Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 23 (2017) 962–979

Transportation Research Procedia

www.elsevier.com/locate/procedia

22nd International Symposium on Transportation and Traffic Theory

Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks

Mohammadreza Saeedmanesh [a], Nikolas Geroliminis* [a]

[a] Ecole Polytechnique Federale de Lausanne (EPFL), School of Architecture, Civil and Environmental Engineering (ENAC), Urban Transport Systems Laboratory (LUTS), Lausanne, Switzerland

Abstract

The problem of clustering in urban traffic networks has been mainly studied in static framework by considering traffic conditions at a given time. Nevertheless, it is important to underline that traffic is a strongly time-variant process and it needs to be studied in the spatiotemporal dimension. Investigating the clustering problem over time in the dynamic domain is critical to better understand and reveal the hidden information during the process of congestion formation and dissolution. The primary motivation of the paper is to study the spatiotemporal relation of congested links, observing congestion propagation from a macroscopic perspective, and finally identifying critical pockets of congestion that can aid the design of peripheral control strategies. To achieve this, we first introduce a static clustering method to partition the heterogeneous network into homogeneous connected sub-regions. This method guarantees connectivity of the cluster, which eases the development of a dynamic framework. The proposed clustering approach obtains a feasible set of connected homogeneous components in the network called snakes, which represent a sequence of connected links with similar level of congestion. Secondly, the problem is formulated as a mixed integer linear optimization to find major skeleton of clusters out of this feasible set by minimizing a heterogeneity index. Thirdly, a fine-tuning step is designed to assign the unclustered links of the network to proper clusters while keeping the connectivity. The approach is extended to capture spatiotemporal growth and formation of congestion. The dynamic clustering is based on an iterative and fast procedure that considers the spatiotemporal characteristics of congestion propagation and identifies the links with the highest degree of heterogeneity due to time dependent conditions and finally re-cluster them while by minimizing heterogeneity and imposing connectivity. The developed framework can be directly implemented in a real-time framework due to its fast computation and proper integration of physical properties of congestion.

Peer-review under responsibility of the scientific committee of the 22nd International Symposium on Transportation and Traffic Theory.

Keywords: graph partitioning; congestion propagation; aggregate modeling; mixed integer linear programming; snake algorithm.

* Corresponding author. Tel.: +41 21 69 32481; fax: +41 21 69 35060.
E-mail address: nikolas.geroliminis@epfl.ch

Peer-review under responsibility of the scientific committee of the 22nd International Symposium on Transportation and Traffic Theory.
10.1016/j.trpro.2017.05.053

1. Introduction

In traffic networks unlike other physical systems, humans make complex choices in terms of routes, destinations and time of travel, which create additional complexity to the system and difficulty in accurate prediction of link level traffic states, especially for real-time applications, like network-level traffic control. Moreover, transport in cities is governed by non-linear interactions with components (users, operators, infrastructure) that influence the system and vice versa. These components also adapt their behavior and decisions when the operators intervene in the system. Additionally, different network topologies and demand profiles affect the way congestion grows in space and time. Aggregated level network modeling has recently gained a lot of interest due to all aforementioned difficulties in traffic modeling at microscopic level.

The main characteristic of traffic in urban networks is that congestion is spatially correlated in adjacent roads and it can propagate with some finite speed in time and space. In fact, once a road gets congested, the probability of neighboring roads to be congested is higher than farther roads (i.e. congestion expands by propagating to neighboring roads). This spatial correlation of urban traffic congestion allows describing heterogeneous network as multiple relatively homogeneous and spatially connected regions Ji et al. (2014). This partitioning facilitates discovering the congested areas in an aggregated level. It also helps to observe well-defined Macroscopic Fundamental Diagram (MFD) relating space-mean flow and density in homogeneous areas of a city. The concept of MFD can be utilized in design real-time traffic control schemes specifically hierarchical perimeter control approaches to protect regions with high level of congestion and avoid gridlock conditions. While dynamic perimeter control for multi-region statically partitioned network has shown to decrease the level of congestion compared to traditional local traffic control strategies (see for example Ramezani et al. (2015)), more advanced techniques might be needed for networks with variant demand profiles. In this case, a dynamic clustering coupled with perimeter control strategies might be necessary. Dynamic clustering can also be important beyond the scope of real-time traffic control, as for example for planning studies, emission analysis, a dynamic user route information or even for advancing the operations of public transport systems.

The problem of clustering in urban traffic networks has been mainly studied in static framework by only considering traffic conditions at a given time. Nevertheless, it is important to underline here that traffic is a strongly time-variant process and it needs to be studied in the spatiotemporal dimension. Investigating the clustering problem over time help us understand and reveal the hidden information during the process of congestion formation and dissolution. The primary motivation of the paper is to study the spatiotemporal relation of congested links, observing congestion propagation from a macroscopic perspective, and finally identifying critical congestion regimes to aid the design of peripheral control strategies and improve mobility. This is not a straightforward task as transport networks despite spatial correlations in congestion are in principle heterogeneous due to road hierarchy and spatial distribution of demand. Thus, it is not always possible to treat congestion as a continuum in space. An example is directional flows towards a city center in the morning peak, where the same roads are highly congested in the one direction and uncongested in the other. The dynamic clustering framework will be capable of replicating how clusters expand or shrink in the process of congestion formation and dissolution. Moreover, it will be able to find new pockets of congestion and merge clusters with similar traffic conditions. In this framework, we will be able to chase where congestion originates and how traffic management systems affect its formation and the time it finishes. To achieve these goals, first we formulate the problem of partitioning networks to a desired number of regions as mixed integer linear optimization (a three-step framework). As it will be explained later, connectivity of clusters is explicitly enforced by imposing some constraints and the homogeneity of clusters is maximized in the objective function. In this framework, we can also specify characteristics such as compactness, size, *etc* for clusters.

There is a strong effort in the last 30 years for traffic flow models in one-dimensional traffic systems (see Helbing (2001) for an overview). With respect to network level, it has been observed with empirical and simulated data (see for example Geroliminis and Daganzo (2008), Buisson and Ladier (2009), Gayah and Daganzo (2011), Mahmassani et al. (2013), and many others) that a well-defined curve (MFD) exists between space-mean flow and density in networks with homogeneous traffic conditions. The idea of MFD has been re-initiated in Mahmassani et al. (1987) and Daganzo (2007) despite the first insight to the problem comes from Godfrey (1969) and Herman and Prigogine (1979). Analytical approximations of MFDs can be found in Laval and Castrillón (2015), Daganzo and Geroliminis (2008), Leclercq and Geroliminis (2013). Spatial heterogeneity in the distribution of congestion can significantly influence the shape and the scatter in MFD curves (see Buisson and Ladier (2009), Geroliminis and Sun (2011), Daganzo et al. (2011), Knoop and Hoogendoorn (2013), Saberi et al. (2014), and others). In case of heterogeneity, network needs to be divided into smaller homogeneous sub-regions where an MFD can be defined for each sub-region (see Ji and Geroliminis (2012)). Clustering techniques for other areas of transport science can be found in Han and Moutarde (2013), Hans et al. (2014) and Anbaroglu et al. (2014). The effect of route choice in the MFD has been investigated in Mahmassani et al. (2013), Yildirimoglu and Geroliminis (2014), Leclercq et al. (2015). A detailed literature review of network modeling and control can be found for example in Haddad et al. (2013), Mahmassani et al. (2013), Keyvan-Ekbatani et al. (2015), and elsewhere.

There is a vast literature on studying clustering algorithms in several fields such as community detection Lancichinetti and Fortunato (2009), data-mining, image segmentation Shi and Malik (2000), and sensor networks. In sensor networks, the sensors are grouped into a few clusters, within each a cluster-head is assigned to communicate with local sensors and transmit data to the base station. The objective is to find a configuration to lower the energy consumption and number of transmitted messages from sensors to base station. Perhaps the earliest of the clustering methods is the identifier-based heuristic called the linked cluster algorithm Baker and Ephremides (1981), which elects sensor to be a cluster-head if the sensor has the highest identification number among all sensors within one hop of its neighbors. The connectivity-based heuristic of (Gerla and Tsai, 1995) selects the sensors with the maximum number of neighbors (i.e., highest degree) to be cluster-heads. Please refer to Afsar and Tayarani-N (2014) for a detailed survey on the existing methods on wireless sensor networks. Indeed, spatial clustering is a well-studied problem in other domains of quantitative sciences. Depending on the type of data, e.g. climate zoning, regionalization, geography, computer science, etc., different approaches including density-based, distance-based, hierarchical clustering have been proposed. In these problems known as contiguity-constrained clustering, researchers are interested in aggregating spatial data (units) into spatially connected homogeneous regions. Spatial contiguity [1] can be either indirectly fulfilled or explicitly imposed within the algorithm. The first category consists mainly of conventional clustering methods in which spatial information is incorporated in the classification data. The method presented in Ji and Geroliminis (2012), building on the established Normalized Cut algorithm Shi and Malik (2000), indirectly imposes connectivity by assuming similarity only between neighboring roads. However, in non-grid networks, this method tends to partition from the locations where network has low connectivity regardless of the level of congestion. Moreover, it requires a connected graph of the network and missing values or malfunctioning detectors might create difficulties in application of the method. This is an important drawback for transport networks with non grid-type structures.

