Aggregation and travel time calculation over large scale traffic networks: An empiric study on the Grenoble City

Giacomo Casadei, Vadim Bertrand, Baptiste Gouin, Carlos Canudas de Wit

Aggregation and Travel Time Calculation over Large Scale Traffic Networks: an Empiric Study on the Grenoble City

G. Casadei, V. Bertrand, B. Gouin, C. Canudas-de-Wit

G. Casadei, V. Bertrand and C. Canudas-de-Wit are with Univ. Grenoble Alpes, CNRS, INRIA, GIPSA-lab, F-38000 Grenoble, France (giacomo.casadei@gipsa-lab.fr, vadim.bertrand@gipsa-lab.fr, carlos.canudas-de-wit@gipsa-lab.fr). [1]

Abstract

In this paper we consider aggregation technique to reduce the complexity of large-scale traffic network. In particular, we consider the city of Grenoble and show that, by *clustering* adjacent sections based on a similarity of speed condition, it is possible to cut down the complexity of the network without loosing crucial and intrinsic information. To this end, we consider travel time computation as a metric of comparison between the original graph and the reduced one: for each cluster we define four attributes (*average speed*, *primary* and *secondary length* and *heading*) and show that, in case of an aggregation rate of 95%, these attributes are sufficient in order to maintain the travel time error below the 25%.

Keywords: large scale traffic networks, travel time estimation, clustering and aggregation

1. Introduction

Large-scale traffic networks are of great interest nowadays due to the impact traffic has in our everyday life, both economically and health-wise. City management are interested in understanding the evolution of traffic and its patterns over the city in order to take decisions to design improved and more functional infrastructure. However, monitoring the current state of a large scale traffic network is a demanding task. The heterogeneity of available measures poses several question on how to merge different sources of information coming from private and public sources. Furthermore, sparsity is an intrinsic issues related to large scale systems: independently from the source we choose to rely on, we cannot expect the measurements to be sufficiently dense to cover the full network in detail. In the past, the main focus of the research on traffic systems has been the analysis of individual roads or small regions (R. Herring et al. (2010); A. Krause et al. (2008); H. Liu and W. Ma (2009); C. de Fabritiis et al. (2008), but less attention has been given to approaches that are capable of giving a global perspective of the state of a large-scale networks (Y. Han and F. Moutarde (2012); P.S. Castro et al. (2012); H. Wang et al. (2015)). For large scale urban networks, managing real-time traffic information from thousands of links simultaneously is an overwhelming task and extracting important and meaningful insights from this tangle of data can be even more challenging. However, the study of theoretical models capable of capturing the macroscopic behavior of traffic have been introduced many years ago (N. Geroliminis and C. F. Daganzo (2007, 2012); H. Bar-Gera and S. Ahn (2010); C. Buisson and C. Ladier (2009)) and nowadays continues to be a popular and deeply studied research topic (Z. Zhang et al. (2015); L. Leclercq and N. Geroliminis (2013)). In recent years more and more data are becoming available from new sources, such as smart phones, GPS navigators, and their technological penetration allows to have an impressive amount of real-time traffic information, as demonstrated by the Mobile Millennium Project (A. Bayen et al. (2011)).

The main advantage of these new sources is that they do not require placing physical sensors over the network, thus reducing costs related to installation and maintenance: in other words, each user becomes

[1] The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement N° 694209).

a moving sensor inside the network (see C. Peng et al. (2012), J.A. Deri et al. (2016), B. Donovan and D.B. Work (2015)). However many data do not necessarily imply meaningful information, since their interpretation, analysis, treatment and display becomes more and more challenging. From a city management point of view, the availability of huge numbers of GPS traces can be used in a aggregated fashion to determine macro-movements and origin/destination patterns of interest: these information become of great importance in order to monitor and (if possible) adapt the traffic infrastructure to the traffic evolution. To this end, one could consider the possibility to simplify the complexity of the problem by performing an aggregation over the traffic network. A recent research C. Lopez et al. (2017, 2018), performed on the city of Amsterdam, showed that thanks to network clustering, the traffic pattern of the entire city can be synthesized in only four features composed by only nine clusters. On one hand one could consider a static partition, based on geographic structure of the network (see H. Etemadnia et al. (2014); Y. Li and N. Geroliminis (2012); K. An et al. (2018)): on the other hand, to capture the dynamic evolution of the network, the natural approach is to apply the clusterization in real-time (see M. Saeedmanesh and N. Geroliminis (2017)). A static partition allows to consider and describe the evolution of precise areas of interest, for instance crucial origin/destination zones inside the network. With a dynamic partition a certain *point of interest* might belong to different cluster in different moments of the day: thus the interest shifts on the evolution of the clusters and their impact/meaning in the network status. Depending on the problem at hand, the two approaches could lead, in terms of performance, to completely different results: if a traffic network presents a regular evolution which repeats daily/weakly following a distinctive pattern, a static analysis/partition of the traffic network will guarantee reliable results. However, if a traffic network has a highly variable evolution, which might depend on different parameters (such as weather, traffic accidents etc etc), a dynamic aggregation allows to be reactive to this variability and to follow the particular evolution in real time.

From a general perspective, the goal of the clustering is to reduce the complexity of an original graph while respecting a certain predefined criterion for the nodes state/edge value (S. Fortunato (2010); F. D. Malliarosa and M. Vazirgiannisa (2013)). It is common knowledge that there is no single best strategy for graph clustering, which justifies the variety of existing approaches. Moreover, most quality indixes for graph clustering have turned out to be NP-hard to optimize and rather resilient to effective approximations (see G. Ausiello et al. (2002)). The research for clusters can be a demanding task from a computational point of view and this constraint becomes more and more evident as the size of the network increases. Solutions have been proposed in the literature in order to deal with networks of big size (see for instance D. Delling et al. (2009)) but they often depend on the particular metric used to evaluate the clustering procedure and in general they are still computationally heavy. Some popular metrics often chosen are coverage (namely the fraction of intra-cluster edges, to be minimized) U. Brandes et al. (2005), performance (namely the fraction of correctly placed vertex pairs, to be maximized) S. M. van Dongen (2000), inter-cluster conductance (namely the thickest bottleneck created by the definition of clusters) R. Kannan et al. (2004) and modularity (namely the coverage ratio between a clustering and the same with randomly rearranged edges, to be minimized) M. E. J. Newman and M. E. J. Newman and M. Girvan (2004). For all these metrics however, the optimization problem is NP-hard and thus not feasible in a real-time implementation. Thus, if the aggregation has to be performed in a time window compatible with a real time application, simpler criteria and procedure needs to be defined.

1.1. Objective of this paper and Structure

In this paper we investigate and propose an aggregation based analysis of large traffic networks in view of evaluating travel times between areas of interest inside the urban network of Grenoble. In particular, by aggregating roads which belongs to similar categories (highways, national roads, urban roads) and have *similar* speed (up to a certain cut-off), we will show that it is possible to reduce drastically the complexity of the network while introducing an acceptable error in the travel time calculation. The information and results we obtain are of great interest to understand the macroscopic evolution of a large-scale traffic network and to evaluate the average time that users spend in transiting between different areas along the day. In particular we will focus on the duality of our approach: on one hand aggregation allows to simplify the complexity of representation and calculation over the network. On the other hand, reduction in the granularity affects the precision in evaluating the state of the network.

The error introduced will be calculated for multi-origin multi-destination path, namely given an aggregation, we will show that the travel time calculated between two clusters is consistent with the average of all the possible path between these two clusters in the original graph. We will show that this error grows (non-linearly) with the aggregation rate and thus a suitable compromise between the reduction of complexity and the error introduced must be found. Furthermore, we show that in the aggregated graph, typical congestion patterns (over a daily and weekly base) can be easily reconstructed and defined. The latter leaves open the possibility to define travel time estimation based on historic data by matching the present condition with a set of typical pre-defined patterns.

The interest for dynamic aggregations is particularly motivated by traffic networks which show erratic and *fragile* behaviors which frequently diverge from a nominal behavior. In our case study, an analysis of the data available showed that the definition of recurrent traffic patterns in crucial parts of the city is a very difficult task. As it will be explained in the following section, the structure of the Grenoble network is not redundant and, due to the presence of mountain ranges around city, it lacks for highway rings. These limitations translates into a highly fragile network which easily (and ultimately unpredictably) leads to high congestion risks.

The paper is structured as follows. In Section 2, we introduce the basic and practical information about data and map we use in order to describe the Grenoble traffic network. In Section 3, we describe in details the aggregation technique we developed and the attributes which describe the clusters: these parameters play a fundamental role in the calculation of travel times, since they allow to reduce the complexity while preserving some of the physical properties of the real network. Then, in Section 4 we show results and comparisons between travel times calculated over the original detailed graph and the aggregated graph: in particular we will consider two aggregation rates, one that keeps the *travel time error* small, the other which simplifies drastically the complexity of the graph.

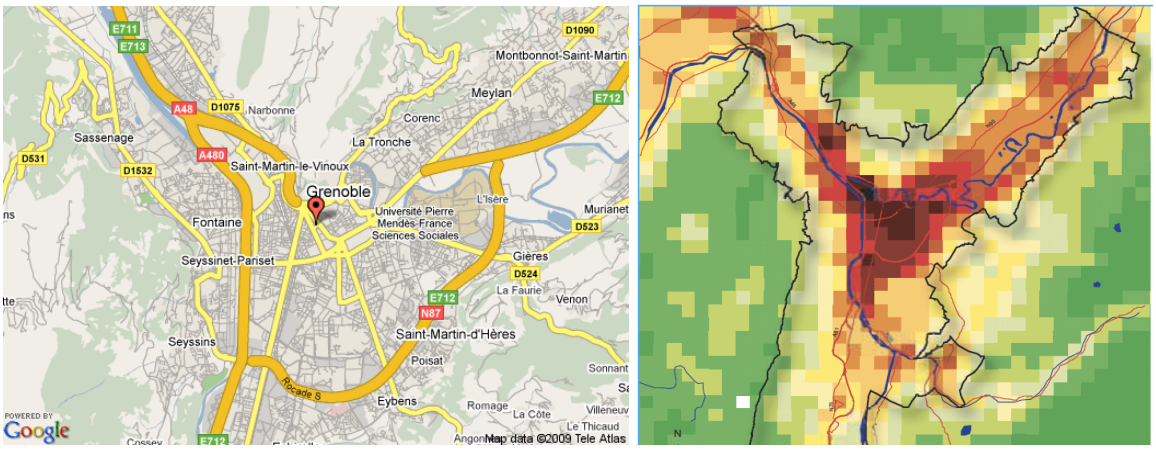

Figure 1: Map of the city of Grenoble and main arteries: in orange the highways entering/exiting the city, in yellow the most important urban roads. On the right, the daily pollution concentration over the Grenoble area.

