Transportation Research Part B 46 (2012) 1639–1656

Contents lists available at SciVerse ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

On the spatial partitioning of urban transportation networks

Yuxuan Ji, Nikolas Geroliminis *

École Polytechnique Fédérale de Lausanne (EPFL), School of Architecture, Civil and Environmental Engineering (ENAC), Urban Transport Systems Laboratory (LUTS), Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 23 November 2011
Received in revised form 3 August 2012
Accepted 3 August 2012

Keywords:
Macroscopic fundamental diagram
Traffic flow
Partitioning
Urban mobility

ABSTRACT

It has been recently shown that a macroscopic fundamental diagram (MFD) linking space-mean network 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. This collective behavior concept can also be utilized to introduce simple control strategies to improve mobility in homogeneous city centers without the need for details in individual links. However many real urban transportation networks are heterogeneous with different levels of congestion. In order to study the existence of MFD and the feasibility of simple control strategies to improve network performance in heterogeneously congested networks, this paper focuses on the clustering of transportation networks based on the spatial features of congestion during a specific time period. Insights are provided on how to extend this framework in the dynamic case. The objectives of partitioning are to obtain (i) small variance of link densities within a cluster which increases the network flow for the same average density and (ii) spatial compactness of each cluster which makes feasible the application of perimeter control strategies. Therefore, 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. Firstly, an over segmenting of the network is provided by a sophisticated algorithm (Normalized Cut). Secondly, a merging algorithm is developed based on initial segmenting and a rough partitioning of the network is obtained. Finally, a boundary adjustment algorithm is designed to further improve the quality of partitioning by decreasing the variance of link densities while keeping the spatial compactness of the clusters. In addition, both density variance and shape smoothness metrics are introduced to identify the desired number of clusters and evaluate the partitioning results. These results show that both the objectives of small variance and spatial compactness can be achieved with this partitioning mechanism. A simulation in a real urban transportation network further demonstrates the superiority of the proposed method in effectiveness and robustness compared with other clustering algorithms.

1. Introduction

Analysis of traffic flow theory and modeling of vehicular congestion have mainly relied on fundamental laws, inspired from physics using analogies with fluid mechanics, queuing theory, many particles systems and the like. One main difference of other 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 distinc-

* Corresponding author. Address: GC C2 389, Station 18, 1015 Lausanne, Switzerland. Tel.: +41 21 6932481; fax: +41 21 69 35060.
E-mail addresses: yuxuan.ji@epfl.ch (Y. Ji), nikolas.geroliminis@epfl.ch (N. Geroliminis).

0191-2615/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trb.2012.08.005

tion 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 Munoz and Daganzo, 2003; Helbing et al., 2009). Thus, the known fundamental diagram (initially observed for a stretch of highway and providing a steady-state relationship among 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 spatially aggregating the highly scattered plots of flow vs. density from individual loop detectors (e.g., 1 min data), 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) and similar approaches were introduced later by Herman and Prigogine (1979) and Daganzo (2007). 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, (1) that urban neighborhoods approximately exhibit a ‘‘macroscopic fundamental diagram’’ (MFD) relating the number of vehicles to space-mean speed (or flow), (2) there is a robust linear relation between the neighborhood's average flow and its total outflow (rate vehicles reach their destinations) and (3) 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 independent of time-dependent origin–destination tables. (1) is important for modeling purposes as details in individual links are not needed to describe the congestion level of cities and its dynamics. It can also be utilized to introduce simple control strategies to improve mobility in homogeneous city centers building on the concept of an MFD, like in Daganzo (2007), Geroliminis and Daganzo (2007), and Haddad and Geroliminis (2012). The main logic of the strategies is that they try to decrease the inflow in regions with points in the decreasing part of an MFD. (2) is important for monitoring purposes as flow can be easily observed with different types of sensors while outflow cannot. (3) is important for control purposes as efficient active traffic management schemes can be developed without a detailed knowledge of O–D tables.

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. Analysis of real data from a medium-sized French city (Buisson and Ladier, 2009) showed that heterogeneity has a strong impact on the shape/scatter of an MFD. With respect to property (3), recent findings from empirical and simulated data (Geroliminis and Sun, 2011; 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. Inconsistent distribution is expressed in terms of time. The aforementioned references observed that if different time periods have similar average density but very different variance of density, then the MFD is expected to experience high scatter or hysteresis phenomena. In case variance is constant but high, this is the case of an uneven distribution, so a low scatter MFD might not exist as well. They also observed low scatter relationships between network flow and variance of link density for a given network. Runs with vastly different demand profiles gave quantitatively same results (aggregated per one signal cycle). (Daganzo et al., 2011; Gayah and Daganzo, 2011) showed for simple networks with two interconnected rings, that networks with densities in the congested regime can produce strong instabilities, hysteresis and bifurcations and lead the system to gridlock. In other words, the average network flow is consistently higher when link density variance is low for the same network density, but higher densities can create points below an MFD when they are heterogeneously distributed.

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 heterogeneously congested 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. Nevertheless, if a cluster contains subregions with significantly different levels of congestion, the control strategies will be inefficient.

There is a vast literature 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). The stopping point depends on different criteria of partitioning. For instance, in some applications, a certain number of final clusters is predetermined. In some other applications, the process continues until there is only one cluster. Then various metrics can be defined to help determine the optimal number of clusters produced in the process. 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 cannot 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 dynamic features and potential control strategies to alleviate traffic congestion should 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 clustering algorithms to be developed need to satisfy: (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 zero. 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, the partitioning mechanism consisting of three consecutive steps of initial segmenting, merging and boundary adjustment is designed and described in detail. Secondly, variance and spatial metrics are introduced to obtain the desired number of clusters and evaluate the partitioning results. Finally, simulation is made in a real transportation network to analyze the quality of partitioning based on various metrics and comparing with other clustering algorithms. Discussion about the implementation and the applicability of partitioning and future directions are also presented.

2. 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) extract a small number of clusters with different congestion levels from the network at a global level, ignoring details and local features, such as a few congested links in a large uncongested area; (3) produce clusters that are spatially near compact 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. Firstly, we over segment the transportation network into several homogeneous regions. This step is achieved by using Normalized Cut algorithm (NCut), which can efficiently extract the major components from the network and guarantee spatially compact shapes. In this step, more than desired number of clusters may be produced by over segmenting which partitions a homogeneous region into several parts. Secondly, we recursively merge a pair of most similar clusters based on the mean values of their densities until a desired number of clusters is reached. This step fixes the problem of Ncut cutting large uniform regions. After these two steps, we will obtain a rough sketch of the network partitioned into clusters with different congestion levels and spatially compact shapes. Finally, we further minimize the variance of link densities within each cluster by repeatedly adjusting the boundaries among the clusters, while keeping the smoothness of their shapes. In addition, based on the three main objectives, several metrics are proposed to estimate the optimal number of clusters and evaluate the effectiveness of the proposed partitioning mechanism (including the homogeneity of densities and spatial compactness). The partitioning mechanism and metrics are described in detail in the following sections.

2.1. Initial segmenting

In initial partitioning, we over segment the transportation network into several clusters. This step is achieved by Normalized Cut algorithm. Ncut is a graph-based partitioning algorithm originally designed 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 many other algorithms mainly separate out the local features.

One of the main reasons that Ncut is applied in the initial segmenting is that it can efficiently extract the most obvious objects and produce spatially compact clusters. Suppose a scenario where there are only a few congested links in the center of a large transportation network. Since these few links are local heterogeneities within a macroscopically homogeneous region, Ncut will ignore them. This means that if two clusters are desired, Ncut will cut the whole uniform network into two balanced regions with similar size and the few congested links in the center may belong to one of them, instead of cutting out the few congested links, which are the details in a graph. Concerning transportation networks, since we aim at big blocks of obviously congested regions and do not focus on small clusters of links, Ncut is appropriate in initial partitioning. The reasons for applying Ncut in initial partitioning are as follows: (1) Ncut is a graph based segmentation algorithm and transportation networks can be designed as connected graphs; (2) it avoids the partitioning of cutting out a very small number of nodes and can produce clusters with balanced size of nodes; (3) it realizes perceptual grouping and extracts global impressions (major and obvious parts) from the graph or image, which is its most important feature; (4) it can produce spatially compact clusters; and (5) it is computationally efficient.