Later, 'Snake' method was proposed in Saeedmanesh and Geroliminis (2016) which tried to overcome the aforementioned difficulties and develops a clustering methodology that is able to find directional congestion within a cluster and has good performance for networks with low connectivity. This paper utilizes, a non-linear optimization technique known as Symmetric Non-negative Matrix Factorization (SNMF) to assign links to proper clusters with high intra-similarity and low inter-similarity. SNMF partitions the data by providing a lower rank approximation of the similarity matrix. A similarity measure is defined between each pair of the roads in the network, based on a newly introduced method, defined as 'snake'. Each sequence of roads is built by starting from a single road and iteratively adding one adjacent road based on its similarity to join previously added roads in that sequence to create a snake. While this framework outperforms previous methods in static clustering, it cannot directly be applied in a dynamic framework for the following reasons: (i) SNMF is a non-convex, non-linear technique, which has some compurational burden and cannot be easily solved in a real-time framework (that a dynamic clustering would require), (ii) spatiotemporal correlations cannot be easily integrated to guarantee connectivity (even if in the static case, clusters are quite connected) and (iii) similar to (ii), an iterative procedure (as the one developed in the current paper) cannot be conceptualized in the similarity matrix and (iv) the number of clusters have to be externally imposed, while we would like to give the flexibility to the developed framework to change the number of clusters based on the level of congestion and heterogeneity.

The second group of methods where contiguity is explicitly imposed, can be broken into two major categories: (i) heuristic Duque et al. (2007); (ii) exact optimization models Shirabe (2005). The heuristic techniques either start by considering each road as one cluster and merge them; or start with the whole network as one cluster and

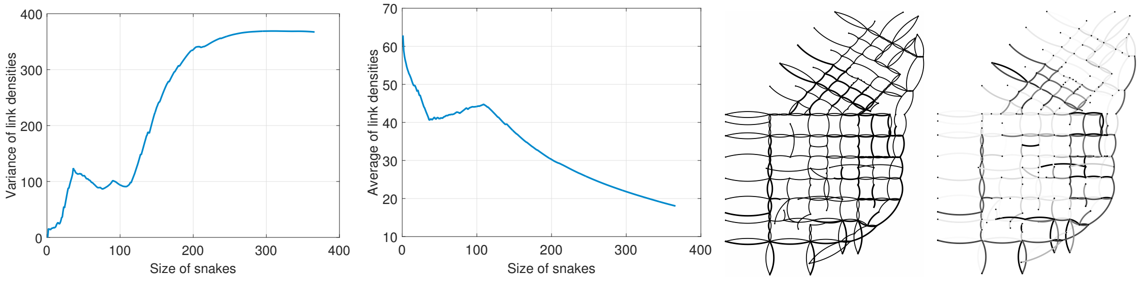
[1] Network can be represented as a graph where each road is a node and an edge exist between nodes (roads) that are connected to a common intersection. A sub-graph is called connected when there exists a path between each pair of nodes through the existing edges

split it to many parts until a level of homogeneity is reached. In Etemadnia et al. (2014), an exact formulation is provided to partition a typical urban transportation network into balanced size regions where the inter-flow among the resulting sub-networks is minimized. Two heuristics are developed to solve the problem in a short time for big networks and the methodology is applied to Dallas metroplex. Note that, the aim of clustering in our study is to find homogeneous regions with minimum heterogeneity in congestion level rather than minimizing the flow between sub-regions. Moreover, we provide the exact solutions at each step (i.e. with zero optimality gap). In this paper we follow the 2nd group of methods (where connectivity is explicitly imposed), as dynamic clustering might create infeasibilities in the 1st approach when congestion varies in space and time.

It is important to underline here that traffic is a strongly time-variant process and it needs to be studied in the spatiotemporal dimension. In Ji et al. (2014), authors visualize the formation of congestion using real data from probe taxis in Shenzhen, China. They utilized the concept of Maximum Connected Component (MCC) to identify connected congested areas in the network and track their evolution over time. However, this approach is a constructive and not an optimization method and depends on the selection of threshold differentiating between congested and uncongested roads. In this study, we propose a clustering framework which partitions the heterogeneous networks into connected and relatively homogeneous areas. Moreover, we extend the approach to capture the spatiotemporal growth (dissolution) of the congestion during the on-peak (off-peak). Note that the proposed dynamic approach utilizes similar logic as the static approach and they are fully consistent. More specifically, it identifies in a dynamic framework the links that mostly changed their level of congestion and tries to re-cluster them by minimizing heterogeneity and guarantee connectivity. This procedure can be applied iteratively from time to time and identify the spatiotemporal propagation of congested pockets. The remainder of the paper is organized as follows: Section 2 introduces the proposed methodological framework for static clustering and explains each algorithmic step separately in details. Section 3 extends the approach to be applied in cases where the traffic conditions varies over time. In Section 4, results of the proposed algorithm are presented and compared with other approaches. The case study is a grid type medium size network with simulation data and time dependent conditions from uncongested to highly congested (San Francisco with about 400 links). The Paper concludes in the last section by identifying current and future research directions.

2. Methodological Framework (Static clustering approach)

The objective of this section is to propose a methodology to partition heterogeneously distributed congested networks into a few number of homogeneous sub-regions with a guarantee of spatial connectivity and a minimization of a heterogeneity metric. In fact, homogeneity and connectivity constraints are two conflicting criteria that need to be taken into account at the same time. In the proposed approach, we consider heterogeneity as a main objective function and explicitly impose spatial connectivity as a constraint in different steps. To achieve connectivity in the clusters, we limit the search space of the optimization problem to a set of predefined homogeneous areas that are connected by construction. As the first step of the algorithm, we aim to obtain homogeneous connected areas around different roads in the network by growing "snakes", which was initially introduced in (Saeedmanesh and Geroliminis, 2016). Each snake has a form of array that is built by attracting the most similar adjacent link iteratively. Note that, at each step, the most similar link to the snake is the adjacent link with the closest density value to the average density values of the links in the snake. This procedure identifies interesting patterns as the variance grows in a non-predictable way with the size of the links. This observation is related to the fact that some parts of the graph are more similar than others and if a large number of dissimilar links is added, high variance is unavoidable. Note that each snake at any given size represents a homogeneous area which is connected by construction. These snakes create a set of feasible solutions which is utilized in the current paper as a domain (search space) for the optimization framework in the next step of the algorithm. In the second step of the algorithm, the main skeleton of clusters is extracted from the aforementioned set of snakes. The aim is to select a few distinct snakes (i.e. snakes with small overlaps) out of the whole set that cover main part of the network and truncate them from the length where they are quite homogeneous. The problem is formulated as a Mixed Integer Linear Programming (MILP) with an objective of minimizing heterogeneity (variance) of the obtained clusters. The advantage of such formulation is that in the OR literature exist fast toolboxes total with well-established techniques to solve these types of problems close to optimality (i.e. the optimality gap, measures how close the solution is to optimal, is known at each step). Finally, a fine tuning step is applied to associate remaining



(a) (b) (c) (d)

Fig. 1. (a) Evolution of the links density variance v.s size of the snake; (b) Evolution of the link average density v.s size of the snake; (c) Snake with size 110 (based on Saeedmanesh and Geroliminis (2016)); (d) Grey-scale representation of congestion in the network.

unassigned links with proper clusters while keeping connectivity inside the clusters. As mentioned in the introduction, the main difference of this work from (Saeedmanesh and Geroliminis, 2016) is the way we enforce connectivity during the procedure of growing snakes while in (Saeedmanesh and Geroliminis, 2016), connectivity of the regions cannot be guaranteed and is affected by some parameters (e.g. parameter ϕ in calculating the similarity between snakes). Different steps of the proposed partitioning framework are described in details in the following sections.

2.1. Running snakes (obtaining the feasible domain)

The aim of this step is to provide a subset of feasible solution representing connected homogeneous areas covering different parts of the network . Output of this step will be utilized later as a search space (feasible solution domain) in the optimization framework (2nd step). Basically, in the first step (following (Saeedmanesh and Geroliminis, 2016)), a sequence of links is built iteratively for each individual link in the network with an objective to minimize the variance of all the chosen links density (or speed). It can be considered as a "snake" that starts from a link and grows by attracting the most similar adjacent link in each step. In other words, a link with the closest density value to the average density value of the existing links in the snake is added at each iteration. Obviously, this step needs graph information about connectivity as only adjacent links could be added at each step. Note that, each snake represents a homogeneous connected area around a road that is obtained by absorbing neighboring roads. As congestion propagates from/to neighboring links in traffic networks, grabbing neighboring roads performs quite well in identifying the correct direction for the snakes to grow.

Evolution of the variance curves of these sequences with the size of the links reveals interesting non-linear patterns. Sharp jumps (Fig. 1(a)) can be observed after some points (e.g. around size 110 in the presented snake) in the variance diagram which reveals very well the fact that some parts of the graph are more similar than others. At that point, snake starts adding links with lower density and average density has been sharply decreased (see Fig. 1(b)). This highlights the necessity of choosing proper snakes and truncate them at appropriate size. We will explain in details how the problem is formulated to identify proper snakes. Figure 1(c) depicts the corresponding snake with size 110. By comparing Fig. 1(c) and 1(d), it can be observed that this snake covers the congested parts of network. This observation justifies the necessity of stopping snakes and truncating them at certain points.