1.2. Problem setting

Situated in the south-east of France close to the Alps, Grenoble is the 16th largest city in France with a population of the agglomeration of 451.000 people. The geometry of the city itself is strictly related to the geography of the surrounding mountains: the Chartreuse range on the north, the Vercors range on the west and the Belledonne range on the east determines the alcove in which Grenoble lies. Due to these physical constraints, the urban network of the city can be seen as star-connection, where the urban network as the center and the three arms as the highways from/to Lyon (north-west), Chambery (north-east) and Gap (south).

As shown in Figure 1, these three fundamental entry/exit points are connected to each other only by the south-ring N-87 and the urban network. This critical and fragile connection is one of the main reason of the sensitivity of the network, which places the city as the seventh most congested in France with an average of more than 45 hours per year spent in traffic (source TomTom). Due to its strategic importance, the N-87 has been the target of the project Grenoble Traffic Lab (GTL), started back in 2009 and became operative in 2013, an innovative online platform which collects data coming from a sensor network (130 magnetometers positioned over a stretch of 10 km) and allows to have a detailed and granular monitoring of the traffic condition of one-direction of the N-87 (see C. Canudas de Wit et al. (2014)). Being GTL a successful experience, in the context of the ERC-Project ScaleFree-Back (C. Canudas de Wit (2015)), the aim is to extend a similar approach to the full scale of the city of Grenoble. There are several reasons of interest in monitoring the status of the traffic network on a city-scale. From the traffic management point of view, it allows to understand the dynamics of traffic, the critical points of the networks, the impact of special events over the normal behavior of the network. At the same time, the status and evolution of the city traffic allow to plan modification in the network infrastructure with the aim to reduce the hours spent by users in traffic jam and incidentally, reduce the pollution. With respect to this last aspect, the city of Grenoble presents again an unfortunate morphology due to the surrounding mountains which limit the air circulation. In the last decade, the city has recorded at least 20 days of pollution peak per year (days in which the pollution level overpass the limits imposed by the EU).

For economical but also practical reasons, it would not be possible to place a dense network of sensor such as the one on the N-87 over a city network. However, as already mentioned, there are nowadays different sources of information which we can exploit to monitor and reconstruct the state of the network. We think in particular to GPS traces from mobiles and cars and the information we are able to retrieve from them. It comes without saying that the number of data available is potentially immense. This is not necessarily a positive aspect since on one hand the availability of information implies robustness but at the same time requires a thorough analysis and manipulation to be exploited and interpreted.

In order to obtain a sufficiently dense and reliable data set, we rely on speed measurements obtained by TomTom. They indeed provide a really granular information with more than twenty thousand sections spanning the urban network. This implies that computation and analysis of these information can be really complex and costly if not opportunely set. Furthermore, despite covering the main arteries of the network, there are several areas where no data is available. Thus it could be of interest, when possible, to infer something about these areas even though a direct measurement is not available. More in general, this paper and the approach presented here are motivated by the possibility to reduce the need of detailed information about a large scale urban network while keeping the error of approximation as small as possible.

2. Architecture of the platform

To monitor large-scale traffic networks, many ingredients have to come into place and different part of the architecture need to be thoroughly defined. As mentioned before, we acquire speed data from TomTom and thus the first step is to retrieve them and make them available for computation. To this end, beside the data itself, the map-graph plays a fundamental role both in displaying and allowing computation over the traffic network. In this section we present these two main features of the architecture: first we introduce data specifications and details, then we explain how the map is obtained and the conversion/manipulation are performed in order to represent TomTom data over a graph structure.

2.1. Data specification and acquisition

We acquire data through a HTTP GET request sent from a white-listed IP using a secure API key. In response we receive a Protocol Buffers - .proto - file. Protocol buffer serializes structured data, as XML does, while producing smaller file and achieving faster reading/writing operations. This allows to transfer and archive a high volume of data in a faster and easier way. A raw data file, in our case a file around 67 Kb, can be seen as a time stamp and a list of samples. Each sample is made of four fields:

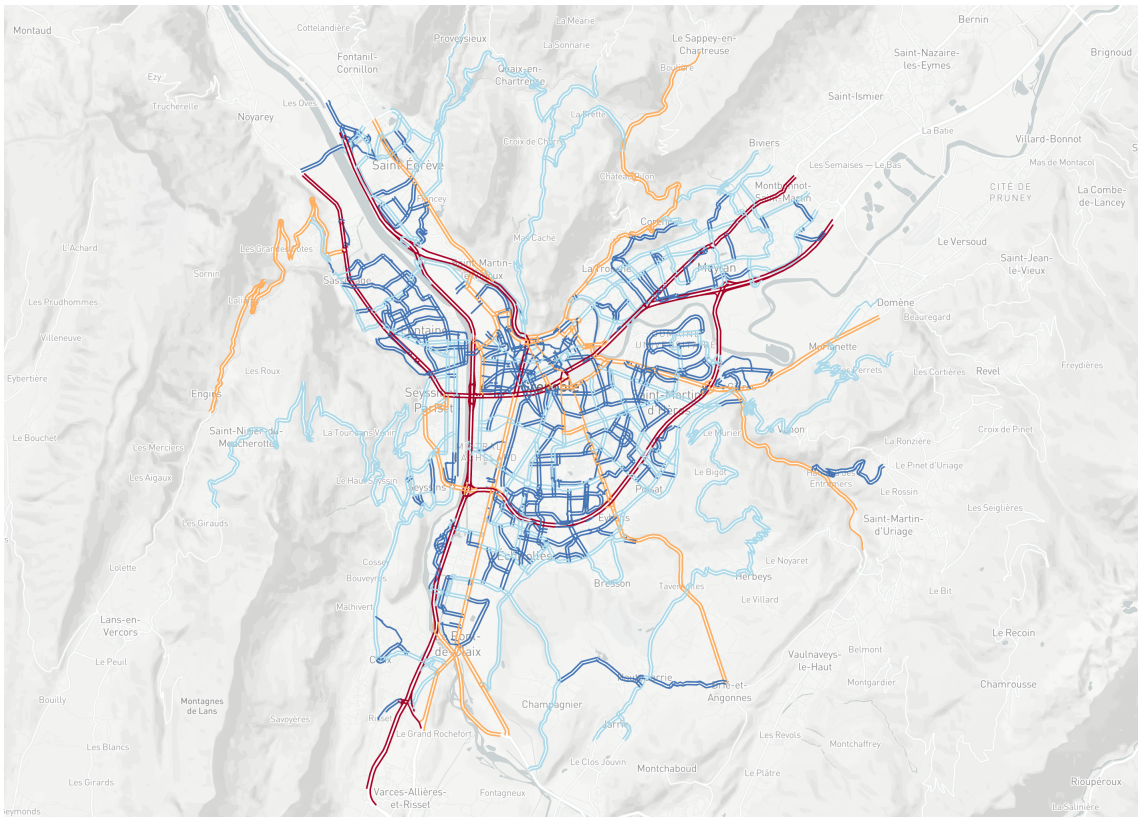

Figure 2: Urban network with TomTom traffic data: the different colors represent the different category of roads. In particular, we identify four road categories FRC defined by TomTom: red corresponds to FRC0 roads, orange to FRC1, blue to FRC2 and light-blue to FRC3-FRC4.

- **Location of Interest:** a road stretch and its direction, whose location is encoded using OpenLR representation, a standard for encoding, transmitting and decoding geographical locations developed by TomTom since 2009
- **Road Category:** an attribute of importance of the road on a range from FRC0 to FRC7 where FRC0 stands for the larger roads and FRC7 for the smaller ones. Typically the can be classified as:
  - FRC0-FRC1 represents highways and freeways
  - FRC02 represents the most important urban and extra-urban arteries
  - FRC03-FRC04 represents the smaller and less important roads in the traffic network
  - FRC05-FRC07 represents even less important roads, but information on these class are not available for the Grenoble network
- **Travel Time:** current travel time in seconds along the location of interest, computed using speed-data measured on the road and aggregated over a time window of 1 up to 2 minutes (the time required from TomTom to acquire and process raw data)
- **Average Speed:** current average speed in km/h, based on the current travel time and the length of the stretch

- **Calculated Data Quality:** confidence value from 0% to 100%, that depends on large number of factors, between which: the freshness of the measurements, the volume of the measurements and the deviation between the measurements and the historical records.

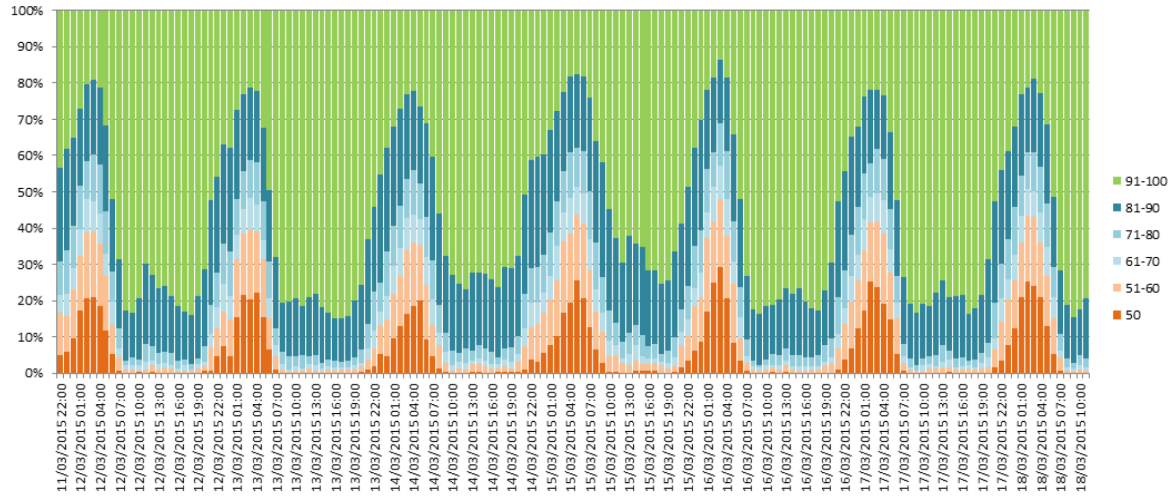

Figure 3: Schematic of the confidence value classification along time for FRC0-FRC2 roads in a typical urban network area.

In France, TomTom guarantees a confidence value greater than 50% which as expected (see Figure 3, source TomTom) is much larger during the day, when more vehicles can provide measurements while it decreases drastically during the night. In particular, during the day, 80% of the data have a confidence value of more than 90% which in other words means that during the crucial moments of the day, TomTom provides highly reliable information. In the Grenoble area, travel time and speed data are available for roads between FRC0 and FRC4. They cover around 550 km which represents more than 50% of the roads in the FRC0-FRC4 class and in a general perspective 20% of the total road network. To validate the correctness of their aggregated measurements, TomTom has performed *a posteriori* experiments to quantify its data reliability using TISA QBench Calculations methodology and for FRC0 and FRC1 roads TomTom claims an excellent score above 80%.

