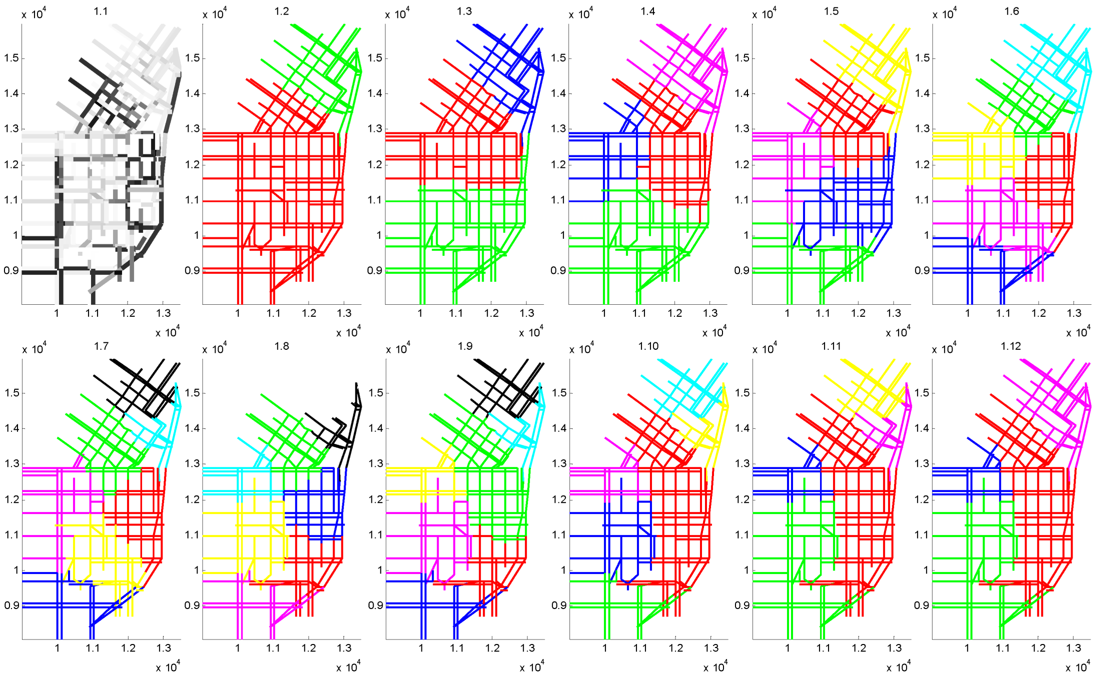

Spatial and Temporal Analysis of Congestion in Urban Transportation Networks

Yuxuan Ji

Nikolas Geroliminis

STRC 2010 **August 2010**

STRC 2010

Spatial and Temporal Analysis of Congestion in Urban Transportation Networks

Yuxuan Ji
Urban Transport Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
GC C2 399, Station 18, 1015 Lausanne, Switzerland
phone: +41-21-69-35397
fax: +41-21-69-35060
yuxuan.ji@epfl.ch

Nikolas Geroliminis
Urban Transport Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
GC C2 389, Station 18, 1015 Lausanne, Switzerland
phone: +41-21-69-32481
fax: +41-21-69-35060
nikolas.geroliminis@epfl.ch

August 2010

Abstract

It has been recently shown that a macroscopic fundamental diagram (MFD) linking space-mean flow, density and speed exists in the urban transportation networks under some conditions. An MFD is further well defined if the network is homogeneous with links of similar properties. However many real urban transportation networks are heterogeneous with different levels of congestion. The objective of this paper is to study the existence of MFD and the feasibility of simple control strategies to alleviate the congestion in the heterogeneous networks, which can be partitioned into homogeneous components. To achieve these goals, this paper focuses on the clustering of transportation networks based on the spatial and temporal features of congestion. A partitioning mechanism, which consists of three consecutive algorithms, is designed to minimize the variance of link densities while maintaining the spatial compactness of the clusters. Small variance of link densities within a cluster increases the aggregated flow for the same average density and spatial compactness makes feasible the application of perimeter control strategies. Firstly, Normalized Cut is applied to over segment the network into several clusters and a new metric is introduced to evaluate the partitioning results. Secondly, a merging algorithm is developed to improve the metric and total variance of link densities, and the optimal number of clusters is estimated and determined. Finally, a boundary adjustment algorithm is designed to further improve the metric and decrease the variances of the clusters while keeping the compactness of the shapes. Both the objectives of smaller variances and spatial compactness can be achieved after this partitioning mechanism. The simulation further demonstrates the superiority of our method in both effectiveness and robustness compared with other clustering algorithms.

Keywords

macroscopic fundamental diagram, network partitioning, simulation

1 Introduction

Analysis of traffic flow theory and modeling of vehicular congestion has mainly relied on fundamental laws, inspired from physics using analogies with fluid mechanics, many particles systems and the like. One main difference of physical systems and vehicular traffic is that humans make choices in terms of routes, destinations and driving behavior, which creates additional complexity to the system. While most of the traffic science theories make a clear distinction between free-flow and congested traffic states, empirical analysis of spatio-temporal congestion patterns has revealed additional complexity of traffic states and non-steady state conditions (see for example Muñoz and Daganzo (2003); Helbing *et al.* (2009)). Thus, the known fundamental diagram (initially observed for a stretch of highway and provide a steady-state relationship between speed, density and flow) is not sufficient to describe the additional complexity of traffic systems and it also contains significant experimental errors in the congested regime (see for example Kerner and Rehborn (1996) for a highway stretch or Geroliminis and Daganzo (2008) for a city street).

Nevertheless, it was recently observed from empirical data in Downtown Yokohama Geroliminis and Daganzo (2008) that by aggregating the highly scattered plots of flow vs. density from individual loop detectors, the scatter almost disappeared and a well-defined Macroscopic fundamental Diagram exists between space-mean flow and density.

The idea of an MFD with an optimum accumulation belongs to Godfrey (1969) but the verification of its existence with dynamic features is recent Geroliminis and Daganzo (2007, 2008). These papers showed, using a micro-simulation and a field experiment in downtown Yokohama, (i) that urban neighborhoods approximately exhibit a "Macroscopic Fundamental Diagram"(MFD) relating the number of vehicles to space-mean speed (or flow), (ii) there is a robust linear relation between the neighborhood's average flow and its total outflow (rate vehicles reach their destinations) and (iii) the MFD is a property of the network infrastructure and control and not of the demand, i.e. space-mean flow is maximum for the same value of vehicle density independently of time-dependent origin-destination tables. References Daganzo (2007); Geroliminis and Daganzo (2007) introduced simple control strategies to improve mobility in homogeneous city centers building on the concept of an MFD. The main logic of the strategies is that they try to decrease the inflow in regions with points in the decreased part of an MFD.