There has been a great contribution in the area of clustering. However, most of the other algorithms are designed for specific applications and may not be suitable for application in transportation networks. For example, *k*-means algorithm aims at minimizing the intra-cluster variance given a certain number of clusters. It can help produce a well defined MFD with similar link densities in the same cluster, but it is hard to guarantee the spatial compactness based on the feature vectors. In addition, this clustering can only guarantee local optima, which means that the final result is based on initial seeds (i.e., the initial centers of the clusters). Another well-known clustering algorithm is Minimum Cut, which seeks to minimize the similarity among the clusters. Minimum Cut can generate homogeneous clusters with similar link densities and spatial compactness but tends to cut small sets of isolated nodes, as explained by its cutting criterion. In transportation network, clusters with only a few links are not desirable as (1) MFD might exhibit high statistical errors and (2) simple perimeter control strategies cannot be easily designed for a network partitioned to a large number of clusters as route choice might change. Although there has been numerous clustering algorithms in the literature, it is not a trivial task to locate a proper one directly applicable in transportation networks and further modifications are needed. Some algorithms of similar principles as Ncut may also be appropriate such as Min–Max Cut, which has a different objective function than Ncut but also seeks to minimize the intra-cluster variance and maximize the inter-cluster variance (Ding et al., 2001).

Although Ncut algorithm cuts the major components with compact shapes out of the graph, its partitioning result may not, to the most extent, satisfy the first objective of minimizing the variance of link densities within each cluster. As we have discussed before, the first criterion conflicts with the other two. A restatement of our criteria is that we aim at achieving the smallest possible variance within all the spatially 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 noticed by Cour et al. (2005), Ncut tends to cut a large uniform region into two if the spatial distance threshold (measured by the length of shortest path in graph) is too low. This threshold helps determine the similarity between two links. When the length of shortest path between two links is larger than the threshold, their similarity is 0. Otherwise, their similarity is then determined by their densities. However, higher spatial threshold value is very likely to produce spatially uncompact clusters. Thus, Ncut algorithm cannot fully comply with our criteria when it is applied to the transportation network, and therefore further modifications and refinements are needed.

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. Specifically, the transportation network is built 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). Two-way roads are represented as two parallel undirected links. 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. Distance $r(i,j)$ is calculated based on the adjacent matrix of the graph G. The adjacent matrix $\{a(i,j)\}$ (with only 0, 1 values) measures the neighboring relation between each pair of links, with $a(i,j) = 1$ denoting that links i and j are adjacent and vice versa. Thus the shortest path is the minimum number of edges node i has to pass to reach node j in the graph G. In order to guarantee spatially connected clusters, we set the spatial distance threshold value to be 1 and define the similarity function $w(i,j)$ between link i and j as follows[1]:

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

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

Suppose the node set V in a graph $G = (V,E)$ where E denotes the set of edges in G 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. Ncut uses 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:

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

$$Nassoc(A,B) = \frac{cut(A,A)}{cut(A,V)} + \frac{cut(B,B)}{cut(B,V)} \tag{3}$$

The two objectives of minimizing $Ncut(A,B)$ and maximizing $Nassoc(A,B)$ can be reached simultaneously since they obey the following relation:

$$Ncut(A,B) = 2 - Nassoc(A,B) \tag{4}$$

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 (Shi and Malik, 2000). Thus the initial segmenting process by Ncut is described below:

1. Given a graph built from the transportation network, set the weight $w(i,j)$ on the edge connecting two nodes to be a measure of the similarity between two links.
2. Solve the equivalent eigenvalue system and get the smallest eigenvalues.

[1] This similarity function is a Gaussian probability distribution function. It is monotonically decreasing, but the rate of decrease is higher for larger difference of $|d_i - d_j|$, which gives higher penalty for dissimilarity among links with different levels of congestion.

3. Pick up the eigenvector with the second smallest eigenvalue, discreticize it and bipartition the graph.
4. Bipartition each subgraph and a new partitioning is obtained with the number of clusters increased by one. Continue this process until a partitioning which has several more clusters than desired is reached.

2.2. Merging

After completing the first step, we have several initial partitioning with different numbers of clusters. We can also evaluate the clustering results based on some metric to get the optimal number of clusters generated by Ncut. We will introduce this later. 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 ($r(i,j)$ in Eq. (1)) is too low. Therefore, in the second step, we develop 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 link densities, until we reach only one cluster. Then we can again use the same metric we will design later to estimate the optimal number of clusters after merging.

The merging algorithm is similar as the agglomerative clustering algorithm (Johnson, 1967). 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 $O(k^2 \log k)$ where k is the initial number of 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 $O\left(n^{\frac{3}{2}}\right)$ from Ncut in (Shi and Malik, 2000). As for the agglomerative algorithm, it costs $O(n^2 \log n)$. Secondly, this Ncut-based merging process can produce near compact clusters when only link densities are taken into account, while it will be difficult for the agglomerative clustering algorithm to achieve this even if both spatial and density information are used.

2.3. 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 the second and third criteria of partitioning have been satisfied. Besides, it is obvious that both Ncut and merging also aim at decreasing the variance of link densities within each cluster during the partitioning process. 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 clear.

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

We introduce a straightforward boundary adjustment algorithm and then extend and implement it in the network. Suppose i denotes a link and B denotes the set of links in cluster B. If link $i \in B$ and link i is adjacent to link j (i.e., $a(i,j) = 1$) where $j \in A$, we say links i and j are on the boundary between cluster A and B. Let $U(A,B)$ denotes the set of all the links on the boundary between cluster A and B. Firstly, we identify all the boundary links and assume $i \in B$ and $i \in U(A,B)$. 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 each cluster. 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 the variances of both clusters. After a few algebraic calculations, we can show that the criterion of decreasing both density variances of clusters A and B are met when:

$$\begin{cases} \frac{(d_i - u_A)^2}{Var(A)} < \frac{N_A + 1}{N_A} \\ \frac{(d_i - u_{B\setminus i})^2}{Var(B\setminus i)} > \frac{N_B}{N_B - 1} \end{cases}, \tag{5}$$

where $Var(A)$ and u_A are the variance and mean of the link densities in cluster A. $B\setminus i$ denotes the set of all the other links from B except i. N_A and N_B are the number of links in cluster A and B respectively. 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 density distance from link i to the center of cluster A is smaller than the average density distance of links within A to its center, adding link i to cluster A will decrease the density variance of A. The center of cluster A is the mean of link densities in cluster A as denoted by u_A and the average density distance of links within A to its center is the standard deviation of link densities in cluster A, $\sqrt{Var(A)}$. 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 weird 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

$Y \subset B$, $Y \subset U(A,B)$, $A' = A \cup Y$. The variances of link densities in both A and B will be decreased if the following two conditions hold.

$$\begin{cases} \frac{(u_A - u_Y)^2}{\mathrm{var}(A) - \mathrm{var}(Y)} < \frac{N_A + N_Y}{N_A} \\ \frac{(u_{B\backslash Y} - u_Y)^2}{\mathrm{var}(B\backslash Y) - \mathrm{var}(Y)} > \frac{N_{B\backslash Y} + N_Y}{N_{B\backslash Y}} \end{cases}. \tag{6}$$

The proof is straightforward.

$$Var(A) - Var(A') = \left[\frac{\sum_{i\in A} d_i^2}{N_A} - \left(\frac{\sum_{i\in A} d_i}{N_A}\right)^2\right] - \left[\frac{\sum_{i\in A} d_i^2 + \sum_{i\in Y} d_i^2}{N_A + N_Y} - \left(\frac{\sum_{i\in A} d_i + \sum_{i\in Y} d_i}{N_A + N_Y}\right)^2\right]. \tag{7}$$

After some manipulations we obtain:

$$Var(A) - Var(A') = \frac{N_A N_Y [Var(A) - Var(Y)] + N_Y^2 [Var(A) - Var(Y)] - N_A N_Y (u_A - u_Y)^2}{(N_A + N_Y)^2}. \tag{8}$$

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 density variance decreasing for cluster B is obtained in a similar way.

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

1. For each cluster, find all the links on the boundary and build a spatial sequence for the links on the boundary based on their spatial neighboring relations.
2. On each boundary, 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 and thus the computational cost for the boundary adjustment algorithm is $O(s^2n)$. 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).

3. Metrics development

The partitioning process produces a series of different partitioning with different number of clusters. Hence we introduce three metrics to evaluate different partitioning and estimate the optimal number of clusters. Firstly, we design a metric to evaluate how well regions of different congestion levels are separated by computing both the intra-cluster (within each cluster) and inter-cluster (between different clusters) similarities of link densities. Secondly, we compute the total variance of link densities in a partitioning to evaluate how close we reach the objective of minimum variance. Finally, we also design a shape metric to evaluate the spatial compactness of the clusters.