With respect to data of the FRC0 category, thanks to the GTL project we could further validate the confidence parameter given by TomTom: in particular we compared TomTom profile of velocities with the one we obtain from our ground-sensors over the whole length of the N-87. The results are reported in Figure 4 and clearly confirms that TomTom is highly reliable in high category roads. The small positive bias and the slight delay (in the order of 1 minute) are totally compatible with the case of study: indeed GTL frequency of update (15 sec) is much smaller than the one of TomTom and, thanks to the 130 physical sensors in place, the granularity is much higher. Nevertheless, TomTom recovers perfectly the traffic trends and evolution and on a larger scale confirms the possibility to use aggregated information.

2.2. Map and and its conversion

In order to accurately represents Grenoble road network we also need a detailed road map. With this respect, TomTom provides two Shapefiles, one relative to the sections and one to the intersections of the whole urban network. A Shapefile is a geospatial data format that allows to describe vector features such as points, lines or polygons. Each feature can hold attributes used to describe it. In our file the roads are represented by lines with their extremities as intersections. Roads have several attributes such as length, number of lanes, maximum allowed speed, road name and many more. From this Shapefile we construct a first graph where intersections are nodes and stretch of roads are edges (about 49000 nodes and 98000 edges,

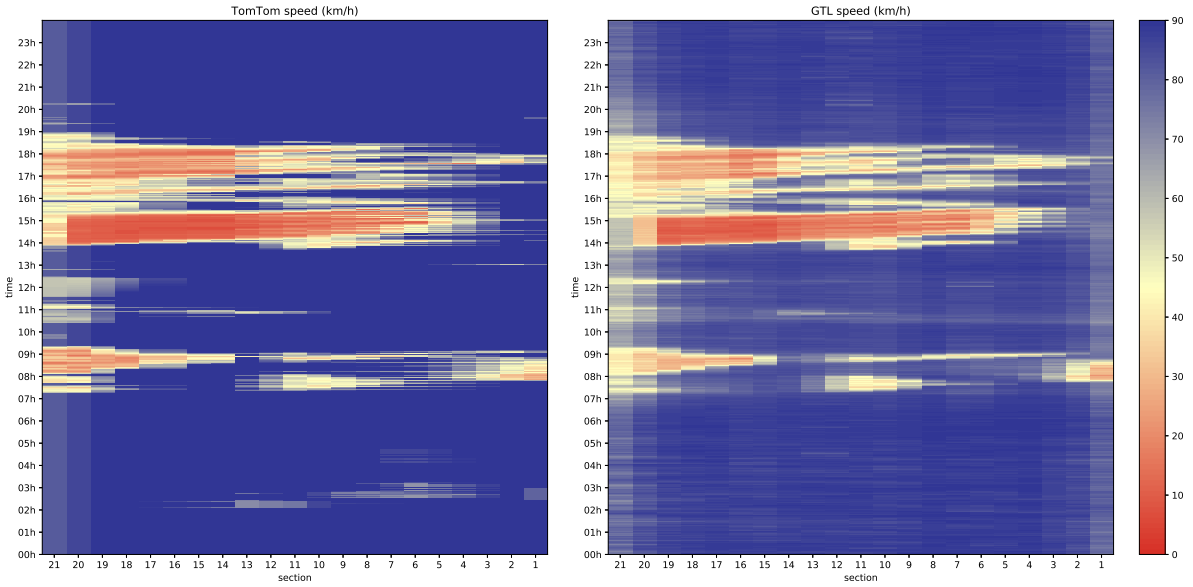

Figure 4: Daily difference between ground truth travel time and TomTom reported travel time.

Figure 5). This graph is directed therefore between two nodes each direction is represented by an individual edge.

In this framework, the speed measurements will be assigned as weight of the edges. To perform this assignment it is necessary to match the locations where TomTom is providing data to the edges of the graph representing the road network. The representation of the road network we just described is easy to understand, graphically convenient and also natural to define paths. On the other hand, in view of performing an aggregation based on speed measurements, it would be much easier to consider a graph where the attribute of speed is referred to the nodes rather than to the edges. To this end, in the following we consider the line graph obtained from the canonical graph, namely the graph where stretches of roads become nodes and intersections become edges. The procedure to transform a road traffic network to its associated line graph is summarized in Figure 6 for the simple case of one intersection (Figure 6.A).

The classic representation (Figure 6.B) is composed by a node (red) with two directed and two bidirectional edges (6 edges), going to four other intersections (blue). The conversion to the line graph produces a new graph with 6 nodes and 11 edges (Figure 6.C). In terms of representation, the latter is clearly less compact but much easier to manipulate and to use to perform aggregation. More specifically given a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{M}\}$ with n nodes, m edges and vertex i out-degrees d_i, the associated line graph $\mathcal{G}_L = \{\mathcal{V}, \mathcal{E}\}$ contains $v = m$ nodes and e edges with

$$e = \frac{1}{2} \sum_{i=1}^{n} d_i^2 - m \tag{1}$$

In general, the passage to the line graph considers each section as a node and each intersection as a set of edges: this set of edges is directed and has to respect the physical interconnection between sections.

3. Clustering and aggregation technique

Once the line graph of the urban network is obtained, it is now possible to consider aggregation and clustering techniques to reduce the network complexity. In this section, we first explain the aggregation

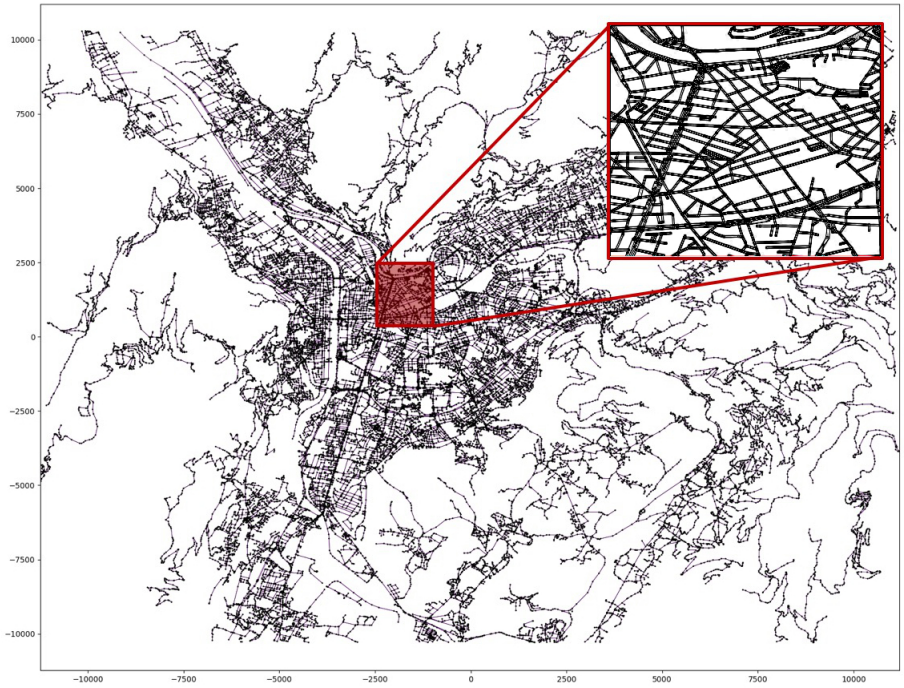

Figure 5: Plot of Grenoble area's road network representation as a graph made of about 49000 nodes and 98000 edges.

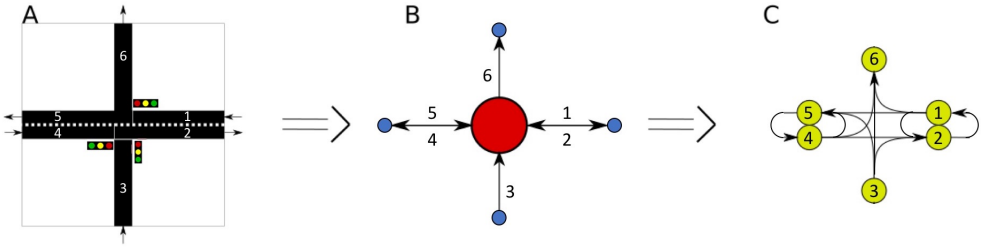

Figure 6: From a road intersection to its representation in a line graph.

principle and how the clustering is performed. Then we introduce in details which attributes we assign to each cluster and how to calculate them.

3.1. Clustering methodology

The principle of clustering strategy we introduce aims to achieve uniqueness in the clustering and at the same time a simple realization since ultimately the goal of this application is to be feasible in a real-time implementation. In words, two nodes belongs to the same cluster if a path exist between them such that every edge of that path satisfies a given criterion. We implemented this clustering by removing from the graph edges that do not satisfy the criterion. Then we compute weakly connected components of the graph, and those weakly components will be our different clusters. We recall the definition of a weakly connected component.

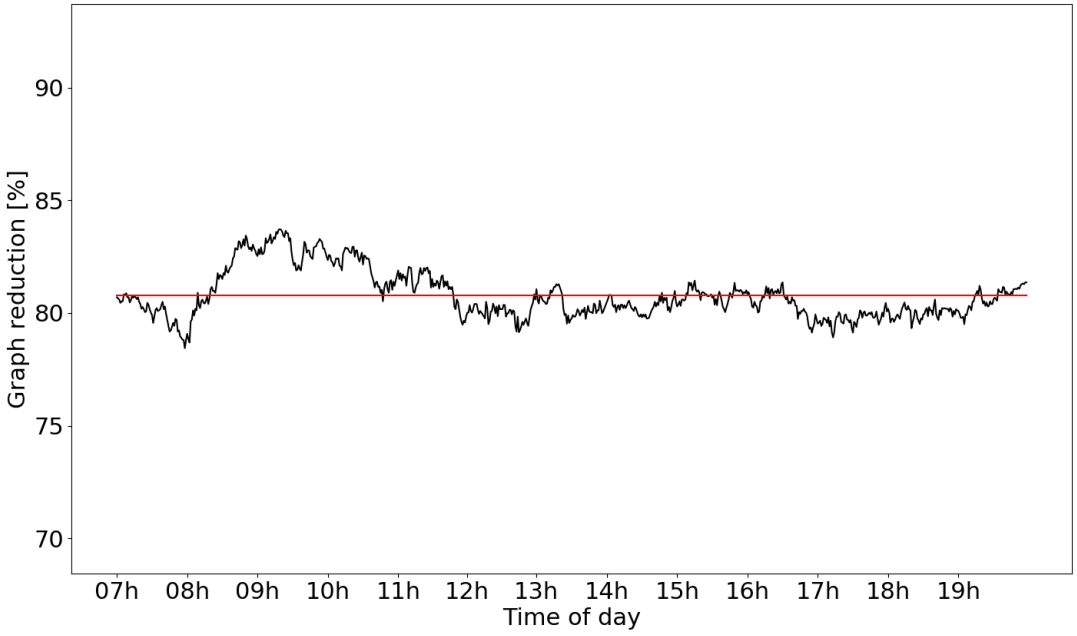

Figure 7: Evolution of the aggregation rate AR during the day: the value is in the average really close to 80% and slightly varies during the day.