Despite these recent findings for the existence of MFDs with low scatter, these curves should not be a universal recipe. In particular, networks with an uneven and inconsistent distribution of congestion may exhibit traffic states that are well below the upper bound of an MFD and much too scattered to line along an MFD. By analyzing real data from a medium-size French city Buisson and Ladier (2009) showed that heterogeneity has a strong impact on the shape/scatter of an MFD. Recent findings from empirical and simulated data Geroliminis and Sun (2010); Mazloumian *et al.* (2010) have identified the spatial distribution of vehicle density in the network as one of the key components that affect the scatter of an MFD and its shape. They observed well defined relations between flow and density when link density variance is constant. In other words, the average network flow is consistently higher when link density variance is low, for the same network density.

These findings are of great importance because the concept of an MFD can be applied for heterogeneously loaded cities with multiple centers of congestion, if these cities can be partitioned in a small number of homogeneous clusters. The work presented in this paper creates clustering algorithms for heterogeneous transportation networks. Our goal is to partition a network into regions with small variances of link densities. This condition is also needed when simple perimeter control strategies are applied and each cluster is considered as a reservoir. If a cluster contains subregions with significantly different levels of congestion, the control strategies will be inefficient.

There has been a huge number of literatures on studying clustering algorithms and they generally fall into two large categories: hierarchical and partitional Jain (2010); Bishop (2007). Hierarchical approaches cluster data either in an agglomerative way in which each individual data point is an initial cluster or divisive way in which the whole data set is an initial one. For example, single linkage is a simple agglomerative algorithm which repeatedly merges the most similar pair of clusters until it reaches the desired result Day and Edelsbrunner (1984). Partitional approaches usually group the data points into a predetermined number of clusters based on an objective function. K-means is a such kind of algorithm which minimizes intra-cluster variance but can not guarantee a global optimal solution. A more complete and recent survey can be found in Jain (2010). Due to these efforts, clustering algorithms have been successfully applied in diverse fields such as data mining Han and Kamber (2006), image segmentation Shi and Malik (2000) and information retrieval Carmel *et al.* (2009).

However transportation networks have unique features and potential control strategies to alleviate traffic congestion will be designed based on the clustering results. Therefore an immediate application of an arbitrary clustering algorithm may not produce a desired solution. Here are several criteria that the developed clustering algorithms need to satisfy in transportation networks: (1) small variance of density values within each cluster, which is meant to guarantee a well defined MFD; (2) a small number of clusters, which can help design simple control strategies without a need for detailed origin-destination tables and route choice information; and (3) spatially near compact shapes of clusters, which can ease the design and deployment of effective controls. However these criteria can be conflicting for a real urban transportation network. For example, the first objective leads to a partitioning of maximum number of clusters, in which each link is a cluster itself and all the variances reach 0. The first one also conflicts with the third one as the objective of small variance is only for the density values (similar as intensity in image) while that of compact shapes is a spatial requirement. The region with even a small amount of noise in density values makes the two criteria incompatible. Designing a clustering mechanism that can achieve a good trade-off among these goals is our foremost task.

The remainder of this paper is organized as follows. Firstly, Normalized Cut algorithm is described which serves as the foundation of the partitioning for urban transportation networks. Secondly, the partitioning mechanism consisting of three consecutive steps of initial segmenting, merging and boundary adjustment is designed and explained in detail. Metrics are also introduced to evaluate the partitioning results and estimate the optimal number of clusters. Finally, simulation is conducted in a real network and comparisons are made with other algorithms to show both the effectiveness and robustness of the partitioning mechanism.

2 Normalized Cut Algorithm

Normalized Cut (Ncut) is a graph-based partitioning algorithm for image segmentation Shi and Malik (2000). Instead of focusing on local features or details, Ncut extracts the global impression of an image. Its principle is that "image partitioning is to be done from the big picture downward, rather like a painter first marking out the major areas and then filling in the details", while most of the previous works are based on local properties of the graph. In order to realize perceptual grouping, Ncut introduces the partitioning criterion and minimizes it by solving a generalized eigenvalue problem. The solved eigenvectors are then used to partition the image from global extractions to further details.

Suppose the node set V in a graph $G = (V, E)$ can be partitioned into two parts A and B. The total similarity between A and B can be expressed as $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$, where $w(u, v)$ denotes the similarity between two nodes u and v. The optimal partitioning objective is to minimize the value of $cut(A, B)$. However, this minimum criterion tends to cut a very small number of isolated nodes out of the graph. To avoid this biased partitioning, Ncut introduces the normalized criteria that are based on both the total dissimilarity between the different groups and the total similarity within the groups. The total disassociation ($Ncut$) between two groups and association ($Nassoc$) within each group are defined as follows:

$$Ncut(A, B) = \frac{cut(A, B)}{cut(A, V)} + \frac{cut(A, B)}{cut(B, V)} \tag{1}$$

$$Nassoc(A, B) = \frac{cut(A, A)}{cut(A, V)} + \frac{cut(B, B)}{cut(B, V)} \tag{2}$$

With the two unbiased cutting criteria, the partitioning with small number of isolated nodes will no longer be of minimum value. The two objectives of minimizing $Ncut(A,B)$ and maximizing $Nassoc(A,B)$ can be reached simultaneously since they simply keep the following relation:

$$Ncut(A,B) = 2 - Nassoc(A,B) \tag{3}$$

Minimizing Ncut value exactly is NP-complete, however the discrete solution can be approximated efficiently by solving an eigenvalue system in the real value domain. Let $x_i = 1$ if $x \in A$ and $x_i = -1$ if $x \in B$. Then the problem of minimizing Ncut value can be expressed in an exactly equivalent form:

$$\min_x Ncut(x) = \min_y \frac{y^T(D-W)y}{y^T D y} \tag{4}$$

under the conditions that $y(i) \in \{1, -b\}$ and $y^T D1 = 0$, where $y = (1+x) - b(1-x)$, D is a $N \times N$ diagonal matrix with $d(i) = \sum_j w(i,j)$ on its diagonal and $W(i,j) = w_{ij}$. Solving a standard eigenvalue system takes $O(n^3)$ operations where $n = |V|$. With special properties of the graph partitioning problem, it usually can be decreased to $O(n^{3/2})$ for an image.

However, this may not be the ultimate solution of the original problem when y is allowed to have only discrete values. So the real value solution is transformed to a discrete one by checking the splitting points in the solution vectors and the one that gives the best Ncut Value is chosen. This method of splitting points is higly reliable.

The Ncut partitioning process is briefly outlined below:

1. Given a graph, set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.
2. Solve the equivalent eigenvalue system and get the smallest eigenvalues.
3. Pick up the eigenvector with the second smallest eigenvalue, discreticize it by checking splitting points and bipartition the graph.
4. Continue to partition each subgraph if needed.

Since Ncut is a graph based partitioning algorithm, it could be directly applied to the transportation network which can be modeled as a graph. We model each street as a node and build their neighboring relationships based on their spatial connections. The density of each street is similar as the intensity value in an image. The reasons that Ncut algorithm is considered in the transportation network are: (1) Ncut is a graph based segmentation algorithm; (2) Ncut avoids biased partitioning which cuts out isolated nodes and produces a balanced results; (3) Ncut realizes perceptual grouping and extracts global impressions (major or obvious parts) from the graph or image, which is the most important characteristic of this algorithm; (4) it can produce spatially compact clusters with proper control of parameters; and (5) it is computationally efficient.