3.1. The variance metrics

In order to evaluate and compare partitioning results with different number of clusters, we introduce several metrics in this section based on the variances and means of the clusters. These metrics will also contribute to identifying the optimal number of clusters during the partitioning process. The first metric we designed to evaluate the partitioning is 'NcutSilhouette' (NS) as follows:

$$NS_k(A,B) = \frac{\sum_{i\in A}\sum_{j\in B}(d_i - d_j)^2}{N_A N_B}, \tag{9}$$

where k is the number of clusters. NS_k does not contain any spatial information and only measures the average quadratic 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:

$$NS_k(A) = \frac{NS_k(A,A)}{NSN_k(A,B)}, \\ \text{where } NSN_k(A,B) = \min\{NS_k(A,K) | K \in Neighbor(A)\}, \tag{10}$$

and *Neighbor*(*A*) denotes the set of clusters that are spatially adjacent to cluster *A*. In this metric, $NS_k(A,A)$ measures the intra-cluster similarity of densities while $NSN_k(A,B)$ measures the inter-cluster similarity. If two clusters are not spatially adjacent, it can also be a good partitioning even if their link densities are close. Therefore, we only measure the inter-cluster similarity of cluster *A* with its neighbors (i.e., *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. (10). 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:

$$NS_k = \frac{\sum_{A\in C} NS_k(A)}{k}, \tag{11}$$

where *C* is the set of clusters and *k* is the total number of clusters. The idea of the *NS* metric is similar as the one defined in Rousseeuw (1987). However the difference lies in the evaluation of the dissimilarity between two clusters, where we only measure the dissimilarity of a cluster to its neighbors instead of all clusters, due to spatial separations.

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

$$NS_k(A,B) = Var(A) + Var(B) + (u_A - u_B)^2. \tag{12}$$

The proof is straightforward.

$$\begin{aligned} &NS_k(A,B)\\ &= \frac{\sum_{i\in A}\sum_{j\in B}(d_i - d_j)^2}{N_A N_B}\\ &= \frac{\sum_{j\in B}\sum_{i\in A} d_i^2 + \sum_{i\in A}\sum_{j\in B} d_j^2 - 2\sum_{i\in A}\sum_{j\in B} d_i d_j}{N_A N_B}\\ &= \frac{N_B\sum_{i\in A} d_i^2 + N_A\sum_{j\in B} d_j^2 - 2N_A N_B u_A u_B}{N_A N_B}\\ &= \frac{N_A N_B\left(\frac{\sum_{i\in A} d_i^2}{N_A} - u_A^2\right) + N_A N_B\left(\frac{\sum_{j\in B} d_j^2}{N_B} - u_B^2\right) - 2N_A N_B u_A u_B - N_A N_B\left(u_A^2 + u_B^2\right)}{N_A N_B}\\ &= \frac{N_A N_B Var(A) + N_A N_B Var(B) - 2N_A N_B u_A u_B - N_A N_B\left(u_A^2 + u_B^2\right)}{N_A N_B}\\ &= Var(A) + Var(B) + (u_A - u_B)^2. \end{aligned} \tag{13}$$

Hence we get:

$$NS_k(A) = \frac{NS_k(A,A)}{NS_k(A,B)} = \frac{2Var(A)}{Var(A) + Var(B) + (u_A - u_B)^2}. \tag{14}$$

Based on Eq. (14), we observe that when the difference of means is large and the variances are relatively small, *NS* value will be small which implies 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 around 1. Generally, for a cluster *A* with smaller variance than *B*, it is properly partitioned since $Var(A) < Var(B) \Rightarrow NS(A) < 1$. For a cluster *A* with larger variance than *B*, 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 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.

Besides we can use the total variance of the clusters to evaluate the quality of a partitioning as follows:

$$TV = \sum_{A\in C} N_A * Var(A). \tag{15}$$

Both *NS* and *TV* metric are used to evaluate the homogeneity of link densities based on our objectives. However, the *NS* metric can also be used to estimate the optimal number of clusters while *TV* cannot since the latter one prefers larger number of clusters and does not consider the inter-class similarity at all (*TV* is minimized to zero when each link is a single cluster). Also note that the *NS* metric alone is not enough to evaluate or even optimize the partitioning since there is also spatial requirement. For example, suppose there are three links *i*, *j* and *l* with neighboring relations $a(i,j) = a(j,l) = 1$ and $a(i,l) = 0$, and densities $d_i = d_l = 1$ and $d_j = 0$. The optimal *NS* value will be obtained if the partitioning contains two clusters *A* and *B*, where $A = \{i,l\}$ and $B = \{j\}$. However, this is not a feasible solution since link *i* and *l* are not connected.

3.2. The shape metric

Based on the objective of spatial compactness, we also design a metric to measure the smoothness of the boundaries along the clustered regions. We firstly build a clockwise (or counter-clockwise) sequence for the boundary nodes along each region to draw the shape of the region. Suppose the sequence is "…, $(i-1)$, i, $(i+1)$, …". Secondly, we design a boundary angle measure for each boundary node in a certain region for the degree of smoothness around this node. Specifically, the boundary angle for node i is defined as:

$$BoundaryAngle(i) = \angle(i-1)i(i+1), \tag{16}$$

where $\angle(i-1)i(i+1)$ is the angle less than π at node i formed by the two segments $(i-1)i$ and $i(i+1)$. We evaluate the smoothness of a boundary node by setting a threshold value for smoothness such as $\pi/2$. For instance, if a boundary angle $\alpha > \pi/2$, we say it is smooth. Otherwise we tag it as a non-smooth node. Finally, for each of the non-smooth boundary node i, we calculate the area of a triangle formed by nodes i,$i-1$ and $i+1$. Then the non-smoothness of the a region R can be roughly estimated with the dimensionless quantity:

$$NonSmoothness(R) = \sum_i A(i)/A(R), \tag{17}$$

where $A(i)$ is the area of a non-smooth boundary node as before and $A(R)$ is the area of the whole region. Therefore, based on this definition, the smoothness of the boundary along a region is appropriately measured as a relative value, which implies that for a large region the existence of only a few non-smooth boundary nodes will not seriously affect the whole smoothness measure while for a small region it can be the opposite.

With the above metric we can now evaluate the spatial smoothness of all the regions in a partitioning. However, building a spatial clockwise (or counter-clockwise) sequence for all the boundary nodes turns out to be a nontrivial task. In this direction, a fast algorithm based on spanning tree to obtain the correct boundary node sequence is developed and applied in the network in our recent work in Ji and Geroliminis (2011).

4. Implementation

In this section, we apply and evaluate the effectiveness of our partitioning mechanism. We show how the optimal *NS* metric and *TV* improve in each step of the partitioning while the shape metric is properly maintained for a real transportation network simulated in a micro-simulation environment. 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 the mechanism 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 ft. 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 s for the west boundary of the area (The Embarcadero) and 60 s for the rest. A 4hr time-dependent traffic demand (120 time intervals of 2 min) is applied to this network, which produces different spatial and temporal levels of congestion. The computational complexity has been studied in previous sections. The practical computational time for the partitioning mechanism including all three steps for this network with around 400 links is within a few seconds, which means that this approach can be applied for control purposes even in real-time.