2.2. Snake segmentation

The aim of the second step is to find the main skeleton of each cluster from a feasible set, consisting of homogeneous connected areas (snakes), obtained in the first step. In other words, we are interested in finding homogeneous parts of a few snakes to cluster the network. It is clear that connectivity constraint holds for snake i with a given size N_i holds if the first N_i elements of the sequence are selected and assigned to a cluster. Conceptually speaking, given a desired number of clusters N_c, the algorithm finds a subset of N_c snakes (each snake represents a cluster) with appropriate size (number of elements), so that they cover most (if possible the whole) of the network and minimize heterogeneity index (variance) in different clusters. The optimization problem is formulated with a linear objective function along with some linear inequality constraints. Note that decision variables are both integer and continuous.

Table 1. Set of variables, indices, and parameters for snake segmentation step

Decision variables	
x_{ik}	Binary value indicating if road k is selected in array i or not
x'_k	Binary variable indicating if link k is assigned to at least one cluster or not
x''_k	Binary variable indicating if link k is assigned to multiple clusters
t_i	Weighted variance of selected link density values in snake i
Parameters and sets	
$\mathcal{L}$	Set of all roads in the network
N	Number of roads in the network
N_s	Number of snakes in the network
N_c	maximum number of clusters
N_{min}	Minimum number of roads in selected snake
$R_i(k)$	Element of cell k in snake i
$R_i^{-1}(k)$	Index of a cell in snake i that contains element k
$\text{var}(S_{\{i,k\}})$	Variance of the link densities for the first k elements in snake i
a'	Minimum percentage of links that should be associated with at least one cluster
a''	Maximum percentage of links that could be associated with multiple clusters

Different types of constraints are specified to impose connectivity and minimum size for selected snakes. In this framework, there is a possibility that some of the links appear in more than one cluster although distinct snakes can be chosen. This is because in the first steps snakes are growing independently from each other, so a link might belong to more than one snakes. To avoid having clusters with high overlaps (same links in different snakes), a set of constraints is specified that restricts the maximum allowable number of such links (overlapping constraint). As the optimization tries to minimize heterogeneity (weighted variance) for selected snakes, many links will remain unassigned in this step. Therefore, we should determine the percentage of network that is needed to be covered in this step (coverage constraint). Note that, it might not be possible to cover all the network (assign all the links in the snakes) in this step as we have overlapping and minimum size constraints. Ideally, overlapping constraint should be set up to zero and coverage constraint to 100%, but this might make the optimization problem infeasible as snakes grow independently. A fine-tuning step is applied afterwards (see section 2.3) to guarantee that all roads are assigned to exactly one cluster. We first define the sets and indices used to describe the model as well as the variables and parameters (see Table 1); then detailed mathematical optimization formulations are presented.

Optimal clusters should have small overlap, high covering rate, and low variance. The objective function (1) is defined as the summation of weighted variances of different snakes (clusters) which has to be minimized. The weighted variance of a cluster is a non-linear function of selected links; however, as the sequence of links is known in each snake, we are able to write it in a linear form by defining auxiliary variables t_i which is equal to the weighted variance of snake i at the optimal solution (Eq. (8) is a binding constraint which determines the lower bound for variable t_i). The mathematical problem is formulated as follows[2]:

$$\min \sum_{i=1}^{N_s} t_i \tag{1}$$

$$x_{iR_i(k+1)} \leq x_{iR_i(k)} \quad \forall i = \{1, \ldots, N_s\}, \quad \forall k \in \mathcal{L} - \{N\} \tag{2}$$

$$x'_k \leq \sum_{i=1}^{N_s} x_{ik} \leq N_s \times x'_k \quad \forall k \in \mathcal{L} \tag{3}$$

$$\sum_{k \in \mathcal{L}} x'_k \geq a' \times N \tag{4}$$

$$x''_k \leq \sum_{i=1}^{N_s} x_{ik} \leq N_s \times x''_k + 1 \quad \forall k \in \mathcal{L} \tag{5}$$

[2] A simpler formulation of this type was initially presented in a conference paper by the same authors in (Saeedmanesh and Geroliminis, 2015)

$$\sum_{k\in\mathcal{L}} x_k^{''} \leq a'' \times N \tag{6}$$

$$\sum_{k\in\mathcal{L}} x_{ik} \geq N_{\min} \times x_{iR_i(1)} \quad \forall i = \{1,\ldots,N_s\} \tag{7}$$

$$(x_{ik} \times R_i^{-1}(k) + x_{ik} - \sum_{k'\in\mathcal{L}} x_{ik'}) \times (R_i^{-1}(k) \times \mathrm{var}(S_{\{i,R^{-1}(k)\}})) \leq t_i \quad \forall i = \{1,\ldots,N_s\}, \quad \forall k \in \mathcal{L} \tag{8}$$

$$\sum_{i=1}^{N_s} x_{iR_i(1)} = N_c \tag{9}$$

$$x_{ik} \in \{0,1\}, \quad x'_k \in \{0,1\}, \quad x''_k \in \{0,1\}, \quad t_i \in \mathbb{R} \tag{10}$$

Constraints (2) ensure that selected links in each are connected. Since, snakes grow by adding adjacent links iteratively, connectivity is guaranteed if the first n consecutive cells are picked from a selected snake. For a given link to be picked in a sequence (snake), constraint (2) requires its previous element to be selected. Constraint (4) ensures that a minimum percentage of the links in the network, denoted by a', will be assigned at least to one of the clusters. To achieve this, a binary auxiliary variable $x_k^{'}$ is defined in (3) for each individual link k that indicates if it is selected in at least one snake or not. To have a minimum percentage of links associated with clusters, it is likely that some links appear in more than one snake (multiple clusters). Binary auxiliary variable $x_k^{''}$ in constraints (5) indicates whether a link is assigned to more than one cluster or not. Constraint (6) restricts percentage of the links (a'') in the network that are allowed to appear in more than one cluster. Constraint (7) ensures that a selected snake has at least $N_{\min}$ links (i.e. The first $N_{\min}$ links have to be picked in a selected snake). $N_{\min}$ is an input to the optimization problem and has to be defined by the user. Constraints (8) are binding constraints for t_i which determine their values at optimal point. For a selected snake with a certain size, the left hand side (LHS) of this constraint gets its maximum value for the last selected link selected in this array. Basically, for the last selected element in the array, the LHS of this constraint takes the value equal to the weighted variance of that snake while for the rest of elements it takes values less or equal than zero. Constraint (9) defines the number of snakes (clusters). Note that, a unique snake is defined from each link, so the first link fully determines the snake.

2.3. Fine tuning

After completing the first two steps, a fine tuning approach is applied to assign the remaining links to proper clusters. Fig. 2 schematically depicts results of the first two steps for a simple network partitioned into three clusters. Besides the links (points) that do not belong to any of the clusters (black dots), some links are assigned either to one cluster (blue dots) or multiple clusters (red dots). Connected blue points are considered as main part (core) of the clusters and there is no need to reassign them; however, red and black elements has to be assigned in this step. Links belong to either no clusters or to more than one cluster are considered as decision variables in the fine-tuning step. Decision variables and parameters used in the current optimization step are defined in Table 2.

Optimal clusters should have low intra-disimilarity and inter-similarity. Hence, the objective function in Eq. (11) is written as the summation of intra-dissimilarity and inter-similarity of clusters which has to be minimized. Note that total number of terms in the objective function should not be changed by the size and number of clusters in order to have precise and fair comparison. Simply speaking, if N is the total number of links, there are in total N^2 terms representing all possible pairs[3]. In this approach, for each pair of links, either intra-dissimilarity or inter-similarity is calculated in objective function, depending on the clusters they belong to. Note that, the objective function in

[3] Number of terms in the objective function depends on the size of clusters if we only consider one of the terms (similarity or dissimilarity) in the objective function. For instance, assume that objective function only has the intra-dissimilarity part (first summation in the RHS of Eq. 11; then it contains $N_1^2 + N_2^2$ terms where N_1 and N_2 denote size of first and second cluster respectively. Apparently, this number is changed by the size of clusters and optimization approach might tend to make this number small by putting $N_1 = N_2$.