These unique features of Ncut comply very well with the second and third requirements in our aforementioned partitioning criteria. Our simulation results in the San Francisco transportation network show that Ncut algorithm can always produce near compact clusters either when we set a higher weight to the spatial distance than density difference or set a low threshold value to the spatial distance measure in similarity function.

Although Ncut algorithm cuts the major components with compact shapes out of the graph as expected, its partitioning result may not be able to satisfy our first objective of minimizing the variance of density values within each cluster. As we have discussed before, the first criterion conflicts with the other two. A restatement of our criteria is that we want to achieve the smallest possible variance within all the compact clusters, given that the second and third criteria are satisfied. Therefore, Ncut can provide us with a good initial partitioning, but it does not necessarily produce the desired optimal results. Furthermore, as pointed out by Timothee in Cour *et al.* (2005), Ncut tends to cut a large uniform region into two if the spatial distance threshold is too low. However, higher spatial threshold value is very likely to produce uncompact clusters. These problems make the Ncut algorithm insufficient and ineffective based on our criteria when it is applied to the transportation network, and therefore further modifications and refinements are needed.

3 Methodology

Our main objective is to partition a transportation network into homogeneous components based on the properties of a well-defined MFD. More specifically, we seek to develop a mechanism of partitioning which can achieve the following goals: (1) minimize the variance of link densities in each cluster to guarantee a well-defined MFD; (2) extracts a small number of main components from the network at a global level ignoring details and local features; (3) produce clusters that are spatially near compact without weird shapes to facilitate effective traffic management strategies. Alternatively, our partitioning criterion is to minimize the variance of link densities within each cluster under the constraints that the number of final clusters is small and they are spatially compact.

Based on these goals we design a partitioning mechanism which consists of three consecutive algorithms. This mechanism can not only solve the problem of Ncut cutting uniform regions, produce spatially compact clusters, but also reach the minimum variances of density values within each compact cluster. Firstly, Ncut provides an over segmenting of the transportation network. This step can help to extract the major components from the network and guarantee spatially compact shapes under a low threshold value for spatial distance measure. Besides, a metric is defined to evaluate the partitioning result in this step. Secondly, we recursively merge a pair of closest clusters simply based on the mean values of their densities until a desired number of clusters is reached. This step solves the problem of Ncut cutting large uniform regions by small threshold values and hence improves the metric value defined in the first step. After these two steps, a partitioning of the network with spatially compact shapes and optimal number of clusters is produced. Finally, we minimize the variance within each cluster by repeatedly adjusting the boundaries between each pair of clusters. The last step seeks to minimize the within cluster density variance. However, the final objective should be achieved while keeping the spatial compactness of the shapes of the clusters provided by the previous two steps. Each step is explained in detail as follows.

3.1 Initial Segmenting and Evaluation

In the first step, we apply Ncut algorithm in the transportation network to produce several initial partitioning solutions with different numbers of clusters. We build the transportation network as an undirected graph G. Each node i in G represents a link in the network and has a density value $d(i)$ of the link at a certain time during a day (time t is omitted from the equation). The spatial distance between two links is denoted by the length of the shortest path $r(i,j)$ between node i and j in G. In order to obtain spatially compact clusters, we set the spatial distance threshold value to be 1 and define the similarity function between link i and j as follows:

$$w(i,j) = \begin{cases} \exp(-(d(i,j))^2), & r(i,j) = 1 \\ 0, & r(i,j) > 1. \end{cases} \tag{5}$$

Based on this definition, each cluster will always have a group of spatially connected links.

In order to estimate the optimal number of clusters and evaluate a given partitioning with k clusters, a metric 'NcutSilhouette' (NS) is defined as follows:

$$NS_k(A,B) = \frac{\sum_{i\in A}\sum_{j\in B}(d_i - d_j)^2}{N_A * N_B}, \tag{6}$$

where k is the number of clusters and N_A is the total number of links in cluster A. Since Ncut always meet our criterion of spatially compact shapes and the ultimate goal is to minimize the variance of the density values, NS_k does not contain any spatial information and only measures the average density distance between cluster A and B. Furthermore, we can evaluate whether the links of a cluster A are properly grouped by the following metric:

$$NS_k(A) = \frac{NS_k(A, A)}{NS_k(A, B)}, \text{where } NS_k(A, B) = \min\{NS_k(A, K)|K \in Neighbor(A)\}. \tag{7}$$

In this metric, $NS_k(A, A)$ measures the intra-cluster similarity of densities while $NS_k(A, B)$ measures the inter-cluster similarity. If two clusters are not spatially connected, it can also be a good partitioning even if their link densities are the same. Therefore, we only measure the inter-cluster similarity of cluster A with its neighbors $Neighbor(A)$. Since A may have several neighbors, it is proper to use the one that is most similar with A (i.e., in the worst case) to evaluate the inter-cluster similarity, as defined in Eq. (7). Therefore cluster A is properly partitioned if $NS_k(A) < 1$. The overall partitioning can be evaluated by the average NS_k value of all the clusters in a given partitioning:

$$NS_k = \frac{\sum_{A \in C} NS_k(A)}{k}, \tag{8}$$

where C is the set of clusters and k is the total number of clusters.

The NS value of a partitioning should be the smaller one which satisfies the spatial constraints. For example, suppose there are three links i, j and l with neighboring relations $r(i, j) = r(j, l) = 1$ and $r(i, l) = 0$, and densities $d(i) = d(l) = 1$ and $d(j) = 0$. The optimal NS value will be obtained if $A = \{i, l\}$ and $B = \{j\}$. However, this is not a feasible solution since link i and l are not connected.

The NS metric can be equivalently expressed by the variances and means of the link densities in the clusters as follows:

$$NS_k(A, B) = Var(A) + Var(B) + (u_A - u_B)^2, \tag{9}$$

where $Var(A)$ and u_A are the variance and mean of the link densities in cluster A. The proof is straightforward.

$$
\begin{aligned}
NS_k(A,B) &= \frac{\sum_{i\in A}\sum_{j\in B}(d_i-d_j)^2}{N_A N_B} \\
&= \frac{\sum_{j\in B}\sum_{i\in A} d_i^2 + \sum_{i\in A}\sum_{j\in B} d_j^2 - 2\sum_{i\in A}\sum_{j\in B} d_i d_j}{N_A N_B} \\
&= \frac{N_B\sum_{i\in A} d_i^2 + N_A\sum_{j\in B} d_j^2 - 2N_A N_B u_A u_B}{N_A N_B} \\
&= \frac{N_A N_B(\frac{\sum_{i\in A} d_i^2}{N_A} - u_A^2) + N_A N_B(\frac{\sum_{j\in B} d_j^2}{N_B} - u_B^2) - 2N_A N_B u_A u_B + N_A N_B(u_A^2+u_B^2)}{N_A N_B} \\
&= \frac{N_A N_B Var(A) + N_A N_B Var(B) - 2N_A N_B u_A u_B + N_A N_B(u_A^2+u_B^2)}{N_A N_B} \\
&= Var(A) + Var(B) + (u_A - u_B)^2.
\end{aligned}
\tag{10}
$$