4.2. Partitioning results

4.2.1. Analysis on variance metrics

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. This means that there is a large range of density values, which will be a strict test for the developed algorithms. The original network with link density values at time $t = 70$ is shown in gray-scale level in Fig. 1.1, where light color means low-density link while dark is a jammed link. Fig. 1.2–.8 are the initial partitioning results by Ncut with number of clusters from 2 to 8. Fig. 1.3 shows the optimal partitioning as determined by *NS* (i.e., optimal number of cluster is 3 by Ncut). In the second step, Fig. 1.9–.14 show the merging process from 8 to 2 clusters and the optimal one is again a 3-cluster partitioning shown in Fig. 1.13. After the first two steps, we get an optimal partitioning with three spatially compact clusters of the network in Fig. 1.13. In order to further improve *NS* metric and *TV* of link densities, boundary adjustment is implemented in the last step for the cluster of Fig. 1.13 and the final partitioning is shown in Fig. 1.15 (note that the lower bound on the length of a group of simultaneously adjusted consecutive links, which is 25% of the full boundary

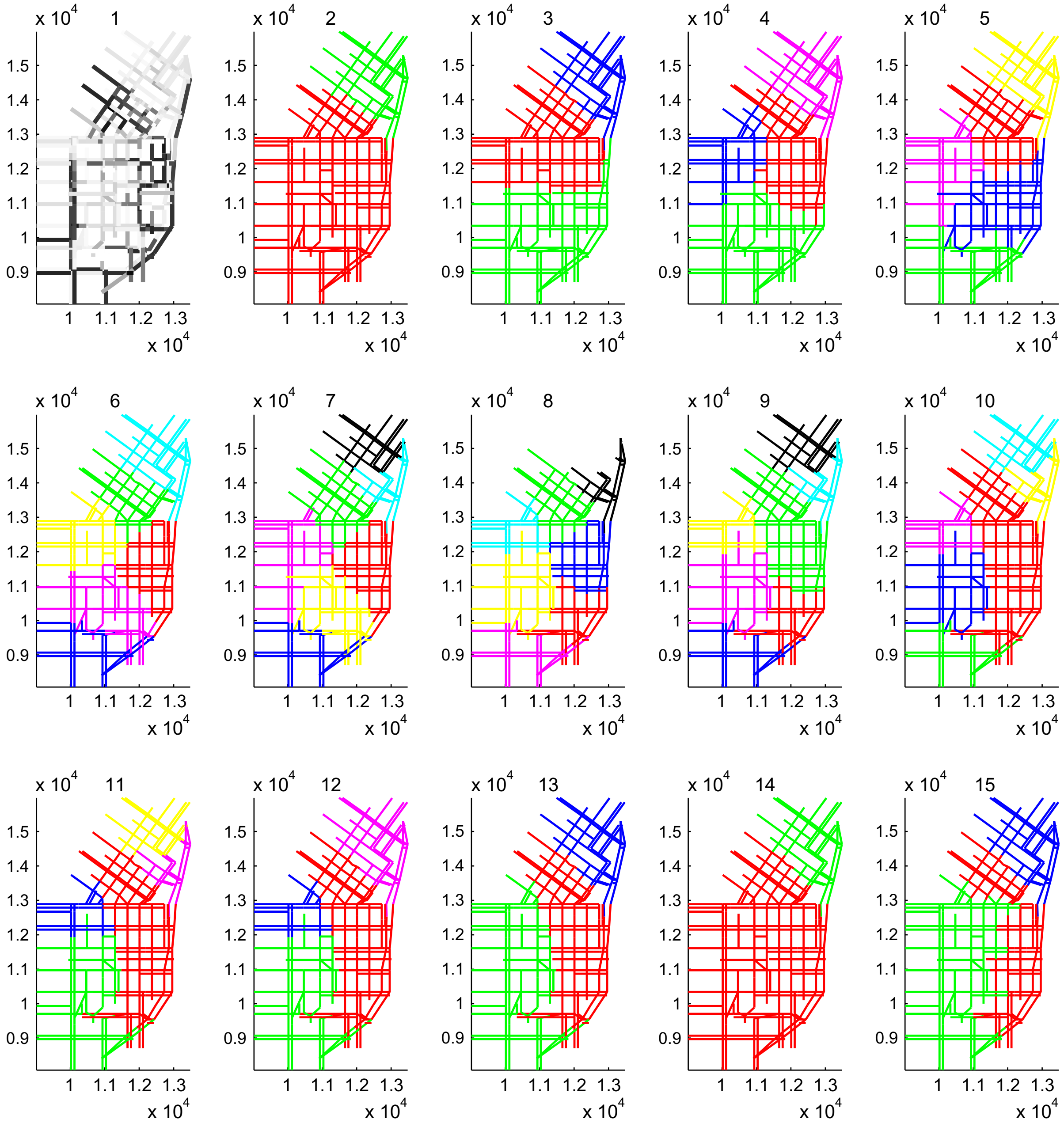

Fig. 1. Partitioning at $t = 70$ by Ncut (1.2–1.8), merging (1.9–1.14) and boundary adjustment (1.15).

length, is implemented in the boundary adjustment algorithm to keep spatial compactness). Figures appear in color in the online version of the paper.

Accordingly, Table 1 explains the metric values. Table 1.1 shows *NS* values defined in Eq. (11) by initial 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 Ncut. The optimal number of clusters 3 is still obtained, but the *NS* value is smaller than the optimal one by Ncut (0.6865 vs. 0.7442).

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 $(\text{veh/m})^2$ and veh/m). Table 1.3 compares the *NS* value and *TV* by Eq. (15) 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, *TV* decreases by around 14.5% 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 13.4% from Ncut to the final result; green decreased by 32.3%; and the blue decreased by 9.1%. As for the *NS*

Table 1
NS metric, variance and mean (both in units) of link densities at $t = 70$.

Table 1.1 Average NS by Ncut (optimal: 3)

# 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 (optimal: 3)

# 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 TV of link densities and average NS

	Ncut	Merging	Bo. Adj.
TV	0.1249	0.1212	**0.1153**
Average *NS*	0.7442	0.6865	**0.6210**

Table 1.4 Variance ($\times 10^{-3}$ units) and NS within each cluster

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.	0.4722/1.0707	0.2809/0.6370	0.0696/0.1553

Table 1.5 Mean of link densities within each cluster

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.	0.0286/149	0.0150/141	0.0075/76

Table 1.6 Average mean difference of each partitioning

	Ncut	Merging	Bo. Adj.
Ave. mean difference	0.00795	0.01155	0.01735

metric, it decreases for both green and blue clusters and keeps around the same for the red. 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 density difference of the neighboring clusters in each partitioning following the concept of *NS* developed in Section 3.1:

$$\frac{\sum_{(A,B)\subset C}|Mean(A) - Mean(B)|}{\|C\|}, \tag{18}$$

where C is the set of pairs of neighboring clusters and $Mean(A)$ is the mean of link densities in cluster A. This metric measures how distinct two clusters are. Note that the mean difference significantly increases 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 Fig. 2 (x-density, y-frequency). Fig. 2.1–.3 show the histogram of frequency of link densities in each cluster by initial Ncut (e.g., Fig. 2.1 describes the frequency of link densities in the red cluster). Similarly, Fig. 2.4–.6 show the histograms after merging and Fig. 2.7–.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 Fig. 2.1 with boundary adjustment in Fig. 2.7). More specifically, Fig. 2.1 shows that after Ncut 59.62% of the links in the red cluster have density less than 0.025 and 26.92% of the links have density more than 0.04. Fig. 2.7 shows that after boundary adjustment the percentages of links in these two density ranges in the red cluster become 45.64% and 38.93% respectively. The observations are similar for the other two clusters. However since the spatial information is not included, it is unlikely to obtain completely separate distributions of link densities in the histograms.

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 then take a look at some other periods of a day when different patterns may occur at $t = 75$ (more congested) and $t = 40$ (less congested).

The network density at $t = 75$ is shown in Fig. 3.1. We compare two different final partitioning, one with two clusters and the other with three clusters. The partitioning process to reach 2 clusters is shown in Fig. 3.2 (after merging) and Fig. 3.3 (after boundary adjustment). For final 3 clusters, it is shown in Fig. 3.4 (after merging) and Fig. 3.5 (after boundary adjustment). Table 2 compare the metrics, and it is shown that the *NS* value of three clusters is worse than the one of two clusters after merging, but significantly better after boundary adjustment. *TV* decreases by 17.2% after boundary adjustment for 3 clusters while 9.5% for two clusters. 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. In the end, the network density at