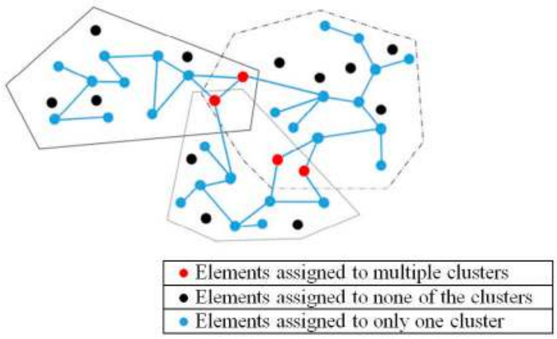


Fig. 2. Schematic result of network partitioning after the first two steps of the algorithm

Table 2. Set of variables, indices, and parameters for fine-tuning step

Decision variables	
y_j	Distance of node 'j' to the center of its cluster
x_{ijk}	Binary variable indicating if node k is connected to the center of cluster 'i' via its neighboring node j or not
h^i_{kl}	Binary variable indicating if roads k and l are both in cluster i or not
x_{iok}	Binary variable indicating if node k is selected as a center of cluster 'i' or not
Parameters and sets	
$\mathcal{L}$	Set of all roads in the network
$\mathcal{N}(k)$	Set of neighboring roads of road k
N_c	number of clusters
N_{min}	Minimum number of links in selected clusters(minimum size)
$d(l,k)$	Dissimilarity between density of roads l and k
D_l	Density of cars in road l
$\bar{d}$	Average dissimilarity of the data

Eq.1 is not appropriate here as the sequence of adding roads is not predefined like in the case of having snakes. The mathematical problem is formulated as follows:

$$\min \theta = \sum_{i=1}^{N_c}\sum_{l=1}^{N}\sum_{k=1}^{N} h^i_{lk} d(l,k) + (1-h^i_{lk})(\bar{d} - d(l,k)) \tag{11}$$

$$\begin{cases} a. \quad x_{ik} + x_{il} - h^i_{kl} \leq 1 \\ b. \quad h^i_{kl} \leq x_{ik} \\ c. \quad h^i_{kl} \leq x_{il} \end{cases} \quad \forall k,l \in \mathcal{L}, \forall i = \{1,\ldots,N\} \tag{12}$$

$$\sum_{i=1}^{N_c} x_{ik} = 1 \quad \forall k \in \mathcal{L} \tag{13}$$

$$\sum_{l\in\mathcal{N}(k)} x_{ilk} = x_{ik} \quad \forall i = \{1,\ldots,N_c\}, \forall k \in \mathcal{L} \tag{14}$$

$$x_{ilk} \leq x_{iol} + \sum_{m\in\mathcal{N}(l), m\neq k} x_{iml} \quad \forall i = \{1,\ldots,N_c\}, \{\forall l,k \in \mathcal{L} \mid k \in \mathcal{N}(l)\} \tag{15}$$

$$y_k \geq y_l - \left((1 - \sum_{i=1}^{N_c} x_{ilk}) \times N\right) + 1 \quad \{\forall l,k \in \mathcal{L} \mid k \in \mathcal{N}(l)\} \tag{16}$$

$$\sum_{i=1}^{N_c}\sum_{k=1}^{N} x_{iok} = N_c \tag{17}$$

$$\sum_{k=1}^{N} x_{iok} = 1 \quad \forall i = \{1, \ldots, N_c\} \tag{18}$$

$$N_{min} \leq \sum_{k=1}^{N} x_{ik} \quad \forall i = \{1, \ldots, N_c\} \tag{19}$$

$$x_{iok} \in \{0,1\}, \quad x_{ilk} \in \{0,1\}, \quad 0 \leq y_k \leq N, \quad x_{il} \in \{0,1\}, \quad h^i_{lk} \in \{0,1\} \tag{20}$$

$$\bar{d} = \sum_{l=1}^{N}\sum_{k=1}^{N} d(l,k)/N^2, \qquad d(l,k) = |D_l - D_k| \tag{21}$$

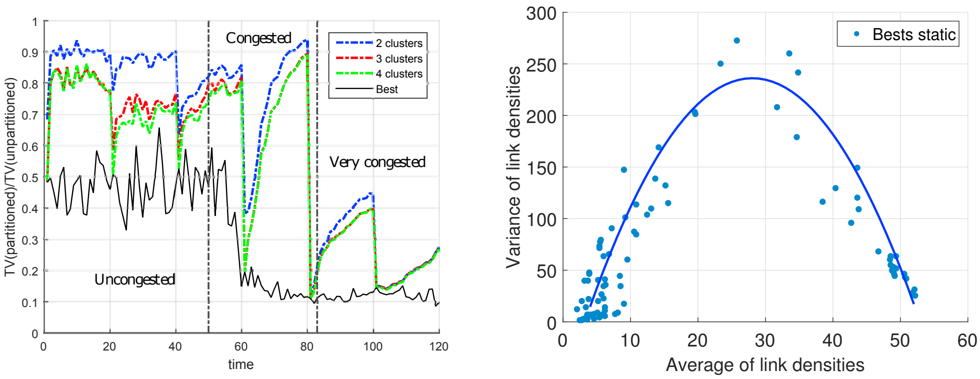
As it was stated before, objective function in Eq. (11) has both intra dissimilarity ($d(l,k)$) and inter similarity ($\bar{d} - d(l,k)$) terms. Note that, the dissimilarity between nodes l , k and average dissimilarity of the data are defined as Eq. (21). Constraints 12(a-c) ensure that binary variable h^i_{lk} takes value 1 if and only if two roads l and k belong to the cluster i.

The important challenge of fine-tuning method is to model connectivity in graphs and enforce clusters to be connected. Here we present how this is succeeded. Note that in our problem each road is represented by a node in graph. Given a connected sub-graph with N nodes and an arbitrary node k in that set, we can always find at least one ordered tree (arborescence) connecting the root node to all the nodes in that sub-graph. Variable x_{ilk} indicates whether the center (root node) of cluster i is connected to link k through its neighboring link l (k and l are neighboring roads) or not. Note that, $x_{ilk} = 1$ also implies that nodes l and k both belong to cluster i. In this case, node l is considered as a parent node for child node k. Constraints (13)-(14) together enforce that each node only belongs to one cluster and it has one and only one parent node. Constraint (13) establishes that the center can be connected to node k passing from node l only if the center is already connect to node l. Decision variable y_k can be interpreted as the level of node k which represents how far that node is from the center. Apparently, for each node, this value depends on its path to the center. Constraint (16) insures that the level of a child node k should be at least one unit bigger than its parent l. Therefore, y_k represents the graph distance (i.e. number of edges in the obtained ordered tree) between node k and the center of its cluster. Note that constraints (13) to (15) are necessary but not enough to impose connectivity. The counter-example is to have multiple disjoint sub-graphs in which at least one of them is a directed cycle graph. Adding constraint (16) helps to avoid having directed cycle graph and ensures connectivity. Constraints (17)-(18) enforce the number of clusters to be equal to N_c. Indeed, number of clusters is equal to the number of points selected as center. Note that each cluster has only one center. Constraint (19) enforces each cluster to include have a minimum number of elements denote by $N_{\min}$.

Equations (11) to (21) represent the general formulation of the clustering problem where there is no information about any roads in the network. This formulation requires the definition of many auxiliary variables h_{lk} which increase the size of the problem. However, in the optimization problem Eq. (1) to (10), the majority of the roads is assigned to the proper clusters and also number of clusters is given. This significantly reduces the size of the optimization problem as we only have variables for remaining unassigned roads. It will be explained in details in the next section how the above formulation is utilized for dynamic clustering framework in consistent way with the static approach.

3. Methodological Framework (Dynamic clustering approach)

Traffic congestion spreads in time and space in a non-trivial way. This raises a question about how clusters evolve over time and how often reclustering has to be performed. By applying static clustering from time to time, we would be able to decrease the heterogeneity. However, if the partitioning time interval is large, the clusters might become heterogeneous after some point between two partitioning intervals. Considering the fact that congestion propagates spatiotemporaly with finite speed, clustering results have correlation in time. Moreover, to study how congestion propagates and to what speed and direction, we should not look at the problem independently over time. Hence, we aim to describe mathematically the process of congestion formation (on-peak) and dissolution (off-peak). Conceptually speaking, the process of congestion formation (dissolution) can be modeled as expansion (shrinking) of



(a) (b)

Fig. 3. (a) Efficiency of static partitioning over time; (b) Well-defined relation between average and variance of optimal clusters.

an individual congested area or merging (splitting) different congested areas depending on their locations. This study focuses on the expansion and shrinking of congested areas to track spatiotemporal growth of congestion.

While the main numerical results of the paper are presented in Section 4, we provide here some preliminary analysis that motivates our methodological framework. To provide a first insight, we investigate static partitioning of the network based on Eq. (1) - (21) every 20 time intervals (40 minutes) and keep the clusters fixed between two partitioning times. For different number of clusters (2-4), a normalized total variance metric $\mathrm{TV_n}$ is defined as the ratio of total variance calculated for partitioned to unpartitioned case:

$$\mathrm{TV_n} = \frac{\mathrm{TV_{partitioned}}}{\mathrm{TV_{unpartitioned}}} = \frac{\sum_{i=1}^{N_c} N_i \times \mathrm{var}(C_i)}{N \times \mathrm{var}(C)} \tag{22}$$

where N_c and N_i denote number of clusters in the network and number of links in cluster C_i respectively. The normalized $\mathrm{TV_n}$ depicted in Fig. 3(a) is utilized to evaluate the efficiency of the clustering at different time intervals. We can easily see the increasing trend in between two consecutive partitioning time which shows the necessity of applying dynamic clustering over time. Note that, the black curve depicts the best case where the partitioning algorithm is applied at each time interval. Note that the value of $\mathrm{TV_n}$ is much smaller for congested conditions than uncongested (0.2 vs. 0.5). This is because the network is uncongested at the beginning and the network is almost homogeneous. As traffic congestion forms in different parts, heterogeneity of the network increases and a proper clustering can significantly improve the heterogeneity (i.e. clustering decreases TV significantly). It is worth mentioning that, the difference between best case and constant clustering is bigger in congested conditions than uncongested. Consequently, during congestion formation, clusters need to be updated more frequently. Nevertheless, while these results highlight the need for dynamic clustering, it is important to develop a technique, which utilizes the information of the previous time interval cluster. If dynamic clustering is performed independently at each time interval (or every some amount of time) some issues might arise. Firstly, propagation of congestion might be invisible as an optimization procedure might create very different clusters at consequent time intervals, just because they have very similar values of the objective function. Secondly, traffic management applications (like perimeter control) might experience oscillations and implementation difficulties if perimeters are changing very fast. Thirdly, we should not also ignore that static clustering has some computational burden and it cannot be applied very frequently (e.g. every minute). Here we present how space and time correlations across clusters are captured with a method building in the framework of the previous section (static partitioning).