Hence we get:

$$
NS_k(A) = \frac{NS_k(A,A)}{NS_k(A,B)} = \frac{2Var(A)}{Var(A)+Var(B)+(u_A-u_B)^2}. \tag{11}
$$

Based on Eq. (11), we observe that when the difference of means is large and the variances are relatively small, NS value will be small which means a well partitioned cluster A. When both the difference of means and the variances are small which implies two similar clusters, the NS value will be close to 1. More generally, for a cluster A with small variance, it is properly partitioned since $Var(A) < Var(B) \Rightarrow NS(A) < 1$. For a cluster A with larger variance, partitioning is not optimal, which means that further partitioning or merging with other clusters is needed, unless the difference of means with its most similar neighbor is big enough to compensate for the the difference of their variances. This implies $(u_A - u_B)^2 > Var(A) - Var(B)$. However, due to the fact that a cluster with smaller variance and NS value is probably accompanied by a neighbor cluster of larger variance and NS, the overall partitioning is evaluated by the average NS value of all the clusters. Therefore, even if there are a few clusters with NS values larger than 1, we can still get a proper partitioning if there are many well partitioned clusters with small NS values.

3.2 Merging

After completing the first step, we have several initial partitioning with different numbers of clusters. We also evaluate the clustering results based on the NS metric and get the optimal number of clusters generated by Ncut. However, the initial partitioning by Ncut is not necessarily an optimal solution, since Ncut tends to cut uniform region into two if the spatial distance threshold is too low. Therefore, in the second step, we use a merging algorithm to form a series of new clusters based on the initial clusters given by Ncut. The merging algorithm is straightforward. Each time, we merge two clusters with the closest means of density values, until we reach only one cluster. Then we use NS to estimate the optimal number of clusters after merging.

The merging algorithm is similar as the agglomerative clustering algorithm. However, this merging process based on Ncut has two significant improvements from directly applying an agglomerative algorithm to the original graph. Firstly, the computation is more efficient. The merging algorithm costs $O(k^3)$ where k is the initial number of over segmented clusters by Ncut. Since usually $k \ll n$ where n is the total number of nodes in the graph, the overall computational cost is simply $O(n^{\frac{3}{2}})$ from Ncut. As for the agglomerative algorithm, it costs $O(n^3)$. Secondly, this Ncut-based merging process can still produce near compact clusters when only density values are taken into account, while it will be complicated and hard for the agglomerative clustering algorithm to achieve this even if both spatial and density information is used.

We compare the NS values of these new clusters after merging with those of the original ones given by Ncut. The total variance of the clusters for each partitioning is also calculated and compared as follows:

$$\text{Total variance} = \sum\nolimits_{A \in C} N_A * Var(A). \tag{12}$$

The experiments show that the merging algorithm improves both NS and total variance. However, note that the NS value is used as a metric to estimate the optimal number of clusters while the total variance can not since the latter one prefers large number of clusters and does not consider the inter-class similarity at all.

3.3 Boundary Adjustment

By Ncut and merging, the major components (or global perceptual grouping) have been obtained from the network with spatially near compact shapes, which means that our second and third criteria of partitioning the transportation network have been satisfied. Besides, it is obvious that both Ncut and merging are also aiming at decreasing the variance of the density values within each cluster while producing the partitioning. Based on the NS metric, we can choose the optimal number of clusters after the second step of merging. However, the criterion of small variances of link densities can be further reached if we apply boundary adjustment. This step is similar as refining the edges of a rough sketch to make it more distinct and clearer.

There are mainly two reasons of applying the boundary adjustment algorithm. Firstly, the links on the boundaries of the two clusters are most likely unstable, which means that by changing their belongings to a neighboring cluster, the objective values of the partitioning may not be significantly affected. Secondly, the Ncut algorithm favors balanced partitioning, instead of minimum variance within each cluster. Therefore, adjusting the links on the boundaries can possibly improve the first objective of small variances in density values. Furthermore, since we do not have a strictly quantitative constraint for balancing or compactness, a proper boundary adjustment algorithm can help us further decrease the variances of densities without violating the other two criteria.

We now introduce a straightforward boundary adjustment algorithm. Firstly, we identify all the boundary links $i \in Boun$. Suppose $i \in B$ and $i \in Boun(A, B)$. Secondly, we move each link i independently from its current cluster B to its neighbor cluster A and calculate the change of the variance of link densities in cluster A and B. Finally, we choose the link i that decreases both the variances in A and B to the most extent, and update the clusters. The whole process is repeated until no link on boundaries can decrease both variances. The criterion of decreasing both density variances of clusters A and B are met when:

$$\begin{cases} \dfrac{(d_i - u_A)^2}{Var(A)} < \dfrac{N_A + 1}{N_A} \\ \dfrac{(d_i - u_{B-i})^2}{Var(B-i)} > \dfrac{N_B}{N_B - 1} \end{cases}, \tag{13}$$

where $B - i$ denotes the set of all the other links from B except i. When the number of links in a cluster is large enough, the right side of the inequality is close to 1, which implies that if the distance from link i to the center of cluster A is smaller than the average distance of links within A to the center, adding link i to the cluster A will decrease the variance of A. A more general result and the proof for adjusting a group of links on the boundaries are provided later.

The criteria of choosing a link in the boundary adjustment algorithm can be different. Preferably, we choose the one that decreases the total variance as a whole, although it may decrease the variance on one side and increase it on the other side. However, if only one link is adjusted each time, the final spatial shapes of the clusters will become uncompact and links in the same clusters are very likely to be disconnected. Therefore, we propose to adjust *a group of spatially consecutive links* on the boundaries to keep the compactness of the cluster shapes. Suppose we move a group of links Y from cluster B to A where $Y \subset B$, $Y \subset Boun(A, B)$ and $y_i \in Y$, the variances of link densities in both A and B will be decreased if the following two conditions in

(14) hold:

$$\begin{cases} \dfrac{(u_A - u_Y)^2}{\mathrm{var}(A) - \mathrm{var}(Y)} < \dfrac{N_A + N_Y}{N_A} \\ \dfrac{(u_{B-Y} - u_Y)^2}{\mathrm{var}(B-Y) - \mathrm{var}(Y)} > \dfrac{N_{B-Y} + N_Y}{N_{B-Y}} \end{cases}. \tag{14}$$

The proof is straightforward.

$$\begin{aligned} &Var(A) - Var(A') \\ &= \left[\frac{\sum_{x_i \in A} x_i^2}{N_A} - \left(\frac{\sum_{x_i \in A} x_i}{N_A} \right)^2 \right] - \left[\frac{\sum_{x_i \in A} x_i^2 + \sum_{y_i \in Y} y_i^2}{N_A + N_Y} - \left(\frac{\sum_{x_i \in A} x_i + \sum_{y_i \in Y} y_i}{N_A + N_Y} \right)^2 \right]. \end{aligned} \tag{15}$$

After some manipulations we obtain:

$$\begin{aligned} &Var(A) - Var(A') \\ &= \frac{N_A N_Y [Var(A) - Var(Y)] + N_Y^2 [Var(A) - Var(Y)] - N_A N_Y (u_A - u_Y)^2}{(N_A + N_Y)^2}. \end{aligned} \tag{16}$$

Let the numerator > 0, so we easily get $(u_A - u_Y)^2/[Var(A) - Var(Y)] < (N_A + N_Y)/N_A$. The condition of variance decrease for cluster B is obtained similarly.