		Cut-Off for FRC0-FRC2					
		1	**2**	**4**	**6**	**8**	**10**
Cut-Off for FRC3-FRC4	**1**	70%	73%	75%	77%	78%	79%
	2	75%	78%	81%	82%	83%	84%
	3	80%	82%	84%	86%	87%	88%
	4	82%	85%	87%	88%	89%	90%
	5	85%	87%	89%	90%	91%	92%
	6	87%	89%	91%	92%	93%	95%

Table 1: aggregation rate in percentage for different selection of the cut-off parameters: by columns the cut-off chosen for roads with category FRC0-FRC2, by rows the cut-off chosen for roads with category FRC3-FRC4, both expressed in km/h.

Definition 1. *A weakly connected component of a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{M}\}$ is a maximal subgraph $\tilde{\mathcal{G}} = \{\tilde{\mathcal{N}}, \tilde{\mathcal{M}}\}$ that is connected and such that there is no edge with a tail outside $\tilde{\mathcal{N}}$ and the head in $\tilde{\mathcal{N}}$.*

To perform the clustering over Grenoble traffic network represented by a line graph $\mathcal{G}_L = \{\mathcal{V}, \mathcal{E}\}$, we choose a criterion based on the speed attribute associated to each section, namely each node. For every pair of adjacent sections i-j, we define the weight of the associated edge e_{ij} in the line-graph as

$$e_{ij} = |v_i - v_j| \tag{2}$$

where v_i is the speed of node i. Then we say that an edge satisfies our criterion if the speeds difference of the two ends node is less than a given cutoff $\bar{e}$: formally, we define a new graph $\mathcal{G}_A = \{\mathcal{V}, \mathcal{A}\}$ (where we delete edges that do not satisfy the condition) as

$$a_{ij} = \begin{cases} e_{ij} \leq \bar{e} \Rightarrow a_{ij} = 1 \\ e_{ij} > \bar{e} \Rightarrow a_{ij} = 0 \end{cases} \tag{3}$$

In order to preserve some of the intrinsic structure of the urban network, we have divided the roads in three classes based on their functional road class:

- motorways and freeways (FRC0 and FRC1)
- important urban roads and national roads (FRC2)
- secondary roads (FRC3 to FRC4)

Based on these three groups, sections belonging to different classes can not belong to the same cluster. Thus the cut-off is defined as a vector with a specific entry associated to each of these groups. As mentioned before we perform the aggregation by condensing the weakly connected components of the graph into super-nodes and thus obtaining a new graph $\bar{\mathcal{G}}_L = \{\bar{\mathcal{N}}, \bar{\mathcal{A}}\}$ where $\bar{\mathcal{N}}$ is the number of nodes after the condensation and $\bar{\mathcal{A}}$ is the set of edges connecting the super-nodes. We can now formally define the aggregation rate as

$$AR = \frac{\mathcal{N} - \bar{\mathcal{N}}}{\mathcal{N}} \tag{4}$$

which measures the percentage of complexity reduction introduced by the clustering. The resulting graph $\bar{\mathcal{G}}_L$ thus defines zones that are homogeneous in speed and importance. Clearly, the aggregation rate is a function of the present state of the network, namely the speed measurements we retrieve: for the sake of simplicity, in the rest of the paper we refer to the aggregation rate as the average of AR along the day. For a fixed cut-offs combination, we observed that the average aggregation rate is constant between different days while the variance might slightly change. To clarify this point, in Figure 7, we plot the evolution of AR with cut-off $\bar{e}_{[0;2]} = 2$ and $\bar{e}_{[3;4]} = 4$. As it can be observed, the average value is really close to 81% (see Table 1) and varies between a max value 83% and a min value 78%.

In Table 1, we summarize some of the results obtained by setting different cut-off $\bar{e}_{[0;1]}, \bar{e}_{[2]}$ and $\bar{e}_{[3;4]}$ for the three different classes of roads. In particular we show how the aggregation rate evolves in function of how the cut-off are set. For a desired reduction of complexity, one can choose the opportune range of cut-offs.

An interesting feature of the clustering procedure just defined is that $\bar{\mathcal{G}}_L$ exhibits a tendency to scale-free property. In fact, it is possible to observe that our clustering procedure based on speed similarity and weakly connected components naturally creates hubs (that collects section homogeneous in importance and speed) which represents the crucial areas inside the traffic network. We recall that a scale-free network is a network whose degree distribution can be expressed, at least asymptotically, as a power low

$$P(k) \backsim k^{-\gamma}$$

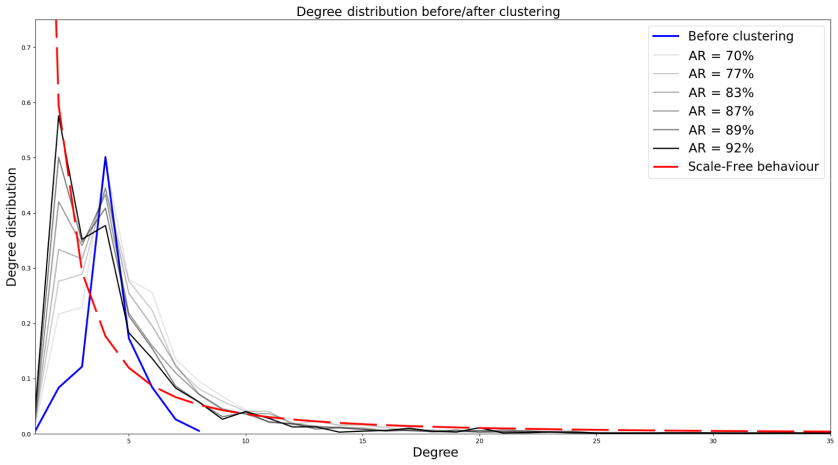

Figure 8: Degree distribution before and after the clustering: for an increasing AR, the degree distribution approaches the scale free behavior up to a $AR = 90\%$ (grey to black plots).

where $P(k)$ is the fraction of nodes in the network having k connections to other nodes and $2 < \gamma < 3$. As shown in Figure 8, for increasing cut-of (and as a consequence aggregation rate AR) the scale-free distribution (in dash-dot red) is approximated better, up to a value around $AR = 90\%$: the different degree distribution move from light a grey to a black line. As a matter of fact, in the degree distribution of the aggregated graph it is always possible to identify characteristic peaks, one around $d = 4$ and one around $d = 2$: in particular it is possible to see that by aggregation, we reduce the number of nodes having degree $d = 4$ and thanks to the hub-structure obtain a degree distribution which is centered around $d = 2$.

3.2. Clusters information

By clustering we obtain a reduction of complexity in terms of number of nodes. However, we still have the possibility to define some cluster properties in such a way that the information of the original physical graph are not all lost. The goal of defining these properties is to capture the essential property of the physical nodes that belongs to a certain cluster and to use them to characterize the cluster. For instance, once a cluster is defined, what is the state (in our case, speed) associated to this super-node? How can it be defined from the original nodes that belongs to this super-node?

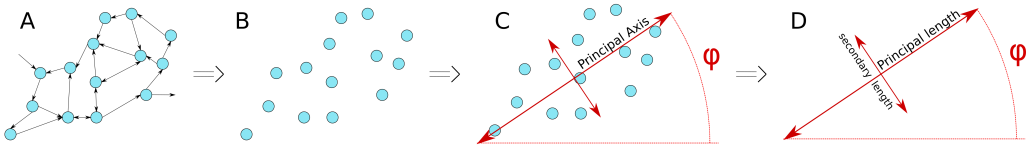

Figure 9: From a set of nodes to the cluster geometric representation.

The reason of using clusters and to introduce a set of properties is to forget the detailed physical information of the original graph without loosing too much of its intrinsic properties. In our case, each node has a huge number of properties between which we recall location in GPS coordinate, associated length, associated speed, information about neighbors section and many many more. By clustering we mainly loose the details of the topology of the graph (namely detailed path from a node to another), on the exact speed

in a certain section (since each section now belongs to a cluster). As a matter of fact, for each cluster we define four properties:

- **the mean speed of a cluster:** is the sum of the length of every section belonging to this cluster divided by the sum of those sections' travel time
- **the center of gravity of a cluster:** depends on the center of mass of each section belonging to this cluster. Each section is considered as unitary mass point and thus the center of gravity coincides with the geometric one.
- **the principal direction of a cluster:** is the angle between the horizontal axis and the direction defined by the primary axis. The latter is provided by the linear regression of this cluster's nodes coordinates (see Figure 9)
- **the lengths of a cluster:** is a vector of two distances that defines the size of this cluster along it's principal direction and the perpendicular direction. They are obtained by computing the variance of this cluster's nodes coordinates along those two directions.

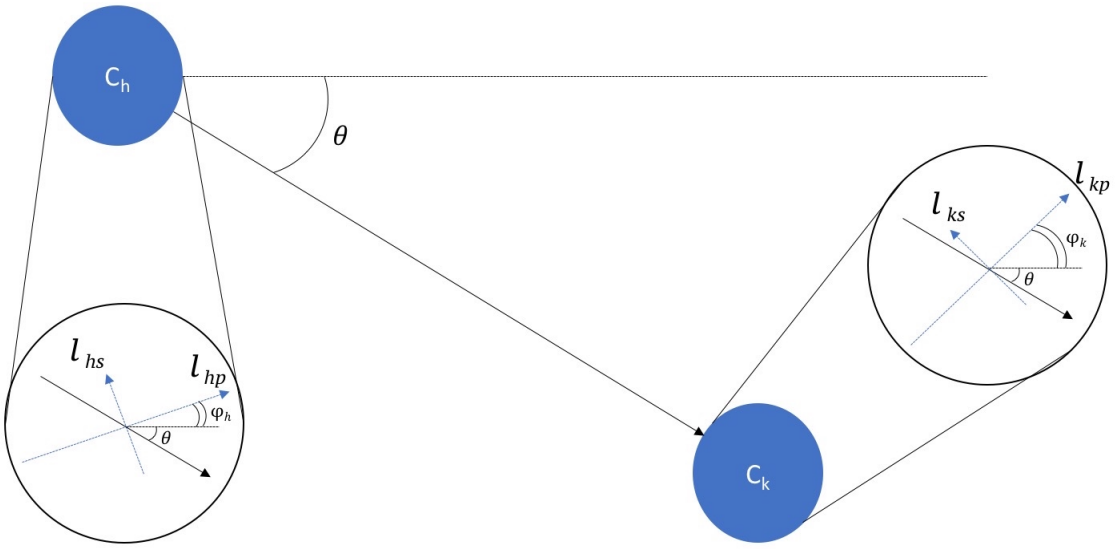

Figure 10: Geometric distance between two clusters, $\mathcal{C}_h$, $\mathcal{C}_k$ the angle between the center of masses is θ.