Given the fact that congestion propagates with finite speed from adjacent roads, we expect to achieve clustering with low variance at time $t + \Delta t$ and track the congestion propagation by only changing a small percentage of each cluster at time t. Given the clusters at time t, we aim to find the best snake with a certain size inside each cluster, as explained in section 2.2, using new traffic measurements at time $t + \Delta t$. This snake represents the connected area in the cluster which still remains homogeneous. The clustering assignment of these roads remains unchanged. Note that, at this step snakes are only allowed to grow in their own clusters. By applying this approach, we find the core parts of the clusters that remain unchanged (i.e. these parts have low variance in the next time interval). To achieve this, we utilize the formulation in section 2.2 with the following modifications: i) constraints (3) to (6) always hold as snakes only grow inside different clusters and they do not have any overlap; ii) constraint (9) is replaced by the following constraint (23) which ensures one and only one snake is selected from each cluster:

$$\sum_{j \in C_i} x_{jR_j(1)} = 1 \quad \forall i = \{1, \ldots, N_c\} \tag{23}$$

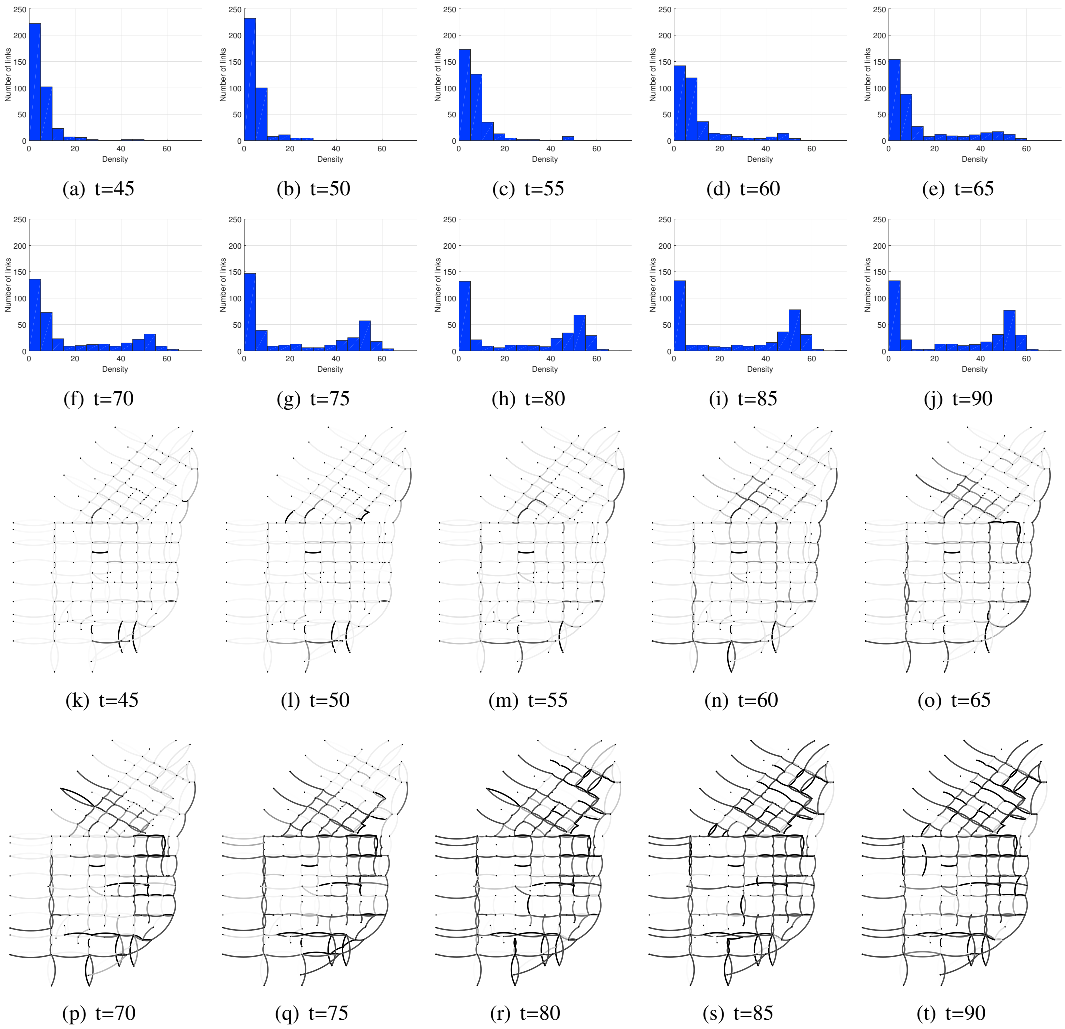
where C_i is the set of roads in cluster i. Finally, by applying the fine-tuning approach (see Eq. (11)-(21)), the remaining links are assigned to appropriate clusters.

While it is expected that the described framework (based on spatial correlations) will provide an efficient approach for dynamic clustering, new conditions in the network (e.g. a strong change of the O-D patterns) might require a full reclustering of all roads (i.e. applying static clustering without considering temporal correlation). Our conjecture is that under high quality clustering, the variance of link densities of a cluster (var(C_i) in Eq. (22) has a well defined relationship with the space mean density of that cluster. Thus, there is a fundamental relationship (similar to MFD) between spatial heterogeneity of congestion and the average congestion level. Empirical data from Yokohama confirm the above conjecture for the homogeneously congested city center (single cluster), even for different O-D demands (see Fig. 4 of Geroliminis and Sun (2011)). The authors found a unimodal relationship with low scatter, where variance increases with the average density up to a critical value and then decreases as the level of congestion is high. By analyzing over simulation a large number of clustering results for various number of clusters, different times and different demand profiles (based on the static clustering of Eq. (1)-(21)) we observe a similar MFD-type relation in Fig. 3(b). We expect that when a clustering is of low quality (e.g. due for example to not frequent update or very different conditions occur), a point above the functional form of Fig. 3(b) will be observed. Indeed, this is the case as we will show in the next section and such an observation triggers a full reclustering from scratch. As explained before, this action should not happen very frequently if the update method of this section performs well. Further empirical observations from various network structures can provide further evidence for the existence of Fig. 3(b) relation in reality.

The computational time of the optimization in different steps (specifically fine-tuning step) specifies the frequency at which the clusters can be updated. As we only re-assign a small percentage of the roads, the fine-tuning step in our case study can be solved in a reasonable time. Note that, the provided solution at each step is optimal (i.e. zero optimality gap is obtained). The clustering algorithm and mathematical models were implemented in MATLAB and GUROBI solver is used for solving MILPs. In the results section, we will provide the standard deviation and average computational time for the fine-tuning.

4. Case study and Results

The case study is a 2.5 square mile area of Downtown San Francisco (Financial District and South of Market Area) modeled and calibrated in a microsimulator. The network includes about 100 intersections with link lengths varying from 100 to 400 m. Traffic signals are all multiphase fixed-time operating on a common cycle length of 100 s for the west boundary of the area and 60 s for the rest. A 4hr time-dependent traffic demand (120 time intervals of 2 minutes) is applied to this network, which produces different spatial and temporal levels of congestion. To have a comparable congestion indicator, for each link in the network, the average density [veh/(km.lane)] is calculated based on Edie's definitions (Edie, 1963). Figure 4 illustrates the network and level of congestion in 10 different times in a gray-scale format. Note that, in all the graphs, the time values refer to different interval of 2 minutes (i.e. t=30 refers to the time 60 minutes after the simulation starts). Note also that, arcs represent different directions in two way roads (counter clockwise direction). The propagation of congestion can be easily seen in these figures. At the initiation of demand increase, there are 3 small pockets of congestion at time t=50 with only a few congested links. It is clear that these pockets grow in different dimensions and at different rates and around time t=70 they merge into a single connected component of congestion which continues to grow until the network almost reaches a state of gridlock at time t=90. The objective of dynamic clustering is to capture these dynamics of congestion in an aggregated manner, identifying the main components of congestion that can also ease the development of active management strategies. We should emphasize that as explained from the mathematical formulation in the previous section, two important constraints



(a) t=45 (b) t=50 (c) t=55 (d) t=60 (e) t=65
(f) t=70 (g) t=75 (h) t=80 (i) t=85 (j) t=90
(k) t=45 (l) t=50 (m) t=55 (n) t=60 (o) t=65
(p) t=70 (q) t=75 (r) t=80 (s) t=85 (t) t=90

Fig. 4. Histogram of road densities (a-j) and gray-scale representation of the network during congestion formation (k-t).

related to the physical properties of aggregated modeling should hold for each cluster: (I) the clusters should not be extremely small (constraint 19); (II) each cluster should contain a connected sub-graph of the network. We will refer to these gray-scale graphs in various parts of the result section for comparison.