In the end we summarize our boundary adjustment algorithm as follows:

1. For each cluster, find all the links on the boundaries and build a sequence for each boundary based on their spatial neighboring relations.
2. For each boundary sequence, find a subgroup of consecutive links that decreases the total variance most after moving them to the neighboring cluster, under the constraints of an upper bound and lower bound for the length of the subgroup. If no such subgroup is found, the algorithm stops.
3. Choose the subgroup that decreases the total variance most among all the boundary sequences, and move it to the new cluster and finally update the partitioning.
4. Continue to step 1.

The computational cost of the boundary adjustment algorithm mainly comes from the second step of finding the optimal subgroup of consecutive links that can decrease the total variance most. With both an upper bound and lower bound constraints on the length of the subgroups, it takes $O(s^2)$ to test all the possible subgroups, where s is the number of links on the boundary. However, if the boundary is long which means s is not significantly smaller than n, the computational cost of the boundary adjustment algorithm will be $O(n^3)$ (variance calculation costs $O(n)$). Improving the efficiency of this algorithm is one of our future tasks. Note that the second step will cost $O(s)$ if there is only an upper bound, and $O(s \log L)$ if there is only lower bound where L is the lower bound of the length Lin *et al.* (2002).

4 Simulation

In this section, we further demonstrate the effectiveness of our partitioning mechanism by simulation. We show how the optimal NS metric and total variance improve in each step of the partitioning for a real transportation network. Results in different time periods during a day are given and discussed. Furthermore, we compare with k-means clustering algorithm and show the superiority of our method in both effectiveness and robustness.

4.1 Network Description

This test site is a 2.5 square mile area of Downtown San Francisco (Financial District and South Of Market Area), including about 100 intersections with link lengths varying from 400 to 1300 feet. The number of lanes for through traffic varies from 2 to 5 lanes and the free flow speed is 30 miles per hour. Traffic signals are all multiphase fixed-time operating on a common cycle length of 100 seconds for the west boundary of the area (The Embarcadero) and 60 seconds for the rest.

4.2 Partitioning Results

We discuss the partitioning results for typical time periods during a day with different congestion levels. We mainly analyze the effectiveness of our mechanism for a semi-congested network at time $t = 70$ when a group of congested links has formed but the network flow is still high. The original network with link density values at time $t = 70$ is shown in gray-scale level in Figure 1.1, where light color means low density link while dark is a jammed link. Figures 1.2-1.8 are the partitioning results by Ncut with number of clusters from 2 to 8. Figure 1.3. is the optimal one determined by NS. In the second step, Figures 1.9-1.14 show the merging process from 8 to 2 clusters and the optimal one is Figure 1.13. After the first two steps, we get an optimal partitioning with three spatially compact clusters of the network in Figure 1.13. In order to further improve NS metric and total variance of link densities, boundary adjustment is implemented in the last step and the final partitioning is shown in Figure 1.15.

Accordingly, Table 1 explains the metric values. Table 1.1 shows NS values defined in Eq. (8) by Ncut partitioning with different numbers of clusters. The optimal number of clusters estimated by NS is 3. Table 1.2 shows the NS values after merging from 8 initial clusters by

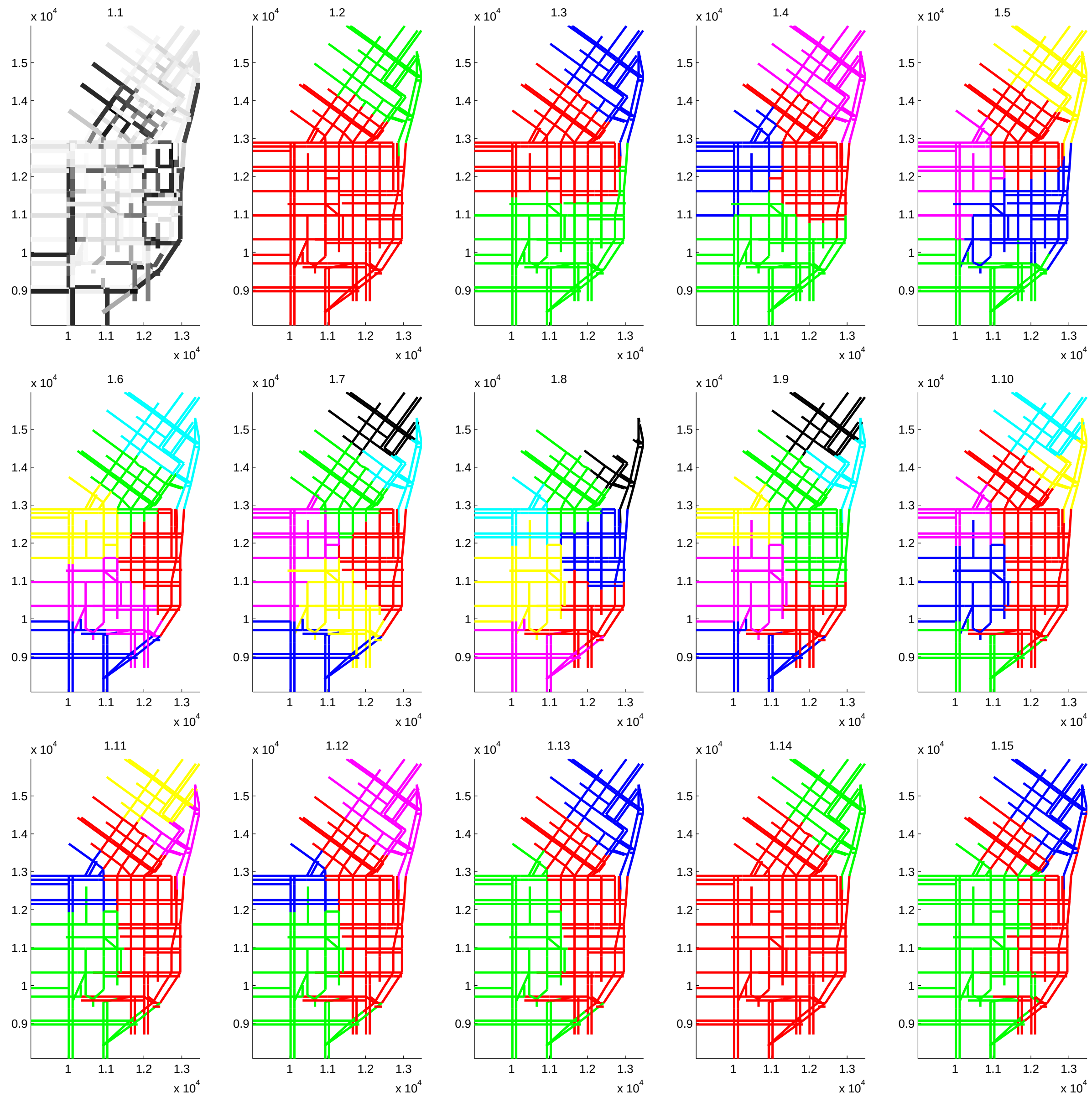

Figure 1: Partitioning at $t = 70$ by Ncut (1.2-1.8), merging (1.9-1.14) and boundary adjustment(1.15)

Ncut. The optimal number of clusters 3 is still obtained, but the NS value is smaller than the optimal one by Ncut.