3.3. Calculation of travel time in the clustered graph

The definition of the last two properties is of great importance in order to simplify drastically the complexity of the network while keeping the approximation error small. First of all, let us introduce a metric to compare the original graph with the clusterized graph. Given two adjacent clusters, $\mathcal{C}_h$, $\mathcal{C}_k$ we define their geometric distance $\ell^{h\leftrightarrow k}$ as the distance between their two center of gravity (see Figure 10), namely

$$\ell^{h\leftrightarrow k} = \frac{\ell^{h\to k}}{2} + \frac{\ell^{k\to h}}{2} \tag{5}$$

with

$$\ell^{h\leftarrow k} = \sqrt{\cos(\theta-\varphi_h)\ell_{hp}^2 + \sin(\theta-\varphi_h)\ell_{hs}^2} \tag{6}$$

$$\ell^{k\leftarrow h} = \sqrt{\cos(\theta-\varphi_k)\ell_{kp}^2 + \sin(\theta-\varphi_k)\ell_{ks}^2} \tag{7}$$

where θ is the angle between the horizontal axes and the the line connecting the two center of gravity. Then, we can compare the length of a path in the original graph with the clustered path and more in general the length of all possible paths connecting nodes of two clusters with the clustered path length $\ell^{h\leftrightarrow k}$.

Definition 2. *For each couple of adjacent clusters $\mathcal{C}_h$, $\mathcal{C}_k$ consider all the nodes $v_i \in \mathcal{C}_h$, $v_j \in \mathcal{C}_k$. For each couple v_i, v_j consider, if it exists, the shortest path connecting them on the original graph $\mathcal{G}$, denoted as $(v_i, v_j)_{\mathcal{G}}$. Then we define the inter-cluster length error $\mathcal{E}_{\ell_{ij}}$ as*

$$\mathcal{E}_{\ell_{ij}} = \frac{\sum_{v_i \in \mathcal{C}_h} \sum_{v_j \in \mathcal{C}_k} |(v_i, v_j)_{\mathcal{G}} - \ell^{h\leftrightarrow k}|}{\sum_{v_i \in \mathcal{C}_h} \sum_{v_j \in \mathcal{C}_k} (v_i, v_j)_{\mathcal{G}}} \tag{8}$$

The inter-cluster length error compares the aggregated path length with the average path lenght between two clusters. Clearly, as the size of clusters increases and as a consequence does the simplification, aggregation can introduce huge errors if a single path is considered. Consider Figure 11 and the three paths connecting cluster A and B: it becomes clear that if one compares $\ell^{h\leftrightarrow k}$ with either path 1 (red) or path 2 (green), the error introduced is big. With respect to path 3 (blue) however, the error is quite small and thus the approximation correct. In general, an opportune definition of the aggregated length allows to maintain the error with respect to the average path between two clusters limited. The error introduced depends drastically not only on the size but also on the shape of the cluster: clusters with symmetric shapes are more likely to have regular distribution of path lengths and thus $\mathcal{E}_{\ell_{ij}}$ can be maintained small. With the calculation of travel time in mind, we also define the aggregated path length for cluster which are not adjacent. Consider two clusters $\mathcal{C}_h$, $\mathcal{C}_{h+i}$ not adjacent, we define the aggregated path length as

$$\ell^{h\leftrightarrow h+i} = \ell^{h\leftrightarrow h+1} + \ell^{h+1\leftrightarrow h+2} + \ldots + \ell^{h+i-1\leftrightarrow h+i} \tag{9}$$

namely we consider half of the length for departure/arrival clusters and the full length for clusters which are completely crossed during the path. Consider Figure 12 and two possible aggregated paths (green and red), connecting the origin/destination clusters $\mathcal{C}_O$, $\mathcal{C}_D$. We have

$$\ell_a^{D\leftrightarrow O} = \ell^{D\leftrightarrow H} + \ell^{H\leftrightarrow M} + \ell^{M\leftrightarrow P} + \ell^{P\leftrightarrow O}$$

$$\ell_b^{D\leftrightarrow O} = \ell^{D\leftrightarrow C} + \ell^{C\leftrightarrow B} + \ell^{B\leftrightarrow E} + \ell^{E\leftrightarrow I} + \ell^{I\leftrightarrow O}$$

Clearly, for origin/destination clusters which are not adjacent, there might be multiple aggregated paths connecting the two and thus multiple aggregated path length: we define the set of aggregated paths connecting $\mathcal{C}_h$, $\mathcal{C}_k$ as $\mathcal{L}_{h\leftrightarrow k}$. In general, depending on which real path we want to compare our approximation with, we could choose the aggregated path that passes through the clusters which contains the sections of the real path. However, being this calculation really tedious and heavy, for non adjacent clusters we define the aggregated path length as

$$\ell^{h\leftrightarrow k} = \min_{i \in \mathcal{L}_{h\leftrightarrow k}} \{\ell_i^{h\leftrightarrow k}\} \tag{10}$$

It is important to stress that the choice of cluster parameters we introduced is *light* in terms of computational load: being our aim to exploit this clusters in a real time scenario, computing for instance the average path length (which ideally would lead to $\mathcal{E}_{\ell_{ij}} = 0$), for huge graphs is not feasible. Our choice instead, based only on geographic distribution of nodes allows to calculate these parameters in a time-window compatible with real-time applications. In fact, the complexity of calculating these parameters is linear with the number of nodes $\mathcal{O}(|\mathcal{V}|)$, where $|\mathcal{V}|$ is the cardinality of vertexes set. As mentioned in previous sections, the main comparison between original and clustered traffic network is based on traveling time calculation. In particular, we will compared the travel time between clusters with the average travel time

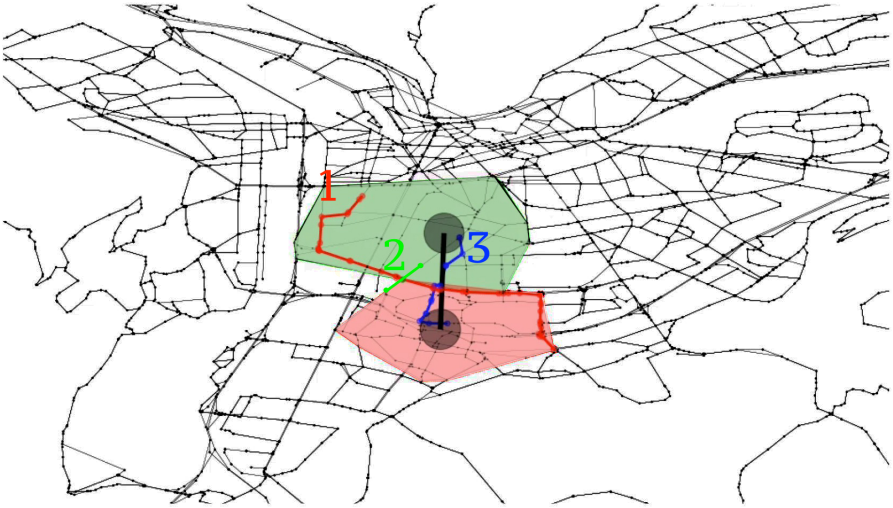

Figure 11: Two adjacent clusters and 3 different paths: in red the longest path 1, in green the shortest path 2, in blue path 3 close to the centers of mass.

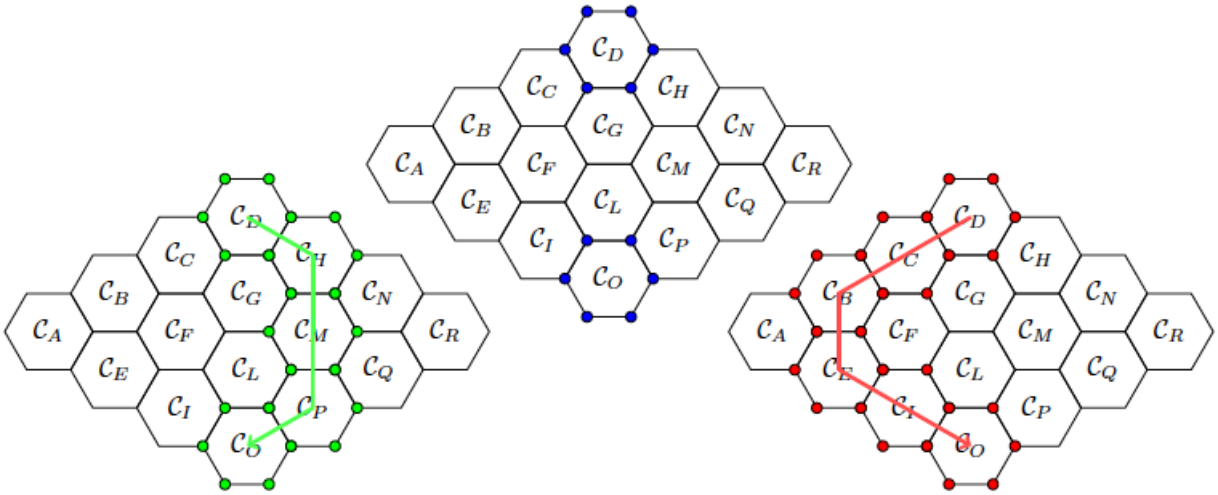

Figure 12: Example of clustered path: considering two origin/destination clusters, $\mathcal{C}_O$, $\mathcal{C}_D$ we have two possible paths in the clustered graph, one in green, one in red.

between origin/destination section chosen inside the departure/arrival clusters. It is worth recalling that from our point of view, if two clusters are adjacent, the travel time to go from one to another and *vice versa* is the same and it is unique.

Following the definition of the distance between two adjacent clusters (4), we define the cluster-to-cluster travel time as

$$\tau^{h\leftrightarrow k} = \frac{\ell^{h\to k}}{2v_h} + \frac{\ell^{k\to h}}{2v_k} \tag{11}$$

namely, we consider the sum of the travel time to go from the center of mass of clusters $\mathcal{C}_h$, $\mathcal{C}_k$ to the boarder

of the cluster, moving along the line connecting the two center of masses, respectively at the cluster average speeds v_h, v_k. Consider Figure 12 and suppose that we are interested in the travel time between the adjacent clusters $\mathcal{C}_A$, $\mathcal{C}_B$ then we have

$$\tau^{A\leftrightarrow B} = \frac{\ell^{A\to B}}{2v_A} + \frac{\ell^{B\to A}}{2v_B}$$

with $\ell^{A\to B}$ and $\ell^{B\to A}$ as in (5).