4.1. Static clustering results

This section presents the results of the static partitioning framework applied in the network of San-Francisco. We select the time t=70 at which there is a high range of average density values and sub-regions with different levels of congestion could be easily seen in different locations (see Fig. 4(p)). It could be also seen in the histogram of link densities (Fig. 4(f)) that the network is heterogeneous with different levels of congestion. Moreover, there are some bi-directional roads with only one congested direction, which will facilitate to test the performance of model for detecting directional congestion.

We apply the proposed clustering algorithm in San-Francisco network during the peak period of congestion and examine the performance with different number of clusters. A set of all different snakes is obtained in the first step and an optimization is solved in the second step to select snakes with proper size and location. The selected values for parameters a', a'', and $N_{\min}$ are 0.7, 0.1, 60 respectively. The results of the clustering method with histograms of

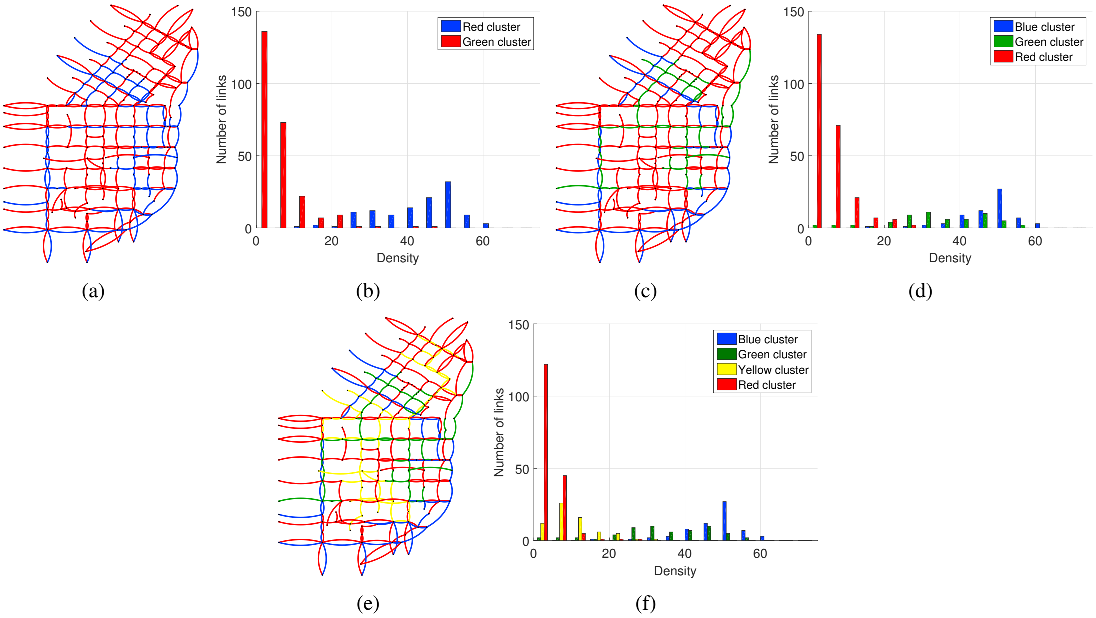


Fig. 5. Clustering results (a, c, e) and histogram of link densities (b, d, f) at time t=70 for 2-4 clusters.

Table 3. Average values (μ), standard deviation (σ) of link densities ([veh/(km.lane))], TV_n, and θ of obtained clusters in San-Francisco network for cases with 2-4 clusters.

Number of clusters	Blue	Green	Yellow	Red	TV_n	$\theta(\times 10^5)$
2	43.84/10.43	-	-	6.22/6.42	0.17	-4.71
3	48.49/7.76	34.60/13.38	-	5.71/5.17	0.16	-5.44
4	48.57/7.79	34.78/13.42	10.70/6.21	3.94/3.65	0.15	-4.15

link (at time t=70) for different number of clusters are depicted next to each other in Fig. 5(a)-5(f). By comparing average density values in different clusters presented in Table 3, we could easily see that the developed method has the ability to differentiate between regions with different density values. It can be observed in Fig 5(c) that blue region is fully congested while the green region slightly gets congested and these two regions are found inside the uncongested red region. Is is also very interesting that by increasing number of clusters to 4, the algorithm is able to break the uncongested part into two parts where one part is almost empty (i.e. red and yellow areas in Fig. 5(e)). Note also that while the value of TV_n is always decreasing with the number of clusters, the value of theta is minimum for 3 clusters, as the similarity terms between various clusters in Eq. (11) highlight over-clustering for 4 clusters. Larger number of clusters have higher theta than -5.44×10^5 and TV_n very close to 0.15 and for this reason, we will focus our analysis for the remaining of the paper for a maximum of 4 clusters.

This static method of clustering produces similar results in terms of the mean and standard deviation of each cluster with the work in (Saeedmanesh and Geroliminis, 2016) (see table 2 in the paper). Nevertheless, as mentioned above, (Saeedmanesh and Geroliminis, 2016) does not impose directly connectivity, which would make the extension to the dynamic case not straightforward. For instance, in (Saeedmanesh and Geroliminis, 2016), clusters have a few disconnected links close to the borders. The developed algorithm in this paper can be utilized for dynamic clustering, as connectivity is enforced in different steps.

4.2. Dynamic clustering results

In this section, the clustering method is applied in a dynamic time framework. As we explained in the methodological part, the proposed algorithm models (replicates) the process of congestion formation using the growth and shrinking of the congested areas where the most heterogeneous links are reassigned while enforcing the connectivity constraints (call this method U for updating). In case aforementioned method creates a clustering, which is far from the ideal one (as described by the density mean-variance relation of Fig. 3(b), a full re-clustering is performed from scratch as described in the static case (R: re-clustering). Re-clustering cannot be performed very frequently due to the computational burden. Nevertheless, as congestion has inherently strong spatial correlations, re-clustering from scratch should not occur often if the updating approach performs well.

The performance of the algorithm is investigated under the scenario where a subset of links (e.g. 20%) is allowed to be assigned to different clusters at each time interval. Basically, given the clusters at time t, we first identify the 80% of links in different clusters that remain homogeneous with respect to measurements at new time $t + \Delta t$. In other words, at least the 80% of the roads inside clusters with low variances will remain in the same cluster. Note that, the formulation described in section 2.2 is utilized to find the unchanged roads. Snakes grow in different clusters and the optimization algorithm finds the part of each snake that has low variance. Finally, by applying the fine-tuning approach, the remaining links are reassigned to the appropriate clusters and minimize θ in Eq. (11). This approach is on the one side consistent with the physics of propagation (it is not logical that too many roads switch from one cluster to other due to spatial correlations) and on the other side eases the computational performance of the algorithm as only a subset of links are considered during optimization. Results for different methods (best static clustering vs. update U vs. update with reclustering if needed, named UR) and values of parameters (frequency of update f and number of clusters N_s) are depicted in Fig. 6.

Figures 6(a)-6(d)-6(g) depict for different number of clusters (two, three and four respectively) the case where clusters are updated every time interval (f=1) in the semi-congested and congested traffic conditions (time index t ranges between t=45 to t=90). The performance of updating (blue curve) is compared with unchanged static clustering (red curve) and time-independent static partitions (green points). Note that there might be no similarity between static clusters at two consecutive time interval (green curve) as the optimization does not assume any time correlation for obtaining the static solutions.

As it is depicted, by changing a few links at each step, we are able to partition the network in homogeneous areas with small variance which is a justification of the fact that congestion propagates with finite speed (see Fig. 6(a), 6(d), and 6(g) for the cases with 2, 3 and 4 clusters and also compare with Fig. 4). Thus, we are able to achieve clustering with low variance and track the congestion propagation by only changing a small percentage of each cluster compared to a previous time interval. Figures 7(a) to 7(j) and 8(a) to 8(j) depict the evolution of two and three clusters over time for the case where the updating has been done frequently (f=1). Different clusters in each of the two cases have different level of congestion as depicted in Fig. 9(a) and 9(c), which show the evolution of space-mean density within each cluster.

Figures 6(b)-6(e)-6(h) represent the same cases where we apply the update algorithm every 5 time intervals (f=5). It can be seen from the blue curves that in the transition period (e.g. from uncongested to congested), updating a few links every 5 cycles cannot perform well as spatial correlations are smaller and propagation is faster. Note that, this approach performs significantly better than keeping clusters completely unchanged (red curve). As it was stated before, by monitoring the average and variance of updated clusters at each time (blue squares in Fig. 6(c)-6(f)-6(i)), we are able to identify when re-partitioning is needed (e.g. at time $t = 65$ for 2-3 clusters and 75 for 4 clusters, the obtained point are deviated a lot from the well-defined curve). Hence, we re-partition the network from scratch (i.e. time correlation is ignored). The black curves in Fig. 6(b)-6(e)-6(h) depict the evolution of clusters after repartitioning which is critical to capture spatiotemporal growth of congestion after time $t = 65$ for 2-3 and $t = 75$ for 4 clusters. The trajectories of the obtained clusters under different scenarios (e.g. (i) nonupdated; (ii) updating (U); (iii) updating+repartitioning (UR)) are shown in Fig. 6(c)-6(f). Note how closer the trajectories are to well-defined curve between average and variance after repartitioning (UR). Trajectories are shown for a subset of representative clusters (mainly the ones with more congestion) to keep the clarity of the figure. The ones that are not shown experience similar trends.