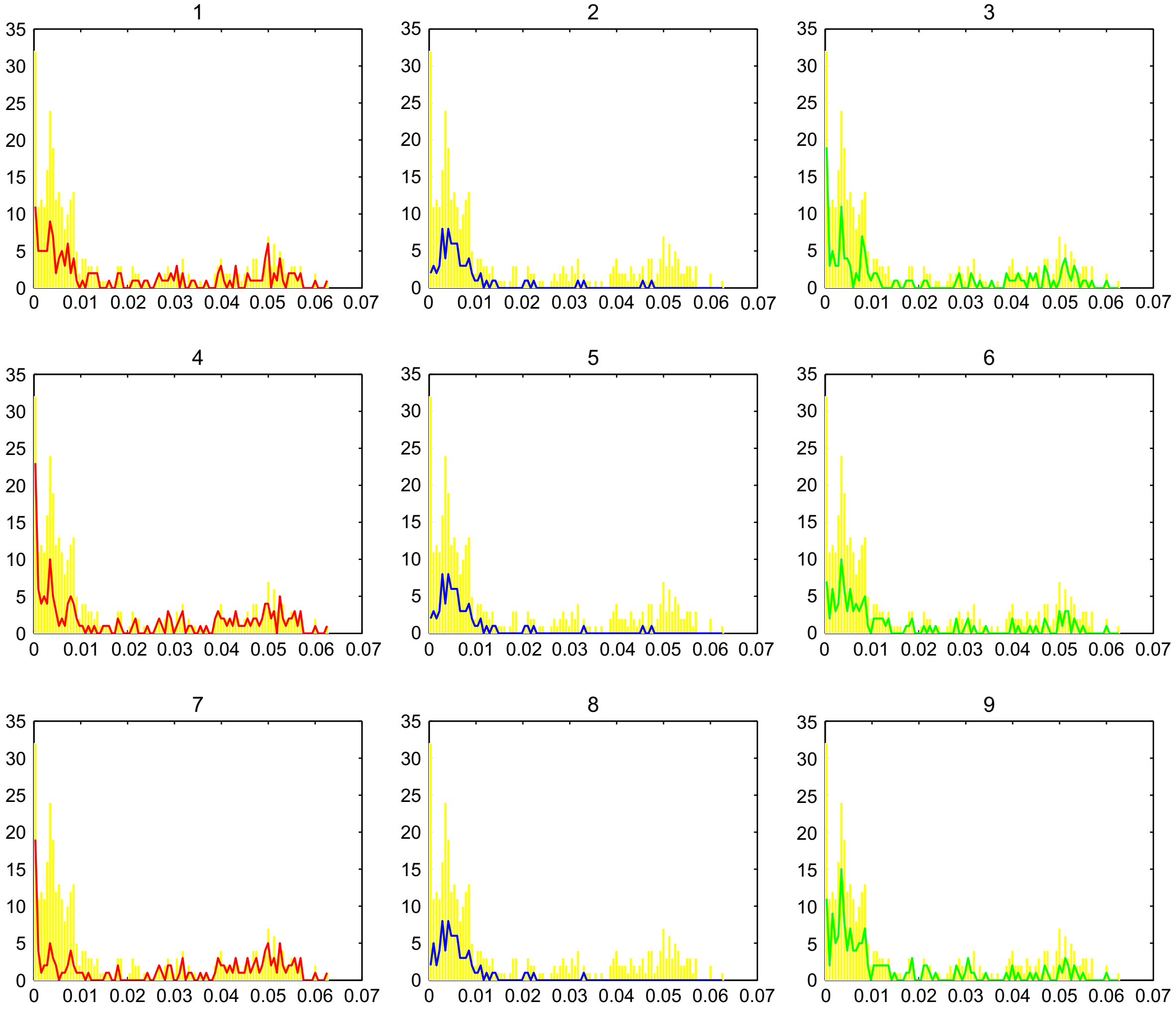

Fig. 2. Histograms of link densities at $t = 70$ (2.1–2.3 by Ncut, 2.4–2.6 after merging, 2.7–2.9 after boundary adjustment).

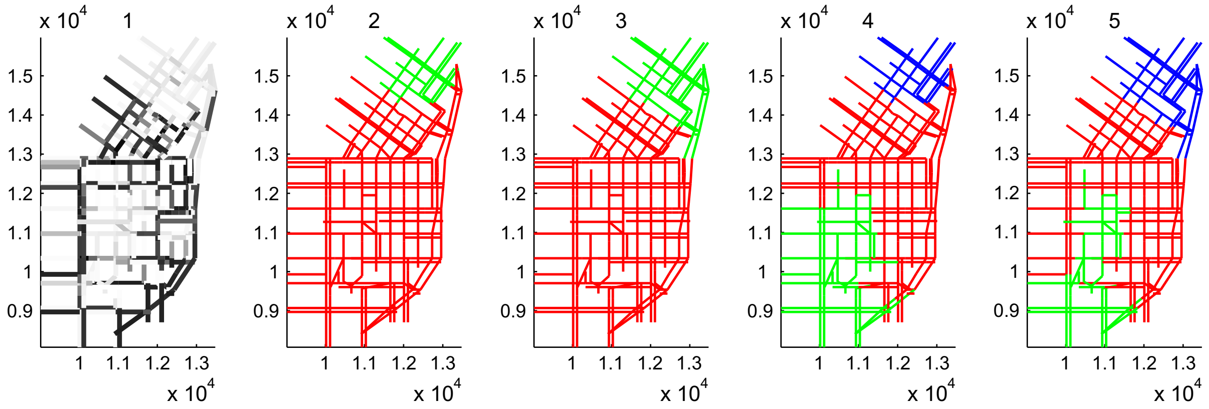

Fig. 3. Network density (3.1) and final partitioning with two clusters (3.2 merging, 3.3 boundary adjustment) and three clusters (3.4 merging, 3.5 boundary adjustment) at $t = 75$.

$t = 40$ shows a uniformly uncongested network with similar link densities (the density figure at this time full of light color links is omitted). We observe that after the network is partitioned into several components with spatially compact shapes, *TV*

Table 2
TV and average *NS* (in units) with two clusters and three clusters at $t = 75$ (TV of original network: 0.1795).

	2 clusters			3 clusters		
	Ncut	Merging	Bo. Adj.	Ncut	Merging	Bo. Adj.
TV	0.1721	0.1658	0.1625	0.1647	0.1617	0.1486
Av. *NS*	0.8737	0.5798	0.6531	0.7826	0.6242	0.5502

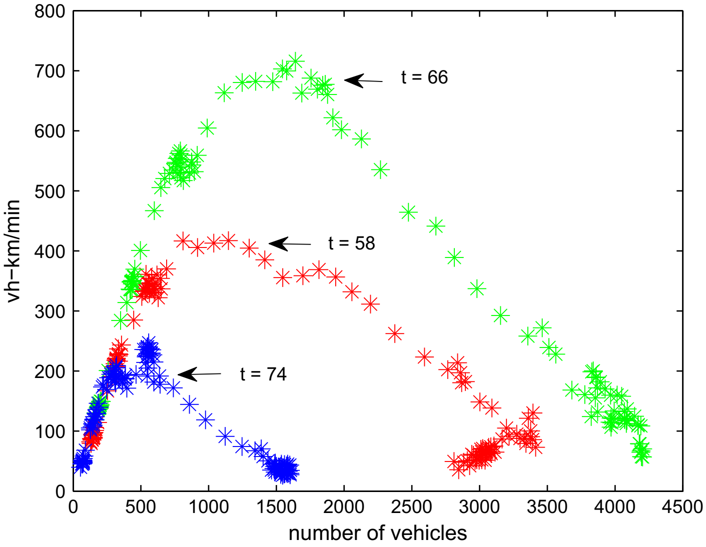

Fig. 4. MFDs for the three partitioned regions.

is not decreased significantly and remains almost the same as the original network without partitioning. In addition, the *NS* metric is always around 1 for an arbitrary number of clusters (from 1 to 8). Both metrics imply the homogeneity of link densities in the network at time $t = 40$. Therefore we conclude that the network at this time period does not need partitioning.

We also investigate the shape of the MFDs for the three partitioned regions. The results are summarized in Fig. 4. This figure plots the number of vehicles vs. the veh-km traveled per minute in each region for the whole time period, given a constant partitioning as of time $t = 70$. Note that all three regions experience MFD with low scatter and points below the graphs are not observed (this is the case for individual links fundamental diagrams). The blue[2] region experiences some more scatter around the critical density once compared with the other regions. This is because it contains the smallest number of links. Nevertheless, there is a clear distinction between congested and uncongested regime for all regions. Note that the time each of the regions reaches the congested regime is very different. The central region (Red) reaches congestion at time $t = 58$ and then it propagates in the Green ($t = 66$) and Blue ($t = 75$) regions. This propagation of congestion would not be observable by looking at the unified MFD, which reaches the congestion at time $t = 61$ (figure not shown here). Thus, perimeter control strategies following up a city partitioning, should try to avoid or postpone congestion to reach regions with high density of destinations (see for example (Daganzo, 2007) or (Haddad and Geroliminis, 2012)). Dynamic partitioning applications, which might further improve the homogeneity of the regions are discussed in the future work.