Next we explain how the partitioning improves by comparing the NS metric, cluster variance and mean difference in each step (the units for variance and mean are $(veh/m)^2 \times 10^{-3}$ and veh/m). Table 1.3 shows the NS value and total variance by Eq. (12) of the optimal partitioning produced in each step. Both of the two metrics for merging and finally boundary adjustment decrease when compared with Ncut. Since the variance of the original network with one cluster is 0.1348, the total variance decreases by around 20% in the end. Table 1.4 examines the variance and NS for each cluster in more detail. The variance of the red cluster is increased by 13% from Ncut to the final result; green decreased by 35%; and the blue decreased by 63%.

Table 1: NS metric, variance and mean

TABLE 1.1 Average NS by Ncut at $t = 70$

# of clusters	2	3	4	5	6	7	8
Average NS	0.8117	0.7442	0.7718	0.8715	0.8363	1.0167	0.9373

TABLE 1.2 Average NS after merging at $t = 70$

# of clusters	8	7	6	5	4	3	2
Average NS	0.9373	0.9390	0.9124	0.9578	0.7802	0.6865	0.7301

TABLE 1.3 Total variance of link densities and average NS at $t = 70$

	Ncut	Merging	Bo. Adj.
Total variance	0.1249	0.1212	0.1069
Average NS	0.7442	0.6865	0.5402

TABLE 1.4 Variance and NS within each cluster at $t = 70$

Variance/NS	Red	Green	Blue
Ncut	0.4091/1.0022	0.4147/0.9885	0.0766/0.2419
Merging	0.4451/1.0790	0.3303/0.8007	0.0696/0.1799
Bo. Adj.	0.4623/0.9876	0.2696/0.5759	0.0284/0.0572

TABLE 1.5 Mean of link densities within each cluster at $t = 70$

Mean / # of links	Red	Green	Blue
Ncut	0.0217 / 156	0.0197 / 133	0.0078 / 77
Merging	0.0236 / 175	0.0166 / 115	0.0075 / 76
Bo. Adj.	0.0286 / 139	0.0143 / 150	0.0062 / 77

TABLE 1.6 Average mean difference of each partitioning at $t = 70$

	Ncut	Merging	Bo. Adj.
Ave. mean difference	0.00795	0.01155	0.01835

As for the NS metric, it decreases in all cases. To further show the improvement, Table 1.5 gives the mean of link densities and number of links in each cluster, and Table 1.6 calculates the average mean difference of the neighboring clusters in each partitioning as follows:

$$\frac{\sum\sum_{A\subset C,B\subset C}|Mean(A)-Mean(B)|*D(A,B)}{\sum_{A\subset C,B\subset C}D(A,B)}, \tag{17}$$

where C is the set of clusters, $Mean(A)$ is the mean of link densities in cluster A and $D(A,B)=1$ if cluster A and B are neighbors ($D(A,B)=0$, otherwise). The mean difference is increased by a large degree from the original Ncut to the final partitioning after boundary adjustment.

Finally we present the histograms of the frequency of link densities in each cluster in Figure 2 (x-density, y-frequency). Figure 2.1-2.3 show the histogram of frequency of link densities

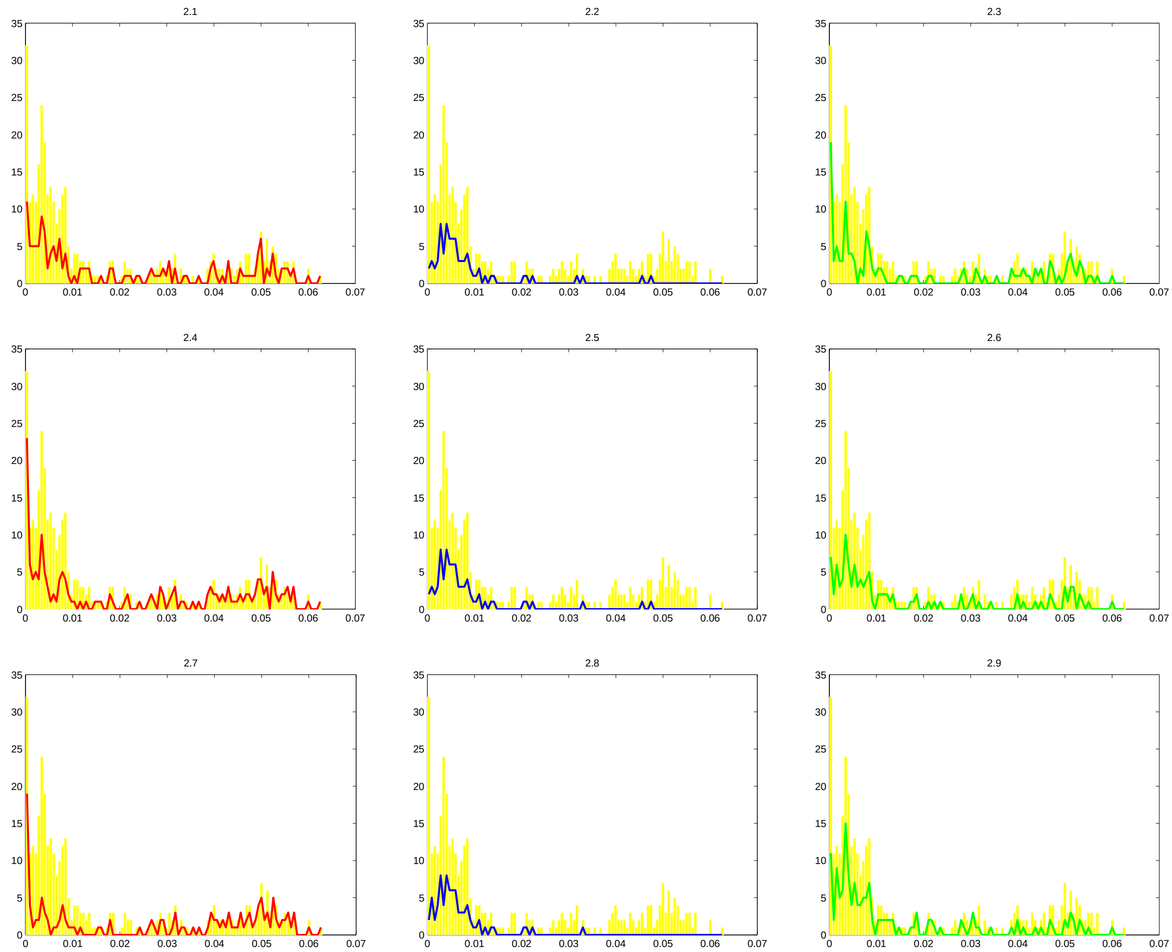

Figure 2: Histograms of link densities at $t = 70$ (2.1-2.3 by Ncut, 2.4-2.6 after merging, 2.7-2.9 after boundary adjustment)

in each cluster by Ncut (e.g., Figure 2.1 describes the frequency of link densities in the red cluster). Similarly, Figure 2.4-2.6 show the histograms after merging and Figure 2.7-2.9 after boundary adjustment. Note that after Boundary Adjustment, the red cluster (with the maximum mean value among the three clusters) contains fewer low-density but more high-density links by comparing Ncut in Figure 2.1 with boundary adjustment in Figure 2.7. The other two clusters are similarly analyzed. However since the spatial information is not included in the histograms, it is very unlikely to obtain completely separate distributions of link densities.