As it was mentioned, it is crucial to know the computational complexity of the fine-tuning step as it has a direct impact on the updating frequency. The average and standard deviation of running time for 2-4 clusters are $\bar{t} = [25, 44, 72], \sigma_t = [7, 13, 35]$ seconds respectively.

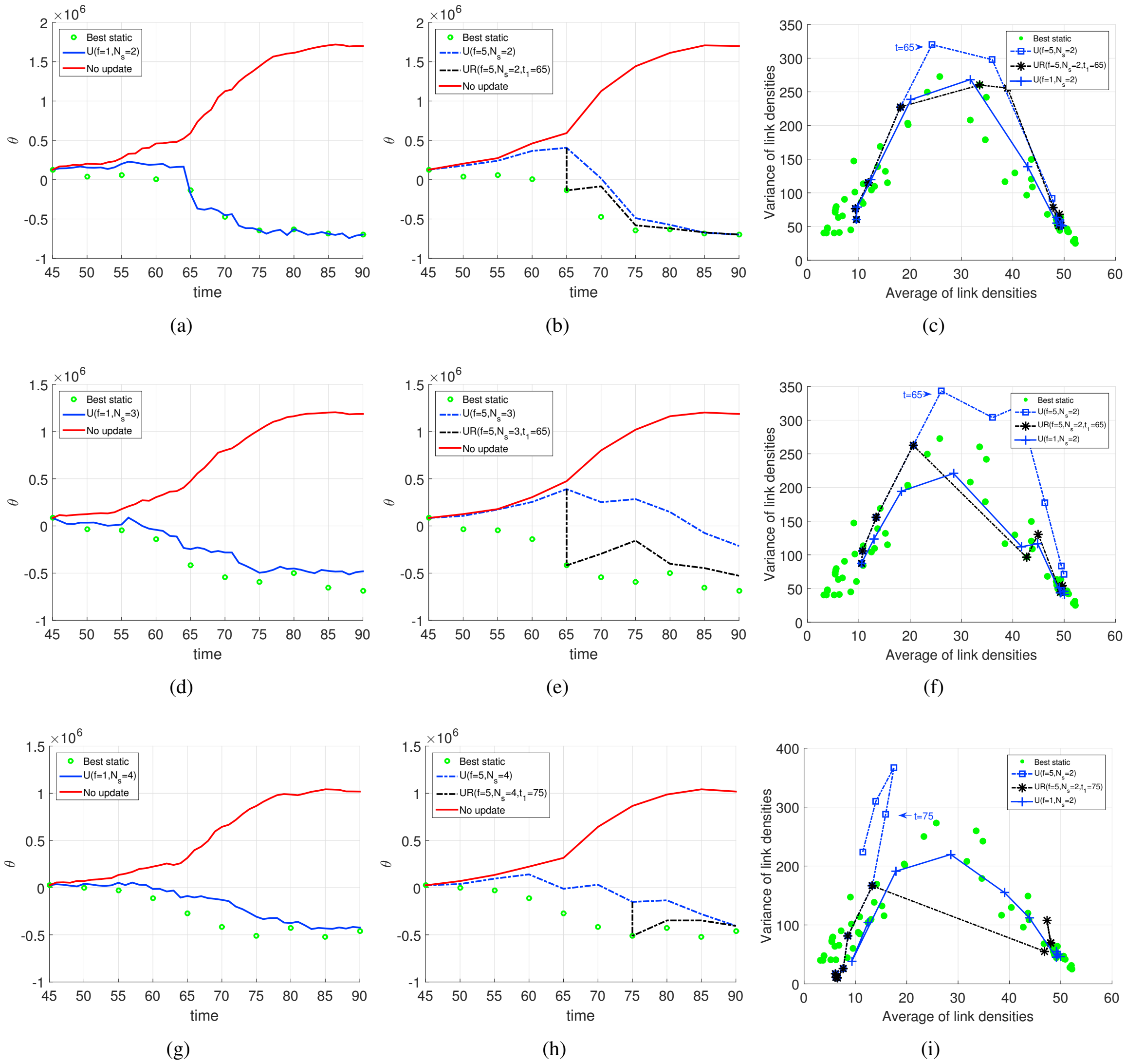


Fig. 6. Clustering results for the dynamic case with 2-4 clusters: Evolution of theta values for static clustering , update (U), update and reclustering and no update for f=1 (Fig. a, d, g), for f=5 (Fig. b, e, h). Fig. c, f, i present the evolution of variance vs. average density trajectories for the various cases.

5. Conclusion

This paper proposes a methodology to dynamically partition heterogeneously distributed congested networks into a small number of homogeneous sub-regions with a guarantee of spatial connectivity and a minimization of a heterogeneity metric. In the proposed approach, we consider heterogeneity as a main objective function and explicitly impose spatial connectivity as a constraint in different steps. To achieve connectivity in the clusters, we limit the feasible set of optimization algorithm to a set of homogeneous connected areas subgraphs, called snakes. The problem is formulated as a Mixed Integer Linear Programming (MILP) with an objective of minimizing heterogeneity (variance) of the obtained clusters. The dynamic clustering is based on an iterative and fast procedure that considers the spatiotemporal characteristics of congestion propagation and identifies the links with the highest degree of heterogeneity due to time dependent conditions and finally re-cluster them to guarantee connectivity and minimize heterogeneity. The proposed algorithm replicates the process of congestion formation by considering growth and shrinking of the congested areas.

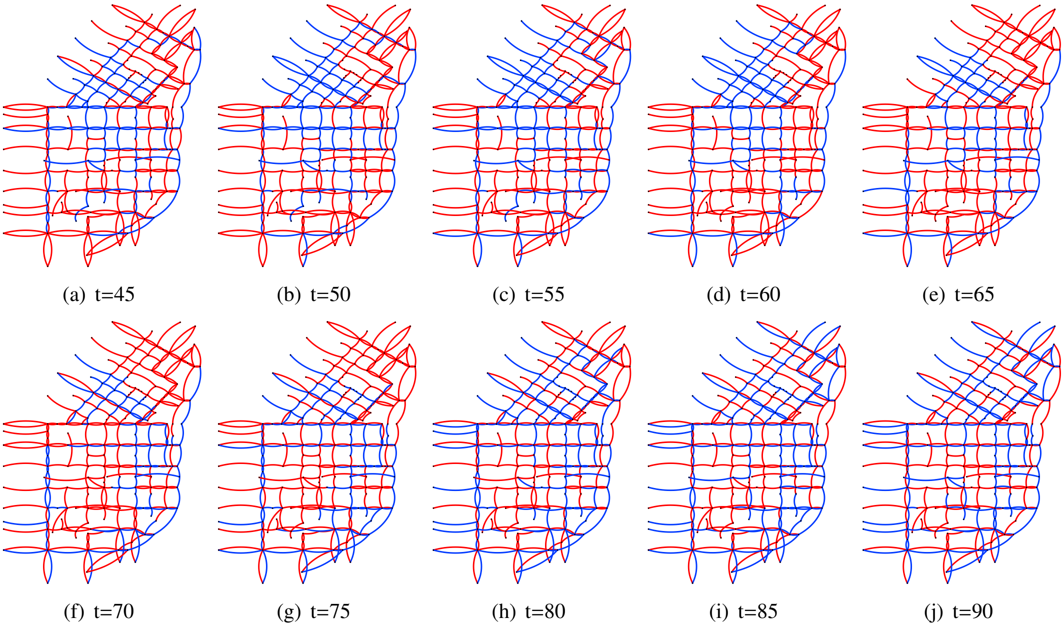


Fig. 7. Evolution of two clusters over time (f=1,N_s=2).

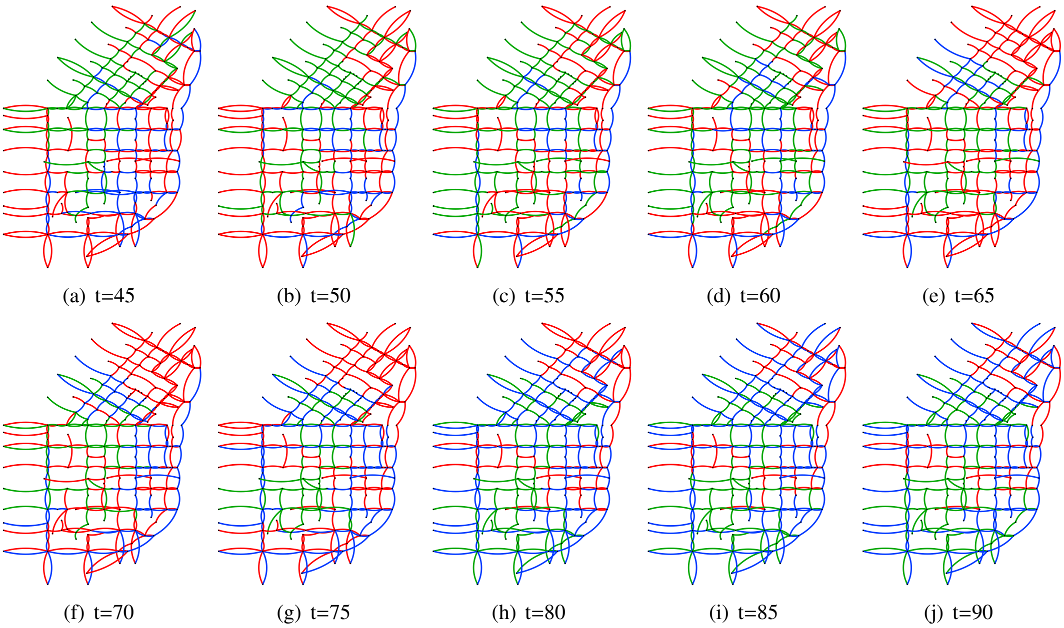


Fig. 8. Evolution of three clusters over time (f=1,N_s=3).