4.2.2. Analysis on shape metric

The above analysis on the variance metrics shows significant improvement of the partitioning quality (i.e., more intra-cluster similarity and inter-cluster dissimilarity) from the initial segmentation to the final results based on the proposed methods. However our objectives of partitioning also include spatial requirement of compactness to facilitate future design of control strategies. In the first two steps of partitioning the spatial compactness is guaranteed by setting a distance threshold which strictly keeps the spatial connectivity and in the final step the adjustment is made based on the *TV* metric under some constraint of spatial compactness (e.g., the lower bound on the number of simultaneously adjusted links). Thus we now evaluate the spatial compactness metric and show the effectiveness of our final boundary adjustment algorithm which can further decrease the variance while at the same time maintain the spatial smoothness properly.

[2] For interpretation of color in Fig. 4, the reader is referred to the web version of this article.

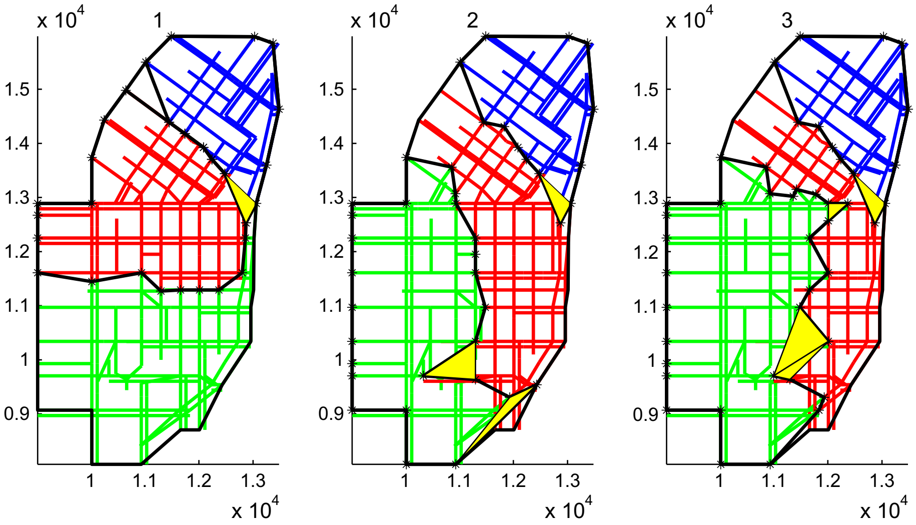

Fig. 5. Spatial compactness measure of partitioning at $t = 70$.

Table 3
Spatial compactness evaluation at $t = 70$.

Partitioning	Ncut	Merging	Bo. Adj.
Shape metric (Non-smoothness)	0.62%	2.73%	3.31%

We evaluate the shape metric for three partitioning results given by initial Ncut in Fig. 5.1, after merging in Fig. 5.2 and boundary adjustment in Fig. 5.3 (all with 3 clusters). The yellow areas are the non-smooth regions in each partitioning detected by the spatial compactness metric. The spatial non-smoothness metric is shown in Table 3. Note that the smoothness along the external network boundary is not included. It is clear that the spatial compactness is properly maintained through the presented mechanism.

4.2.3. Comparison with other algorithms

In previous section, we show the improvement of the partitioning mechanism from initial 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 still analyze the 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 does not change (i.e., the clusters are stable). In k-means algorithm, feature vector is used to

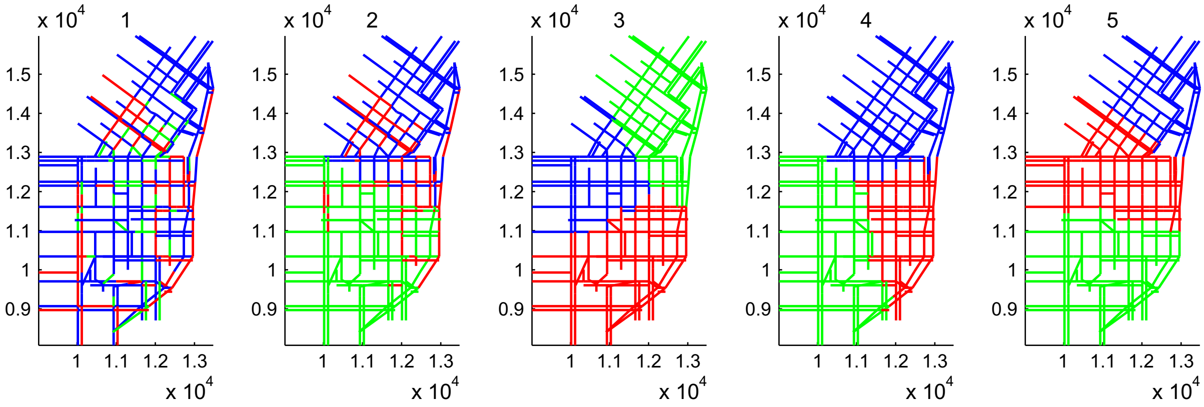

Fig. 6. k-Means clustering at $t = 70$.

Table 4
TV and average *NS* (in units) by *k*-means at $t = 70$, compared with our mechanism.

	Fig. 6.1	Fig. 6.2	Fig. 6.3	Fig. 6.4	Fig. 6.5	Ours
TV	0.0073	0.0530	0.1339	0.1287	0.1284	0.1153
Ave. *NS*	0.0977	0.6208	1.0295	1.0218	0.8278	0.6210

measure the similarity and make clusters. Therefore we include both spatial (as $x - y$ coordinates) and link density 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 $(w_s * x, w_s * y, w_d * d)^T$. Fig. 6.1–.5 shows different partitioning results by *k*-means with $k = 3$ clusters and Table 4 gives the corresponding metric values for each partitioning. For instance, Fig. 6.1 is a partitioning with $w_s/w_d = 1$. In this case, the *NS* metric and *TV* are very low. However, this partitioning is meaningless since there is no spatial compactness or connectivity, which also explains the high conflicts between spatial and density criteria. Fig. 6.2 is the case when $w_s/w_d = 4$. Spatial compactness exists to some extent but some links are still highly disconnected. Fig. 6.3–.5 shows three different partitioning when $w_s/w_d = 9$. When the spatial feature receives higher weight, connectivity 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 and the *TV*, as seen from Table 4.

k-Means is not very appropriate 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 local optimality. Secondly, *k*-means algorithm is based on cluster centers (means), which cannot be easily 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, while in *k*-means, we have to build a feature vector for each of the link. Spatial coordinates are often used as two features but cannot guarantee the connectivity of links in the final clusters. However in fact, the links in transportation networks should be more reasonably grouped based on their neighborhood and connectivity, instead of their physical distance or lengths of links.

As discussed in Section 2, an appropriate clustering algorithm in the initial partitioning is crucial. It is impossible to exhaust all the clustering algorithms in literature but here we can show some different results by replacing Ncut by *k*-means as initial segmenting. This means that the 3-step partitioning now includes initial segmenting of *k*-means, merging and boundary adjustment (the latter two steps are the same as before). We have seen above that when $w_s/w_d = 1$ or $w_s/w_d = 4$, the partitioning by *k*-means does not make any sense due to the spatial chaos of clusters. So in the initial partitioning of *k*-means at time $t = 70$, we set $w_s/w_d = 9$. Since *k*-means is not stable, three different partitioning results are given in Fig. 7.1–.3 (Fig. 7.4 is partitioning with Ncut as initial to be compared). We see clearly that even with high distance/density ratio, the spatial compactness can still be hardly guaranteed by *k*-means and the partitioning results are highly unreliable. Although the *NS* and *TV* metrics in the partitioning with Ncut as initial are to a small extent sacrificed compared to the one with *k*-means as initial, the partitioning with Ncut as initial is more desirable since the spatial smoothness is well maintained.

