014.
- [4] G. Maji, S. Mandal, and S. Sen. A systematic survey on influential spreaders identification in complex networks with a focus on k-shell based techniques. *Expert Systems with Applications*, 161:113681, 2020.
- [5] L. Tortosa, J. F. Vicent, and G. Yeghikyan. An algorithm for ranking the nodes of multiplex networks with data based on the pagerank concept. *Applied Mathematics and Computation*, 392:125676, 2021.
- [6] Y. Hamdouch, W. Y. Szeto, and Y. J. Jiang. A new schedule-based transit assignment model with travel strategies and supply uncertainties. *Journal of Transportation*, 67:35–67, 2014.
- [7] Z. Y. Gao, H. J. Sun, and L. L. Shan. A continuous equilibrium network design model and algorithm for transit systems. *Transportation Research Part B*, 38(3):235–250, 2004.
- [8] H. Hotelling. Analysis of a complex of statistical variables into principal components. *British Journal of Educational Psychology*, 24:417–520, 1933.
- [9] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998.
- [10] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*, 18(4):314–329, 1988.
- [11] Luc M. Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:583–598, 1991.

10 Appendix

10.1 Stakeholders and Their GDP Data Sources

Stakeholders	GDP Data Sources
Local Residents	https://fred.stlouisfed.org/series/PCPI24510 , https://fred.stlouisfed.org/series/MDBALT5POP
Freight Enterprises	https://fred.stlouisfed.org/series/GDPALL24510
Tourists	https://baltimore.org/about-us/

10.2 Improved K-Shell algorithm based on node weights and edge weights

```

1  def weighted_k_shell_with_node_weights(graph, node_weights):
2      weighted_degree = {node: sum(data['weight']) for _, data in
3          graph[node].items() for node in graph.nodes}
4      k_shell = {}
5      k = 0
6      alpha = 0.32534
7      while graph.number_of_nodes() > 0:
8          composite_score = {node: alpha * weighted_degree[node] + (1
9              - alpha) * node_weights[node] for node in graph.
10             nodes}
11         min_composite_score = min(composite_score.values())
12         to_remove = [node for node, score in composite_score.items
13             if score <= min_composite_score]
14         for node in to_remove:
15             k_shell[node] = k
16         for node in to_remove:
17             neighbors = list(graph.neighbors(node))
18             graph.remove_node(node)
19             del weighted_degree[node]
20             for neighbor in neighbors:
21                 if neighbor in graph.nodes:
22                     if neighbor in graph and node in graph[neighbor]:
23                         weighted_degree[neighbor] -= graph[neighbor][
24                             node].get('weight', 0)
25
26         k += 1
27
28     return k_shell

```