When two clusters $\mathcal{C}_h$, $\mathcal{C}_{h+i}$ are not adjacent, we define the traveling time as

$$\begin{aligned}\tau^{h\leftrightarrow h+i} &= \tau^{h\leftrightarrow h+1} + \tau^{h+1\leftrightarrow h+2} + \ldots + \tau^{h+i-1\leftrightarrow h+i} \\ &= \frac{\ell^{h\to h+1}}{2v_h} + \frac{\ell^{h+1\to h}}{2v_{h+1}} + \frac{\ell^{h+1\to h+2}}{2v_{h+1}} + \frac{\ell^{h+2\to h+1}}{2v_{h+2}} + \ldots + \frac{\ell^{h+i-1\to h+i}}{2v_{h+i-1}} + \frac{\ell^{h+i\to h+i-1}}{2v_{h+i-1}}\end{aligned}$$

namely we consider half of the length for departure/arrival clusters and the full length for clusters which are completely crossed during the path.

Consider again the example in Figure 12 and suppose we are interested in the origin/destination travel times between clusters $\mathcal{C}_O$, $\mathcal{C}_D$: to the two options that passes through clusters $\mathcal{C}_H$, $\mathcal{C}_M$, $\mathcal{C}_P$ (in green) and another passing through $\mathcal{C}_C$, $\mathcal{C}_B$, $\mathcal{C}_E$, $\mathcal{C}_I$ (in red). corresponds two possible travel times $\tau_a^{D\leftrightarrow O}$, $\tau_b^{D\leftrightarrow O}$. Similarly to the case of aggregated length for non-adjacent clusters, we define the aggregated travel time as

$$\tau^{h\leftrightarrow k} = \min_{i\in\mathcal{T}_{h\leftrightarrow k}} \{\tau_i^{h\leftrightarrow k}\} \tag{12}$$

where $\mathcal{T}_{h\leftrightarrow k}$ is the set of possible aggregated travel time between the two clusters. It is worth stressing again that this choice is really conservative but allows to compute aggregated travel time uniquely without considering detailed physical path. In the following Section we will indeed show that all these simplification have a *limited* impact on the estimation of travel time from an area to another. The following definition explains how we compare in details our aggregated travel time with the ones on the original graph.

Definition 3. *For each couple of adjacent clusters $\mathcal{C}_h$, $\mathcal{C}_k$ consider all the nodes $v_i \in \mathcal{C}_h$, $v_j \in \mathcal{C}_k$. For each couple v_i, v_j consider, if it exists, the shortest path connecting them on the original graph $\mathcal{G}$, denoted as $\tau(v_i, v_j)_{\mathcal{G}}$. Then we define the inter-cluster travel time error $\mathcal{E}_{\tau_{ij}}$ as*

$$\mathcal{E}_{\tau_{ij}} = \frac{\displaystyle\sum_{v_i\in\mathcal{C}_h}\sum_{v_j\in\mathcal{C}_k} |\tau(v_i,v_j)_{\mathcal{G}} - \tau^{h\leftrightarrow k}|}{\displaystyle\sum_{v_i\in\mathcal{C}_h}\sum_{v_j\in\mathcal{C}_k} \tau(v_i,v_j)_{\mathcal{G}}} \tag{13}$$

As a matter of fact, for large aggregation rates AR, there are several interesting paths which could lie inside a unique cluster. In order to take into account for these paths, we define also an intra-cluster aggregated travel time as

$$\tau^{h\circlearrowleft} = \min_{i\in\mathcal{T}_{h\circlearrowleft}} \{\tau_i^{h\circlearrowleft}\} \tag{14}$$

where $\mathcal{T}_{h\circlearrowleft}$ is the set of all possible travel times inside cluster h and each $\tau_i^{h\circlearrowleft}$ depends on a intra-cluster length

$$\ell^{h\circlearrowleft} = \sqrt{\cos(\theta-\varphi_h)\ell_{hp}^2 + \sin(\theta-\varphi_h)\ell_{hs}^2}\,.$$

Clearly, the minimal value coincides with the secondary length ℓ_{hs} of the cluster h (namely for $\theta = \varphi_h + \frac{\pi}{2}$), namely the shortest distance to *cross* the cluster.

Definition 4. *For a given cluster $\mathcal{C}_h$, consider all the nodes $v_i, v_j \in \mathcal{C}_h$ which are not neighbors. For each couple v_i, v_j consider, if it exists, the shortest path connecting them on the original graph $\mathcal{G}$, denoted as $\tau(v_i, v_j)_{\mathcal{G}}$. Then we define the intra-cluster travel time error $\mathcal{E}_{\tau_{ij}}$ as*

$$\mathcal{E}_{\tau_{ij}} = \frac{\sum_{v_i, v_j \in \mathcal{C}_h} |\tau(v_i, v_j)_{\mathcal{G}} - \tau^{h\circlearrowleft}|}{\sum_{v_i, v_j \in \mathcal{C}_h} \tau(v_i, v_j)_{\mathcal{G}}} \tag{15}$$

This choice of the intra-cluster aggregated travle time is motivated by the fact that, even though we neglect all trivial paths (namely paths which only include two adjacent sections), in our framework possible intra-cluster paths can be *really short*. More sophisticated choices could be made but we remark again that we are trying to simplify as much as possible the complexity of the overall architecture.

More in general, in terms of macroscopic traffic evolution, one could define a cutoff for the minimal length of the paths to be considered and thus restrict the analysis only to paths which play an important role into the traffic evolution. If *short* paths were neglected, a more accurate choice would coincide with the average of all possible $\tau^{h\circlearrowleft}$.

4. Travel Time Computation and Results

In order to validate our approach we wanted to compare our aggregated travel time with the travel time calculated over the original graph. To this end, we have selected eleven points of interest of the road network (see Figure 13) which represent the entry and exit points to the city network and some locations inside the city. The selection of this set of these points, labeled $\mathcal{POI}$, aims to cover the most important and typical traffic flows in the urban network of Grenoble. The urban targets are for instance the train station, the university campus and Grand Place (commercial area), while the entry/exit point to the city are the highways A41, A48 and A480. Thanks to this selection, we will be able to analyze the travel time evolution for typical paths that stay inside the urban area (typical citizens users) but also paths that cross the urban network (commercial transportation, commuters).

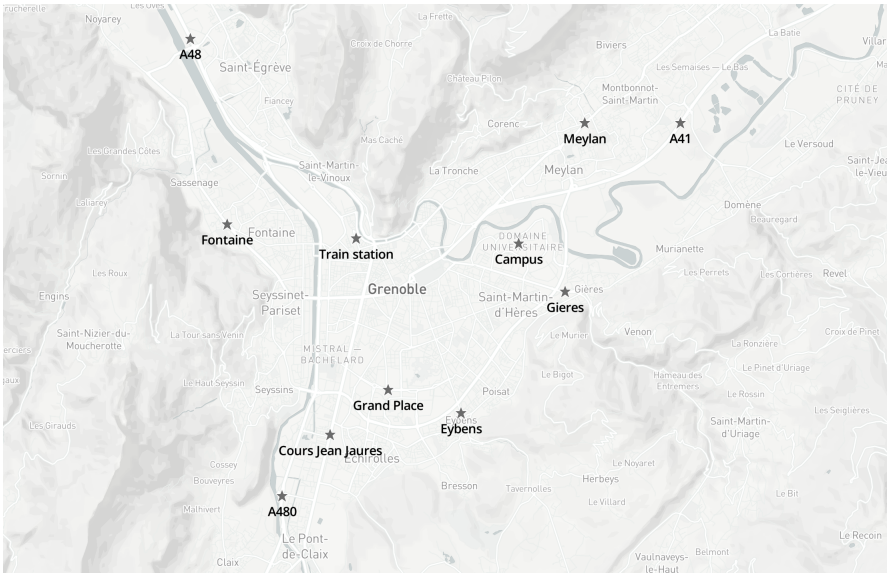

Figure 13: Selected points of interest of Grenoble area.

As describe in Section 3, our clustering is dynamic and thus during the day these points of interest will belong to different clusters or potentially lie in the same. Thus when calculating the evolution of travel times we will select the shortest travel time path in function of the traffic evolution. The comparison we present in the following relies on Definition 3-4 and in particular on the travel time errors (13)-(15) between the computation over the original graph and the aggregated graph. As mentioned before, these definitions are disadvantageous with many extents: for instance, since in the aggregated graph we chose the shortest travel-time path, this path could be substantially different from the one in the original network. Furthermore, in case two points of interest lie inside the same clusters, we will compare our estimate with the average of all the possible paths inside the clusters, namely we will consider a lot of paths which are not meaningful from a macroscopic perspective. Nevertheless we will show that even with high aggregation rate ($AR = 95\%$) we are able to keep the error limited (around 25%).

4.1. Evolution of the Clusters

The first aspect to be analyzed is the evolution of clusters during the evolution of traffic. As already explained in previous sections, the aggregation we perform is dynamic and thus is fundamental to understand how and why clusters changes. As a matter of fact, the more we aggregate the more stable the clusters: this behavior is actually expected by the definition of the clustering procedure since by increasing the cut-off we move towards a scale-free distribution.

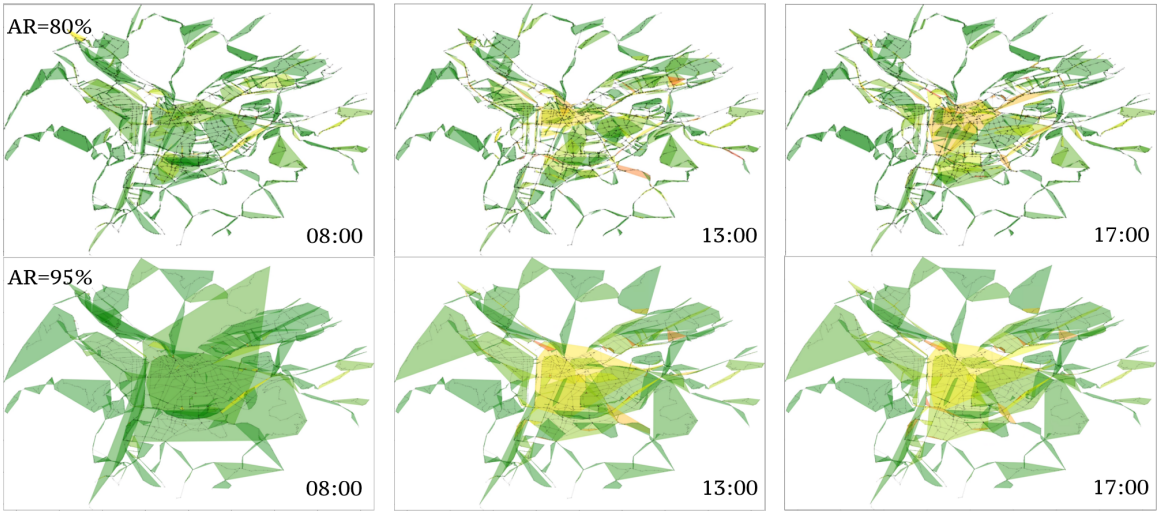

Figure 14: Clustering evolution along the day for $AR = 80\%$ and $AR = 95\%$. Different colors mean different fluidity, where green represents high fluidity and red low fluidity. The fluidity is defined for each clusters as the ratio between the actual average speed and the historic average speed.