4.2.4. Traffic propagation

Until now we have discussed the partitioning of a transportation network at a certain time period based on both density similarity and geographical connectivity of links. However, traffic conditions are changing during a day and the congested area may grow or shrink with time. In order to capture congestion spreading phenomena, we extend our mechanism to

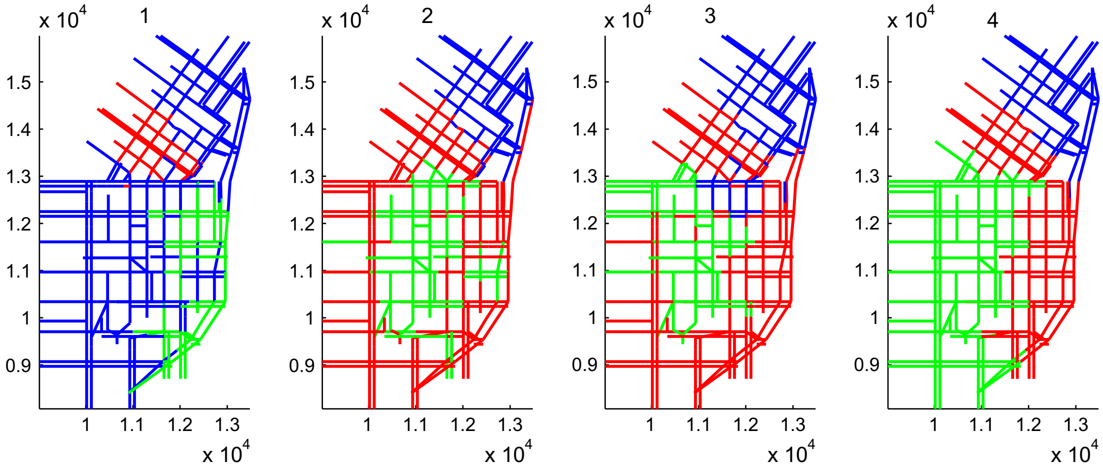

Fig. 7. 3-step partitioning with *k*-means as initial segmenting ($w_s/w_d = 9$, 7.1–7.3) instead of Ncut (7.4).

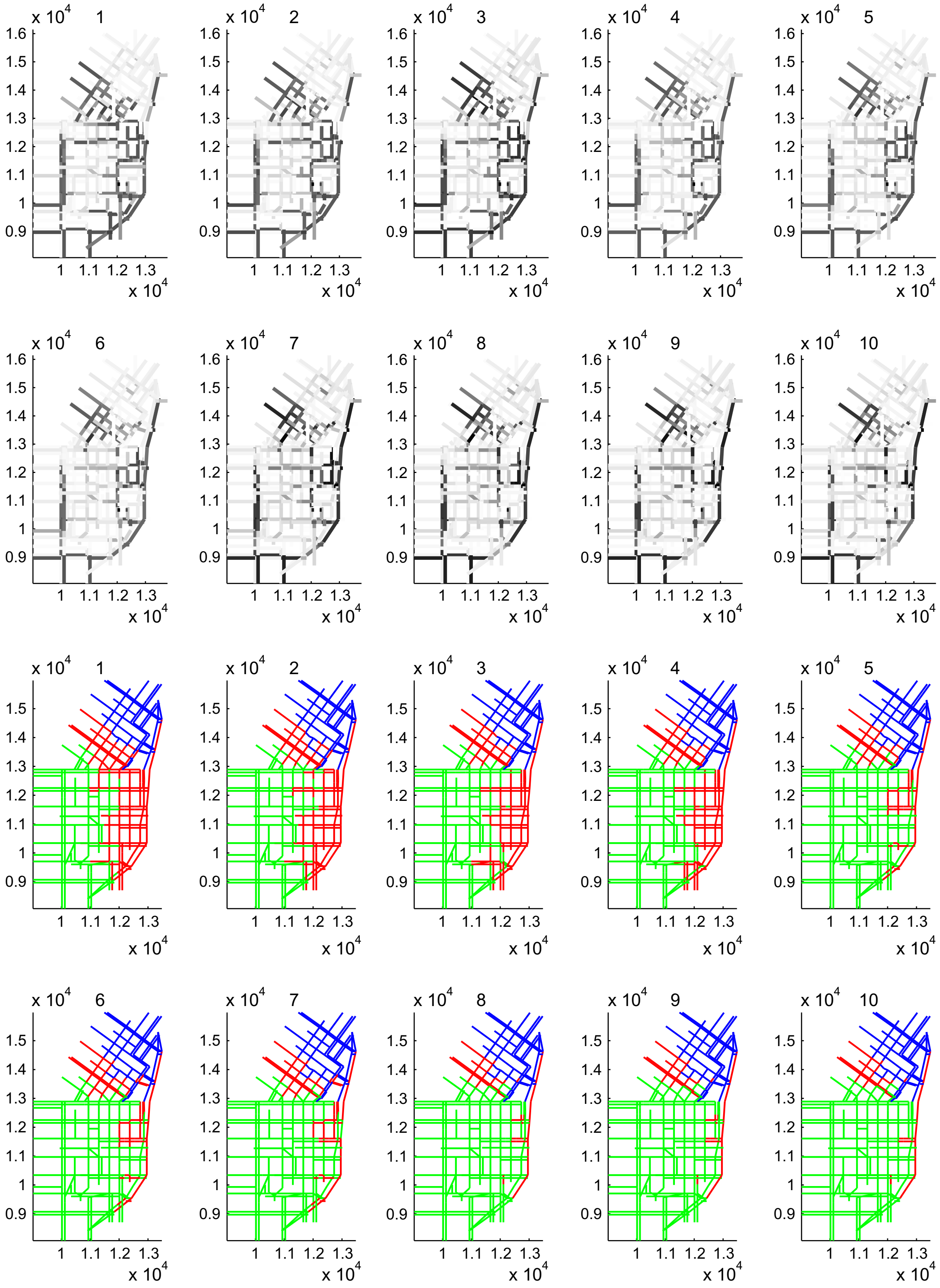

Fig. 8. Network density (grayscale) and traffic propagation (shrink) from $t = 72$ to $t = 63$.

the dynamic cases by repeatedly applying the boundary adjustment algorithm to the next time period based on the partitioning of the current time period. This simple extension is based on the fact that the traffic conditions of two close time periods might be very similar. Our aim is to provide some initial evidence that a boundary adjustment can capture congestion spreading in cases this is smooth. Nevertheless, this is a difficult problem that deserves further attention.

We first partition the network at time t by applying the 3-step partitioning mechanism. Then we apply only the boundary adjustment algorithm to the obtained partitioning at t with renewed link densities at $t + 1$ (or $t - 1$). More generally, the partitioning result at $t + i$ (or $t - i$) is obtained by applying boundary adjustment to the partitioning at $t + i - 1$ (or $t - i + 1$) with renewed link densities at $t + i$ (or $t - i$). In simulations, we set the initial partitioning at $t = 72$ when the network is highly congested but still can be partitioned, and repeatedly run the boundary adjustment algorithm until $t = 63$ when the network tends to be homogeneously uncongested. The reason for this backward partitioning in time is explained in detail by the end of this section. In order to capture the traffic propagation more clearly, we loose the spatial compactness constraint in the boundary adjustment process (i.e., the lower bound of simultaneously adjusted links). As a result, the clusters may be disconnected to some extent. The network densities and partitioning from $t = 72$ to $t = 63$ are shown in Fig. 8.

From Fig. 8, we see that the extended dynamic partitioning mechanism can properly capture the shrinking of the congested area. However we also notice that as time elapses, occasional links are disconnected from their clusters. Although both *TV* and *NS* metric we get for time $t = 65$ (and $t = 64{,}63$) are very small, a partitioning with only one cluster (generally homogeneous) or two may be more appropriate since there are few links in the congested (red[3]) region.

The reason we choose time $t = 72$ as the initial partitioning is that the network is partitionable at this time (optimal number of clusters is larger than 1). With a well partitioned network at a certain time, we can capture the growing and shrinking phenomenons of congested regions in the network by simply applying boundary adjustment algorithm. We repeat the same procedure by setting the initial partitioning at $t = 70$ and let time increase until $t = 80$. Our approach captures well the growing of the congested area. On the contrary, if we use $t = 63$ or $t = 80$ as initial partitioning when there is no obvious clusters in the network (i.e., the network is homogeneously congested or uncongested), the proposed dynamic partitioning is not producing desired clusters with compact shapes when it reaches partitionable time periods. This is mainly because the boundary adjustment algorithm only works on a network that has already been roughly well partitioned after some preprocessing, such as initial segmenting and merging in the proposed method. Repetitive boundary adjustment can help to further decrease the link density variance but cannot perfectly guarantee spatial compactness, so it is rarely possible to produce clusters with compact shapes. This observation provides additional evidence why boundary adjustment algorithm alone cannot fulfill all the requirements in partitioning a transportation network and the initial segmenting and merging algorithms are important and necessary.