While spatial homogeneity should be an important property of clusters, the shape of the clusters and also their boundaries should facilitate the control objectives. Given that drivers tend to choose routes without many turns, a non-smooth boundary where perimeter control is applied might create shortest paths with a large number of turns and change the behavior of drivers in non-predictable ways. Hence, defining a shape metric to get more compact clusters and make a trade-off between compactness and homogeneity is challenging topic which alleviates the utilization of perimeter control. Indeed, integrating dynamic clustering with perimeter control strategies should be a challenging

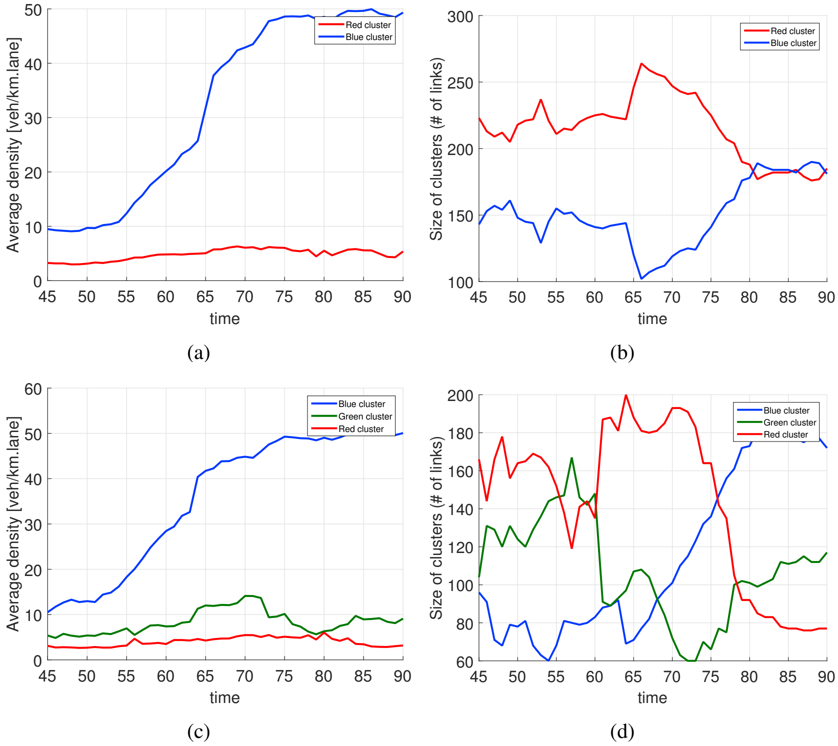


Fig. 9. Evolution of average cluster densities and size over time: (a-b) f=1,N_s=2;(c-d) f=1,N_s=3

research priority. Moreover, developing heuristic approaches is critical to solve the optimization problem for larger networks in a short amount of time.

Acknowledgements

This work has been supported by the (European Research Council) ERC Starting Grant "METAFERW: Modeling and controlling traffic congestion and propagation in large-scale urban multimodal networks".

References

Afsar, M.M., Tayarani-N, M.H., 2014. Clustering in sensor networks: A literature survey. Journal of Network and Computer Applications 46, 198–226.

Anbaroglu, B., Heydecker, B., Cheng, T., 2014. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. Transportation Research Part C: Emerging Technologies 48, 47 – 65.

Baker, D., Ephremides, A., 1981. The architectural organization of a mobile radio network via a distributed algorithm. IEEE Transactions on Communications 29, 1694–1701.

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.

Daganzo, C.F., 2007. Urban gridlock: macroscopic modeling and mitigation approaches. Transportation Research Part B: Methodological 41, 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: Methodological 45, 278–288.

Daganzo, C.F., Geroliminis, N., 2008. An analytical approximation for the macroscopic fundamental diagram of urban traffic. Transportation Research Part B: Methodological 42, 771–781.

Duque, J.C., Ramos, R., Suriach, J., 2007. Supervised regionalization methods: A survey. International Regional Science Review 30, 195–220.

Edie, 1963. Discussion of traffic stream measurements and definitions. Proc. 2nd Int. Symp. On the Theory of Traffic Flow, (J. Almond, editor) , 139–154.

Etemadnia, H., Abdelghany, K., Hassan, A., 2014. A network partitioning methodology for distributed traffic management applications. Transportmetrica A: Transport Science 10, 518–532.
Gayah, V.V., Daganzo, C.F., 2011. Clockwise hysteresis loops in the macroscopic fundamental diagram: an effect of network instability. Transportation Research Part B: Methodological 45, 643–655.
Gerla, M., Tsai, J.T.C., 1995. Multicluster, mobile, multimedia radio network. Wirel. Netw. 1, 255–265.
Geroliminis, N., Daganzo, C.F., 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. Transportation Research Part B: Methodological 42, 759–770.
Geroliminis, N., Sun, J., 2011. Properties of a well-defined macroscopic fundamental diagram for urban traffic. Transportation Research Part B: Methodological 45, 605–617.
Godfrey, J.W., 1969. The mechanism of a road network. Traffic Engineering and Control 11, 323–327.
Haddad, J., Ramezani, M., Geroliminis, N., 2013. Cooperative traffic control of a mixed network with two urban regions and a freeway. Transportation Research Part B: Methodological 54, 17–36.
Han, Y., Moutarde, F., 2013. Statistical traffic state analysis in large-scale transportation networks using locality-preserving non-negative matrix factorisation. Intelligent Transport Systems, IET 7, 283–295.
Hans, E., Chiabaut, N., Leclercq, L., 2014. Clustering approach for assessing the travel time variability of arterials. Transportation Research Record: Journal of the Transportation Research Board , 42–49.
Helbing, D., 2001. Traffic and related self-driven many-particle systems. Reviews of modern physics 73, 1067–1141.
Herman, R., Prigogine, I., 1979. A two-fluid approach to town traffic. Science 204, 148–151.
Ji, Y., Geroliminis, N., 2012. On the spatial partitioning of urban transportation networks. Transportation Research Part B: Methodological 46, 1639–1656.
Ji, Y., Luo, J., Geroliminis, N., 2014. Empirical observations of congestion propagation and dynamic partitioning with probe data for large-scale systems. Transportation Research Record 2422, 1–11.
Keyvan-Ekbatani, M., Yildirimoglu, M., Geroliminis, N., Papageorgiou, M., 2015. Multiple concentric gating traffic control in large-scale urban networks. IEEE Transactions on Intelligent Transportation Systems 16, 2141–2154.
Knoop, V., Hoogendoorn, S., 2013. Empirics of a generalized macroscopic fundamental diagram for urban freeways. Transportation Research Record: Journal of the Transportation Research Board , 133–141.
Lancichinetti, A., Fortunato, S., 2009. Community detection algorithms: A comparative analysis. Phys. Rev. E 80, 056117.
Laval, J.A., Castrillón, F., 2015. Stochastic approximations for the macroscopic fundamental diagram of urban networks. Transportation Research Part B: Methodological 81, 904–916.
Leclercq, L., Geroliminis, N., 2013. Estimating mfds in simple networks with route choice. Transportation Research Part B: Methodological 57, 468–484.
Leclercq, L., Parzani, C., Knoop, V.L., Amourette, J., Hoogendoorn, S.P., 2015. Macroscopic traffic dynamics with heterogeneous route patterns. Transportation Research Part C: Emerging Technologies 59, 292–307.
Mahmassani, H., Williams, J., Herman, R., 1987. Performance of urban traffic networks, in: Transportation and Traffic Theory (Proceedings of the Tenth International on Transportation and Traffic Theory Symposium, Cambridge, Massachusetts), NH Gartner, NHM Wilson, editors, Elsevier.
Mahmassani, H.S., Saberi, M., Zockaie, A., 2013. Urban network gridlock: Theory, characteristics, and dynamics. Transportation Research Part C: Emerging Technologies 36, 480–497.
Ramezani, M., Haddad, J., Geroliminis, N., 2015. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. Transportation Research Part B: Methodological 74, 1–19.
Saberi, M., Mahmassani, H.S., Hou, T., Zockaie, A., 2014. Estimating network fundamental diagram using three-dimensional vehicle trajectories: Extending edies definitions of traffic flow variables to networks. Transportation Research Record: Journal of the Transportation Research Board 2422, 12–20.
Saeedmanesh, M., Geroliminis, N., 2015. Optimization-based clustering of traffic networks using distinct local components. IEEE 18th International Conference on Intelligent Transportation Systems , 2135–2140.
Saeedmanesh, M., Geroliminis, N., 2016. Clustering of heterogeneous networks with directional flows based on snake similarities. Transportation Research Part B: Methodological 91, 250–269.
Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence 22, 888–905.
Shirabe, T., 2005. A model of contiguity for spatial unit allocation. Geographical Analysis 37, 2–16.
Yildirimoglu, M., Geroliminis, N., 2014. Approximating dynamic equilibrium conditions with macroscopic fundamental diagrams. Transportation Research Part B: Methodological 70, 186–200.