As shown in R. Albert and A. L. Barabasi (2002), scale-free graphs are more resilient and robust to edges attacks. During the evolution of the traffic, the change of speed in a section such that the condition (2) is met or not can be seen as an attack that erase one of the edges in the graph $\mathcal{G}_A$. In the context of traffic, a similar analysis has been performed in D. Li et al. (2015), where the authors analyzed the sensitivity of a traffic network by means of percolation theory.

By increasing the cut-offs of the clustering procedure (and thus the aggregation rate AR) we obtain clusters that are more stable at the price of simplifying/loosing detailed information. In Figure 14 we show 3 samples of the evolution of the clusters during the day for two different aggregation rate, at three crucial moments of the day, namely 08h00, 13h00 and 17h00. We can clearly observe that, for $AR = 80\%$ the clusters changes drastically with the traffic evolution and while a repetitive pattern can be observed over a daily basis, it is much harder to find a stable structure in the clusters organization. On the other hand for $AR = 95\%$ the clusters are stable along the day and only little and marginal variation can be observed.

4.2. Result for $AR = 80\%$ and $AR = 95\%$

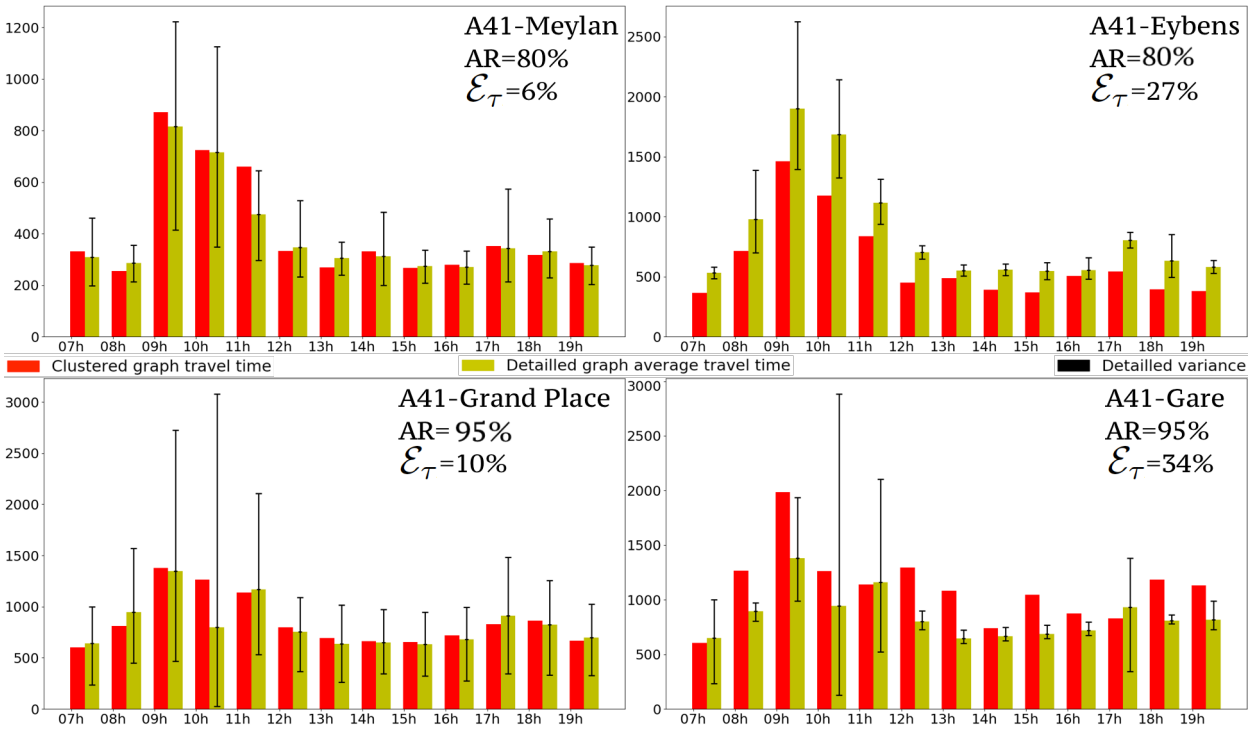

Figure 15: Travel time comparison between the aggregated graph and the detailed graph: evolution over the day, between 07h00 and 19h00, with the aggregated travel time (in red), the average of all the travel times in the detailed graph (in yellow) and the variance of these travel times (black line). Best and worst travel time per origin/destination are displayed for $AR = 80\%$ and $AR = 95\%$ respectively in the upper and lower line of the image.

In terms of travel time calculation, we expect the error introduced by clustering to grow linearly with the aggregation rate. As a matter of fact however, this error does not grow linearly with the aggregation rate and thus an opportune compromise has to be defined between the desired simplification and precision we want to keep. A comparison can be observed in Figure 15 where we considered two origin/destination paths for each of the two cases $AR = 80\%$ and $AR = 95\%$: in particular we chose the path where we commit the smallest error (from A41 to Meylan in $AR = 80\%$, from A41 to Grand Place in $AR = 95\%$) and the one where we commit the largest error (from A41 to Eybens in $AR = 80\%$, from A41 to Gare in $AR = 95\%$): the comparison is between the travel time calculated in the aggregated graph (red), the average of all the possible paths connecting the sections in the origin/destination clusters (yellow) and their variance (black). The aggregation were performed with $e_{[0,1]} = e_{[2]} = 4\,\text{km/h}$ and $e_{[3,4]} = 2\,\text{km/h}$ to obtain $AR = 85\%$ and with $e_{[0,1]} = e_{[2]} = 10\,\text{km/h}$ and $e_{[3,4]} = 6\,\text{km/h}$ to obtain $AR = 95\%$. Furthermore, the results presented in the following consider the time interval between 07h00 and 19h00, where TomTom data are more accurate and reliable.

It becomes immediately clear that the error between the aggregated travel time and the average of the traveling times between the two clusters defined in (15) is over the day acceptable in both cases (between 8% and 27% for $AR = 80\%$, between 10% and 34% for $AR = 95\%$) while the variance depends strongly on the aggregation rate selection. In particular, with $AR = 95\%$, the variance becomes in the worst case more than 250% of aggregated travel time. It is however interesting to notice that both aggregation rates preserve the traffic evolution over the day with a congestion peak early in the morning and another milder peak late in the evening.

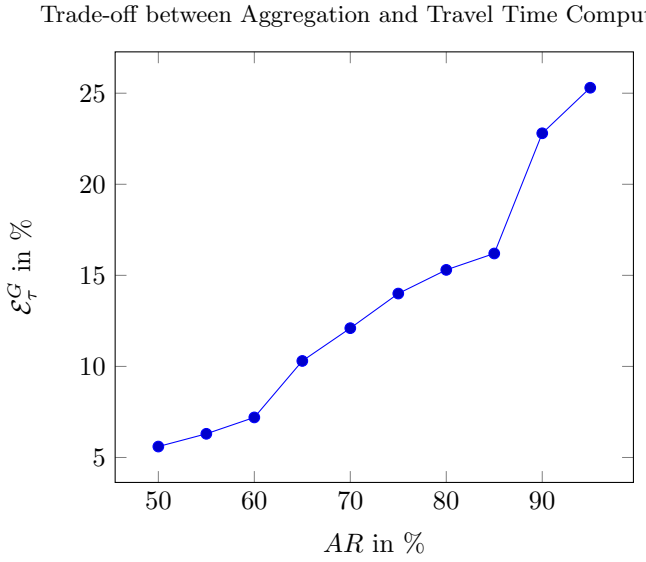

Figure 16: Evolution of the global inter-cluster travel time error $\mathcal{E}_\tau^G$ in function of the aggregation rate AR: both values are in percentage.

An important aspect to be observed is that in case of $AR = 80\%$, the worst case error is mainly a *underestimation* of the travel time while for $AR = 95\%$ the worst case error is clearly an *overestimation* of the traveling time. The former is due to the fact that, by aggregation, we approximate real paths with projections of clusters dimension: thus, as long as the simplification is not too high, we calculate paths which in general are much shorter than the real ones. The latter mainly comes from the *size* of clusters which, for really high aggregation rate, produces potentially really high aggregated length. As a consequence, even origin-destination clusters which are *close* might have aggregated path lengths that are really long. We want to stress that this is however a natural price to pay for the huge simplification introduced.

Another aspect that is interesting to observe is the evolution of the variance during the day. In particular, during rush hours (especially in the morning), travel time is really sensitive to the particular path chosen and thus the aggregated travel time can only aim to be correct in the average but the variance will be potentially important. On the other hand, during the day, when traffic smooths down, the variance is notably smaller until 18/19 p. m. when the second traffic peak is observed.

4.3. General Results

More in general, in order to compare the travel time calculation for different aggregation rate AR, we define a global error metric, namely a global inter-cluster travel time error

$$\mathcal{E}_\tau^G = \sum_{\mathcal{C}_h, \mathcal{C}_k \supset \mathcal{POI}} \frac{\sum_{v_i \in \mathcal{C}_h} \sum_{v_j \in \mathcal{C}_k} |\tau(v_i, v_j)_\mathcal{G} - \tau^{h \leftrightarrow k}|}{\sum_{v_i \in \mathcal{C}_h} \sum_{v_j \in \mathcal{C}_k} \tau(v_i, v_j)_\mathcal{G}} \tag{16}$$

where the notation $\mathcal{C}_h, \mathcal{C}_k \supset \mathcal{POI}$ means pairs of clusters containing pairs of points of interests. This global travel time error can be computed over different time windows, thus in order to obtain a big picture point of view we considered time windows of one week. This definition allows to estimate the evolution of the travel time error for increasing aggregation rate and the values, calculated over an entire week of datas, are reported in Figure 16. As a matter of fact, even when the aggregation rate is 95%, the global error is limited to 25% which with a certain extent is an astonishing result. Figure 16 shows also that the relationship

between the aggregation and the error introduced is not linear, thus it is necessary to carefully choose the aggregation cut-off to obtain the desired trade-off between AR and $\mathcal{E}_\tau^G$.