The above analysis demonstrates a significant improvement of the partitioning mechanism compared to the original Ncut in urban transportation networks based on our criteria.

The time periods around $t = 70$ have very similar pattern. We now take a look at some other periods of a day when different patterns may occur at $t = 40$, $t = 75$, $t = 80$.

The network density At $t = 80$ (more congested network) is shown in Figure 3.1 and the optimal number of clusters estimated by NS is still three for both original Ncut in Figure 3.2 and merging in Figure 3.3, and the final boundary adjustment is in Figure 3.4. Main metrics

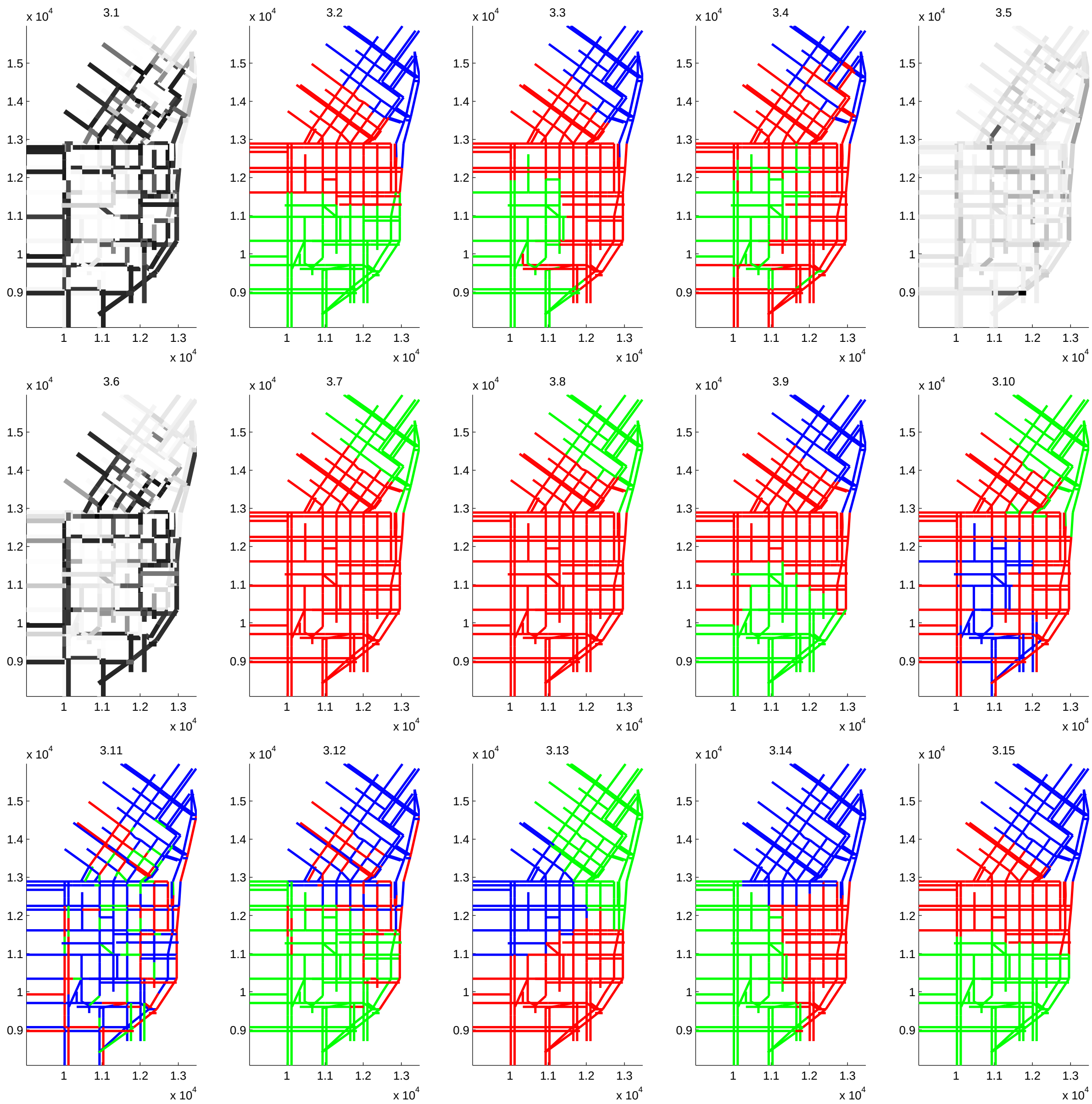

Figure 3: Network density and partitioning at $t = 80$ (3.1-3.4); density at $t = 40$ (3.5); density and partitioning at $t = 75$ (3.6-3.10); and k-means clustering at $t = 70$ (3.11-3.15)

are given in Table 2.1. The total variance of the original network at $t = 80$ is 0.1988 and decreases by 20% after partitioning. From the geographical presentations of the clustering at $t = 70$ and $t = 80$, the congestion propagation can be easily and clearly identified by comparing Figure 1.15 and 3.4. The network density at $t = 75$ is shown in Figure 3.6, and two different partitioning with 2 (merging in Figure 3.7 and boundary adjustment in Figure 3.8) and 3 clusters (Figure 3.9 and Figure 3.10) are given. Table 2.2 compares the metrics, and it is shown that the NS value of 3 clusters is worse than the one of 2 custers after merging, but significantly better after boundary adjustment. This observation demonstrates the effectiveness of the boundary adjustment algorithm, but also suggests more consideration on the merging algorithm, which is currently simple but highly efficient. Improving the merging process based on both spatial and temporal features is one of our future tasks. Lastly, the network density

Table 2: NS metric and variance

TABLE 2.1 Total variance and average NS at $t = 80$

	Ncut	Merging	Bo. Adj.
Total var.	0.1926	0.1892	0.1587
Average NS	0.9010	0.8629	0.5130

TABLE 2.2 Total variance and average NS at $t = 75$

2 clusters	Ncut	Merging	Bo. Adj.	3 cluster	Ncut	Merging	Bo. Adj.
Total var.	0.1634	0.1567	0.1553		0.1587	0.1551	0.1185
Ave. NS	0.8378	0.6519	0.6283		0.7773	0.7019	0.3633

TABLE 2.3 Total variance and average NS by k-means at $t = 70$

k-means	Fig. 2.11	Fig. 2.12	Fig. 2.13	Fig. 2.14	Fig. 2.15	
Total var.	0.0073	0.0530	0.1339	0.1287	0.1284	
Ave. NS	0.0977	0.6208	1.0295	1.0218	0.8278	

at $t = 40$ is given by Figure 3.5, which shows a uniformly uncongested network with similar link densities. We observe that after the network is partitioned into several components with spatially compact shapes, the total variance is not decreased significantly, which implies the homogeneity of the link densities. Therefore we conclude that the network at this time period does not need partitioning.