5. Discussion

Traffic congestion is increasing in urban cities. 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. The proposed mechanism can produce a partitioning with a desired number of clusters that has both small link density variances and spatially compact shapes, which are validated by both density variance metrics and spatial compactness metrics. Furthermore, by comparing with some other clustering algorithms such as original Ncut algorithm and *k*-means, this mechanism demonstrates superiority of both effectiveness and robustness in partitioning a real urban transportation network. The 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 are currently working on designing control strategies for the heterogeneous network with different levels of congestion. Another research priority is to investigate partitioning and perimeter control strategies for networks with very heterogeneous topology (e.g., non-redundant networks with limited number of connectivity points between regions of a city) or hierarchical structure (e.g., mixed networks of grid arterials and freeway systems). Perimeter control is not studied in this paper, but recent results show the applicability of the MFD concept (e.g. Geroliminis et al., 2012; Keyvan-Ekbatani et al., 2012).

Partitioning is a necessary step before the implementation of perimeter control at urban networks. The spatio-temporal propagation of congestion is highly related with the type of partitioning and the application of the appropriate control strategy. Urban systems experience highly dynamic behavior and different traffic patterns may arise for different times of day (think of morning–evening commuting patterns or stochastic variations of traffic flow). In these cases, very likely one needs to identify different optimal sets of clusters depending on these patterns. During the implementation, partitioning might have a two-step approach, an offline based on historical data and an online part based on real-time data. We can also consider two different cases for propagation of congestion. The simplest case is the one that propagation is mainly temporal. In this case the size of the congested region does not change with time and the level of congestion decreases or increases (roughly) homogeneously for the regions. Thus, a static partitioning based on historical data would be appropriate to identify

[3] For interpretation of color in Fig. 8, the reader is referred to the web version of this article.

the different regions and then apply a perimeter control logic. For example in the case of Yokohama, congestion area was well-defined and was growing evenly (Geroliminis and Daganzo, 2008; Geroliminis and Sun, 2011) so there was no need for the clustering to be dynamic. One can solve the static case for a time period rather than a single interval by applying the current methodology, but in Eq. (1) a time vector of density is needed. We have to note that if congestion propagates both in time and space, the shape of low variance regions might change with time. It might not be easy to have a region that has both a constant shape (with time) and a low scatter MFD; in this case, dynamic clustering is necessary.

Let us consider now the case where the size of the congested region changes with time. Then, partitioning should follow a dynamic approach and the offline algorithm will determine how clusters change with time. For the specific spatio-temporal regions an MFD can be estimated from the data. Thus, dynamics partitioning algorithms is a research priority. Most of the work of this paper focuses on the static partitioning, but our work provides useful insights and we currently study the dynamic case algorithms. A simple example of dynamic partitioning is the one of Section 4.2.4. As mentioned, by ‘‘boundary adjustment’’ only small perturbations of the boundaries can be captured and further research is needed towards this direction.

The developed algorithms of this paper, given their short computational time (a few seconds), can be directly applied real-time in these cases. Nevertheless, further research is needed to identify how often a partition should be adjusted. Our understanding is that dynamic clustering can be performed at a time resolution that it is smaller than the control decisions, e.g. if control decisions for traffic signals through perimeter control are made every 5 min, clustering might need to be performed every 15–30 min. This problem has a strong link with the spatio-temporal propagation of congestion in transportation networks.

Regarding the online part of the algorithm, one can check if the variance of the predefined offline clusters exceeds some threshold values, which would indicate that the network would require repartitioning. Note that these thresholds do not need to be extremely small. Geroliminis and Sun (2011) have noticed from the real network of Yokohama that a low scatter MFD exists even if there is some variance in the distribution of congestion (coefficient of variation in link density, i.e. dimensionless standard deviation divided by the mean, was around 0.25). Computationally speaking an online partitioning is not a problem, as the method developed in this paper is fast and can be applied real time if data are available. A difficulty might arise if the new partition does not exist in the historical database and the shape of the MFD cannot be directly estimated from the data. This question would require some further investigation and more empirical databases and case studies can shed further light towards this direction. Nevertheless, (Mazloumian et al., 2010) have shown that there is a low scatter relationship between the network flow and variance of link density (which expresses the spatial heterogeneity of congestion) for a given mean network density. Thus, the shape of the MFD can be estimated by integrating an analytical formulation (e.g. the one of Geroliminis and Daganzo (2008) or Geroliminis and Boyaci (2012) with the empirical spatial distribution of link density).

Another interesting research direction is to identify the monitoring needs to provide an accurate and dynamic partitioning, that will lead to development of efficient control strategies. Information in every link in a network is not necessary. Our analysis produces robust (almost identical) partitioning results if density data in 20% of the links exist. In a similar matter, MFD in downtown Yokohama was observed with flow-density data in about 20% of the links. A key point is that the monitored links create a connected graph of the region. In case of highly variable link density (e.g. high directional flows), data needs might be larger for some part of the network (especially the ones with higher spatial heterogeneity). Further investigation is needed for combination of different sensors (e.g. fixed loop detectors and mobile GPS sensors).

References

Bishop, C.M., 2007. In: Pattern Recognition and Machine Learning. Springer.

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

Carmel, D., Roitman, H., Zwerdling, N., 2009. Enhancing Cluster Labeling Using Wikipedia. In: ACM SIGIR. Boston, Massachusetts, USA, pp. 139–146.

Cour, T., Benezit, F., Shi, J., 2005. Spectral segmentation with multiscale graph decomposition. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2. Vancouver, Canada, pp. 1124–1131.

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

Daganzo, C.F., Gayah, V.V., Gonzales, E.J., 2011. Macroscopic relations of urban traffic variables: bifurcations, multivaluedness and instability. Transportation Research Part B 45 (1), 278–288.

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

Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D., 2001. A min-max cut algorithm for graph partitioning and data clustering. In: IEEE International Conference on Data Mining. Vancouver, Canada, pp. 107–114.

Gayah, V.V., Daganzo, C.F., 2011. Clockwise hysteresis loops in the macroscopic fundamental diagram: an effect of network instability. Transportation Research Part B 45 (4), 643–655.

Geroliminis, N., Boyaci, B., 2012. The effect of variability of urban systems characteristics in the network capacity. Transportation Research Part B http://dx.doi.org/10.1016/j.trb.2012.08.001.

Geroliminis, N., Daganzo, C.F., 2007. Macroscopic modeling of traffic in cities. In: Transportation Research Record 86th Annual Meeting. Washington, DC, USA.

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

Geroliminis, N., Haddad, J., Ramezani, M., 2012. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. IEEE Transactions on Intelligent Transportation Systems http://dx.doi.org/10.1109/TITS.2012.2216877.

Geroliminis, N., Sun, J., 2011. Properties of a well-defined macroscopic fundamental diagram for urban traffic. Transportation Research Part B 45 (3), 605–617.

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

Haddad, J., Geroliminis, N., 2012. On the stability of traffic perimeter control in two-region urban cities. Transportation Research Part B 46, 1159–1176.

Han, J., Kamber, M., 2006. In: Data Mining: Concepts and Techniques. Morgan Kaufmann.

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

Herman, R., Prigogine, I., 1979. A two-fluid approach to town traffic. Science 204 (4389), 148–151.

Jain, A.K., 2010. Data clustering: 50 years beyond *k*-means. In: 19th International Conference in Pattern Recognition (ICPR), vol. 31. Istanbul, Turkey, pp. 651–666.

Ji, Y., Geroliminis, N., 2011. Exploring spatial characteristics of urban transportation networks. In: The 14th IEEE Conference on Intelligent Transportation Systems (ITSC). Washington, DC, USA, pp. 716–721.

Johnson, S.C., 1967. Hierarchical clustering schemes. Psychometrika 32 (3), 241–254.

Kerner, B.S., Rehborn, H., 1996. Experimental properties of complexity in traffic flow. Physical Review E 53 (5), R4275–R4278.

Keyvan-Ekbatani, M., Kouvelas, A., Papamichail, I., Papageorgiou, M., 2012. Exploiting the fundamnetal diagram of urban networks for feedback-based gating. Transportation Research Part B http://dx.doi.org/10.1016/j.trb.2012.06.008.

Lin, Y.L., Jiang, T., Chao, K.M., 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., Geroliminis, N., Helbing, D., 2010. The spatial variability of vehicle densities as determinant of urban network capacity. Philosophical Transactions of Royal Society A 368 (1928), 4627–4648.

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

Rousseeuw, P., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20 (1), 53–65.

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