4.3 Comparison with k-means

In previous section, we show the improvement of the partitioning mechanism from original Ncut. Now we examine the superiority of this mechanism by comparing with the clustering algorithm of k-means widely used in the field. In order to show the difference, we analyze a partitionable network at time $t = 70$.

K-means algorithm randomly chooses k samples from the set to be clustered as the initial centers and assigns each of the samples to its nearest center. Then it recalculates the center of each cluster (usually by mean value) and repeats the assigning process until the assignment do not change (i.e., the clusters are stable). In k-means algorithm, feature vector is used to measure the similarity and make clusters. Therefore we include both spatial (as $x - y$ coordinates) and density information of the links in the vector as $(x, y, d)^T$ with d denoting the density value, and assign different weights w_s and w_d to them by $(w_s * x, w_s * y, w_d * d)^T$. Figure 3.11$\sim$3.15 show different partitioning results by k-means with $k = 3$ and Table 2.3 gives the corresponding metric values for each partitioning. For instance, Figure 3.11 is a partitioning with $w_s/w_d = 1$. In this case, the NS metric and total variance are very low. However, this partitioning is meaningless since there is no spatial compactness at all, which also explains the high conflicts between spatial and density criteria. Figure 3.12 is the case when $w_s/w_d = 4$. Spatial compactness exists to some extent but some links are still highly disconnected. Figure 3.13$\sim$3.15 show three different partitioning when $w_s/w_d = 9$. When the spatial feature receives higher weight, compactness can usually but not always be guaranteed by k-means. However, the partitioning is very unstable due to the local optimality. In addition, even if when k-means can generate spatially compact clusters, our partitioning method still outweighs k-means in both NS metric measuring the partitioning quality and the total variance measuring the closeness to the ultimate goal, as compared from Table 2.3 and Table 1.3.

K-means is not suitable in partitioning the transportation network for two main reasons. Firstly, the clustering result depends on the choice of the initial k centers. Therefore it is unstable and often reaches the local optimality. Secondly, k-means algorithm is based on cluster centers (means), which can not be easily or appropriately realized in a graph-based network. In similarity function, we measure the spatial distance of two links by the length of their shortest path instead of Euclidean distance, and do not calculate the center of a cluster. However in k-means, we have to build a feature vector for each of the link. Spatial coordinates are often used as two features but can not guarantee the connectivity of links in the final clusters. In addition, the links in transportation networks should be grouped based on their neighborhood and connectivity, instead of their physical distance or lengths of links. If we still want to apply the same graph-based distance in k-means algorithm, it will be very hard to build the feature vectors and calculate the center of a cluster.

5 Conclusions and Future Work

Traffic congestion is increasing in urban cities. Improving traffic mobility and congestion has always been on the top agenda in both academics and industries. In this paper, in order to further study the existence of MFD and traffic control from a macroscopic level, a partitioning mechanism based on the criteria of a well defined MFD in the urban transportation networks is designed, which consists of three consecutive algorithms: initial segmenting, merging and boundary adjustment. This mechanism can produce a partitioning with an optimal number of clusters that have both minimum variances and spatially compact shapes. Furthermore, by comparing with Ncut and k-means clustering algorithms by simulation, our mechanism demonstrates the superiority of both effectiveness and robustness in partitioning a real urban transportation network. Our work in this paper has laid a solid foundation for the future research on designing practical control policies to realize effective congestion alleviation in the urban transportation systems. In the future work, we will continue to study the traffic propagation by exploring the spatial and temporal features of congestion and their correlations. Based on these findings, we will design control strategies for the heterogeneous network with different levels of congestion.

References

Bishop, C. M. (2007) *Pattern Recognition and Machine Learning*, Springer.

Buisson, C. and C. Ladier (2009) Exploring the impact of homogeneity of traffic measurements on the existence of macroscopic fundamental diagrams, *Transportation Research Record*, **2124**, 127–136.

Carmel, D., H. Roitman and N. Zwerdling (2009) Enhancing cluster labeling using wikipedia, paper presented at *SIGIR '09*, 139–146, Boston, USA.

Cour, T., F. Benezit and J. Shi (2005) Spectral segmentation with multiscale graph decomposition, paper presented at *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1124–1131, San Diego, CA, USA.

Daganzo, C. F. (2007) Urban gridlock: Macroscopic modeling and mitigation approaches, *Transportation Research part B*, **41** (1) 49–62.

Day, W. H. E. and H. Edelsbrunner (1984) Efficient algorithms for agglomerative hierarchical clustering methods, *Journal of Classification*, **1**, 1–24.

Geroliminis, N. and C. F. Daganzo (2007) Macroscopic modeling of traffic in cities, paper presented at *86th Annual Meeting of the Transportation Research Board*, Washington, DC. Paper No. 07-0413.

Geroliminis, N. and C. F. Daganzo (2008) Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings, *Transportation Research Part B-Methodological*, **42** (9) 759–770.

Geroliminis, N. and J. Sun (2010) Properties of a well-defined macroscopic fundamental diagram, paper presented at *Transportation Research Board 89th Annual Meeting*, Washington, DC. Paper No. 10-3521.

Godfrey, J. W. (1969) The mechanism of a road network, *Traffic Engineering and Control*, **11** (7) 323–327.

Han, J. and M. Kamber (2006) *Data Mining: Concepts and Techniques*, Morgan Kaufmann.

Helbing, D., M. Treiber, A. Kesting and M. Schönhof (2009) Theoretical vs. empirical classification and prediction of congested traffic states, *European Physical Journal B*, **69** (4) 583–598.

Jain, A. K. (2010) Data clustering: 50 years beyond k-means, paper presented at *19th International Conference in Pattern Recognition (ICPR)*, vol. 31, 651–666, Tampa, Florida, USA.

Kerner, B. S. and H. Rehborn (1996) Experimental properties of complexity in traffic flow, *Physical Review E*, **53** (5) 4275–R4278.

Lin, Y. L., T. Jiang and K. M. Chao (2002) Efficient algorithms for locating the length-constrained heaviest segments, with applications to biomolecular sequence analysis, *Journal of Computer and System Sciences*, **65** (3) 570–586.

Mazloumian, A., N. Geroliminis and D. Helbing (2010) The spatial variability of vehicle densities as determinant of urban network capacity, *Philosophical Transactions of Royal Society A*. in press.

Muñoz, J. C. and C. F. Daganzo (2003) Structure of the transition zone behind freeway queues, *Transportation Science*, **37** (3) 312–329.

Shi, J. and J. Malik (2000) Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (8) 888–905.