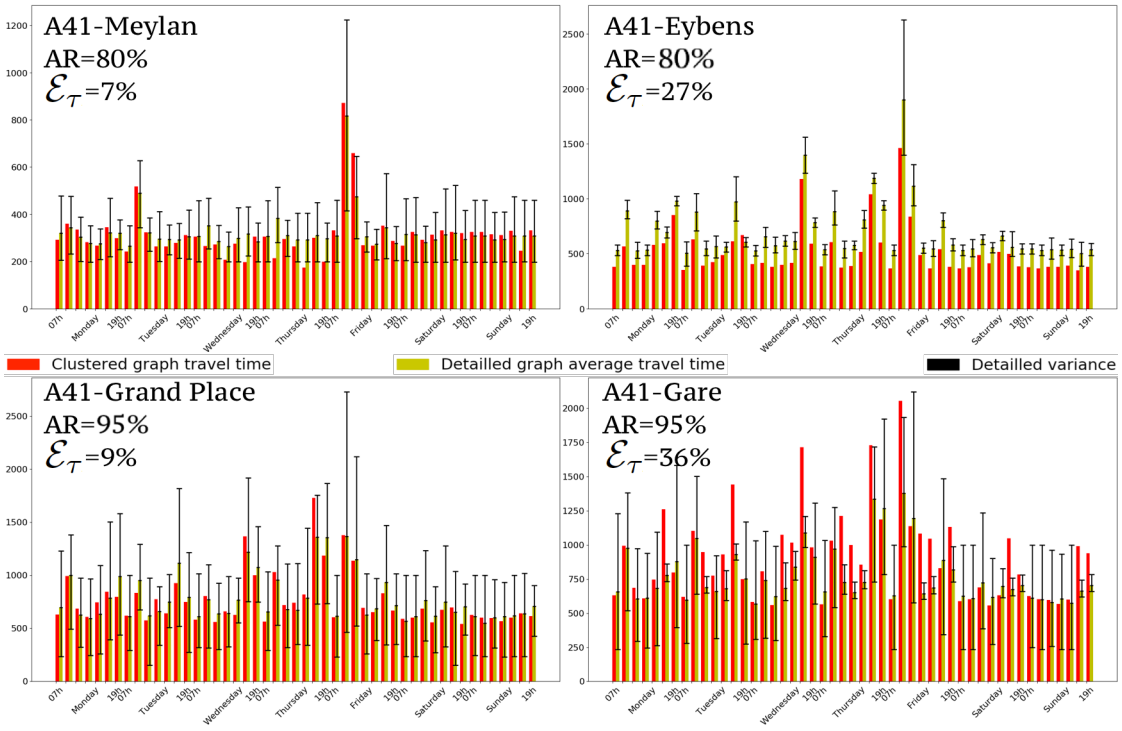

Figure 17: Travel time comparison between the aggregated graph and the detailed graph: evolution over a week, each day between 07h00 and 19h00, with the aggregated travel time (in red), the average of all the travel times in the detailed graph (in yellow) and the variance of these travel times (black line). Best and worst travel time per origin/destination are displayed for $AR = 80\%$ and $AR = 95\%$ respectively in the upper and lower line of the image.

In Figure 17, details about the evolution of the best and worst aggregated path over the week are shown. Both for aggregation rate $AR = 80\%$ and $AR = 95\%$, the best case error is below 10% while the worst case in respectively $\mathcal{E}_\tau^{80} = 27\%$ and $\mathcal{E}_\tau^{95} = 36\%$. Daily repetitive patterns can be observed, with congestion forming early in the morning and later in the afternoon: a particular event (namely a snowy day), can be observed on Friday morning resulting in a wide-spread congestion. Both displayed examples suggests however that there is the possibility to *predict* with a certain accuracy the travel time between different areas as long as the traffic evolution follows its natural behavior. Days with special events, accidents, adverse weather conditions, can be easily recognized even in the aggregated graph.

4.4. Complexity reduction

An interesting advantage of our architecture is the reduction of computation time to calculate huge numbers of origin/destination travel time. For different values of AR we thus compare the computation time for 250.000 (randomly chosen) paths in the original graph and the aggregated graph. Obviously, the algorithm to find the *shortest path* is the same for both cases, namely the Dijkstra's algorithm (see M. L. Fredman and R. E. Tarjan (1984)).

The result we present have been obtained with commercially available platforms and software: thus the absolute values of the values obtained might vary depending on the particular *hardware* setting available. As a reference, we obtained (roughly) 55 minutes of calculation time over the original graph. As it can be observed with the blue plot in Figure 18, the computation time reduces almost linearly with respect to the aggregation rate. Already for $AR = 60\%$, using the aggregated graph allows to save 50% of time:

for the aggregation rates we considered in previous simulations (namely $AR = 80\%$ and $AR = 95\%$), the computation time save ranges between 75% and 90%. In practical cases, this computation time gain has to find an opportune trade-off with respect to the precision required: in red we plot the error $\mathcal{E}_\tau$ relative to the 1000 paths chosen. As already remarked previously, the error in travel time computation is not linear with respect to AR and thus, depending on the desired accuracy and the time constraints, the compromise between these to aspects have to be clearly defined.

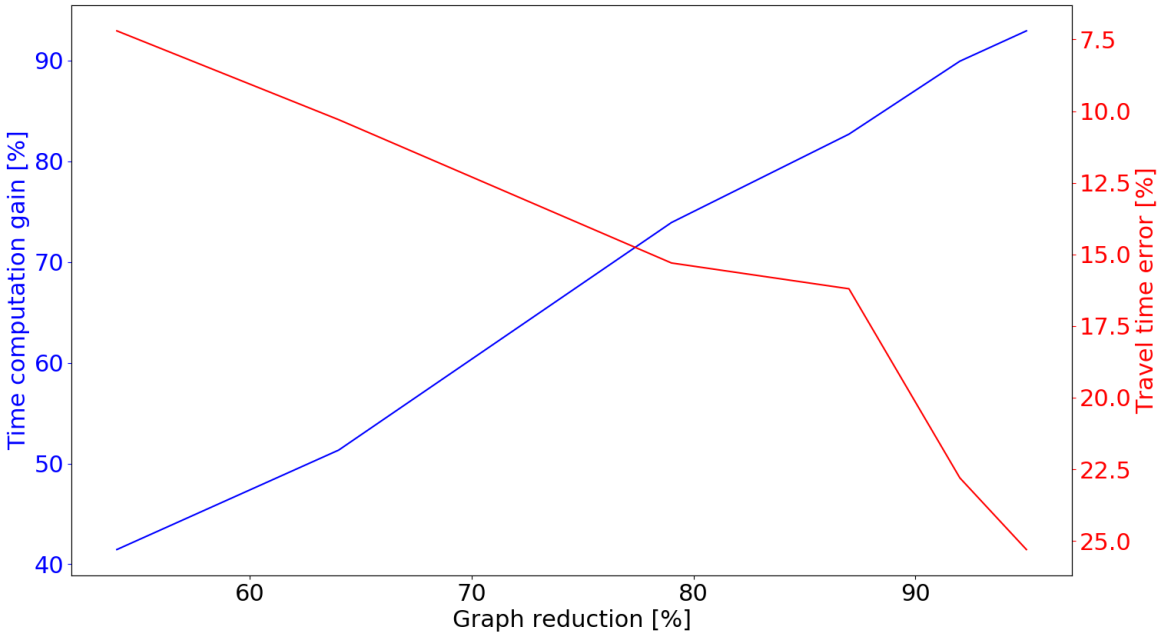

Figure 18: Trade off between reduction of complexity and precision in the calculation. For different AR, in red we plot the $\mathcal{E}_\tau^G$ while in blue we represent the gain in terms of computation time for calculating the shortest paths between 250.000 couples origin/destination.

It is also worth remarking that the gain in computation time is independent from the number of paths considered. As a matter of fact, the complexity of the Dijkstra's algorithm can be expressed as $\mathcal{O}\big(|\mathcal{E}| + |\mathcal{V}|\log(|\mathcal{V}|)\big)$, where $|\mathcal{V}|$ and $|\mathcal{E}|$ are the cardinality of the vertexes and edges set respectively. After the aggregation both the set of vertexes and edges is highly reduced: we can thus express the complexity of our approach as

$$\mathcal{O}\big(|V|\big) + \mathcal{O}\big(|\hat{\mathcal{E}}| + |\hat{\mathcal{V}}|\log(|\hat{\mathcal{V}}|)\big) \tag{17}$$

where where $|\hat{\mathcal{V}}|$ and $|\hat{\mathcal{E}}|$ are the cardinality of the vertexes and edges sets of the aggregated graph respectively. The first term in (17) is relative to the complexity of the aggregation algorithm (and thus depends on the original graph vertexes $|\mathcal{V}|$) while the second term is the complexity of the Dijkstra's algorithm over the reduced graph. As a matter of fact, $|\hat{\mathcal{V}}|$ and $|\hat{\mathcal{E}}|$ cardinalities vary along the day according to the aggregation and thus it is not possible to give an exact expression of the complexity reduction. However, guided by the intuition, we know that both the set of vertexes $|\hat{\mathcal{V}}|$ and the set of edges $|\hat{\mathcal{E}}|$ in the reduced graph are much *smaller* than the one of the original graph. Thus, we can conclude that *in the average*, the complexity of our architecture (namely aggregation plus travel times calculation over the reduced graph) is computationally much more convenient than computing over the original graph.

4.5. Discussion of the Result

The result presented in this sections all display that it is possible to reduce the complexity of the traffic network while preserving a fairly accurate description of traffic evolution. On one hand, we obtained that by setting the aggregation rate $AR = 95\%$ we introduce a global travel time error $\mathcal{E}_\tau^G = 25\%$ which in turn implies that the accuracy of multi-origin multi-destination travel time calculation does not require a dense and granular set of information. In other words, if we are interested in understanding traffic evolution from a macroscopic point of view (namely traffic patterns between one zone and another or inside a zone), we need much less information than the ones that are actually available. On the other hands, if we are interested in precision in travel time calculations, namely we seek a small global travel time error $\mathcal{E}_\tau^G$ with also a small variance, there is the possibility to *trade-off* between these factors and the aggregation rate: as a matter of fact, with a aggregation rate $AR = 80\%$ we are capable of keeping the error $\mathcal{E}_\tau^G = 15\%$.

Another interesting and promising element emerging from this analysis is that aggregated traffic evolution presents typical daily patterns. It can be observed that aggregation evolves similarly between *similar days* (working and not-working day), namely presenting the same congested areas for different moments of the day. Similarly, special *events* like strikes, adverse weather conditions and accidents in crucial arterial roads can be easily distinguished from the normal behavior of the network. This suggests that a travel-time *prediction* in a aggregated fashion might be possible: prediction over the aggregated graph would reduce enormously the computation, necessary parameters, measures and would be much more robust to *small variations* from the nominal behavior recorded in the past. An approach similar to the one proposed in A. Ladino et al. (2017) could allow to identify typical aggregated travel time patterns which describes normal and exceptional behavior of the traffic evolution and the prediction would be based on a *matching* between the historical data and the present time.

5. Conclusion

In this paper we considered aggregation as a mean to simplify the complexity of a large scale traffic networks. By aggregating neighbor sections which have a *similar* speed up to a cut-off parameter, we showed that it is possible to reduce drastically the size of the network while preserving the most important information. By defining a set of properties based on the shape each cluster, we showed that the error introduced by the simplification in the calculation of traveling time is absolutely acceptable: reductions in the order of 95% introduce error which in the average are less than 25%.

Clearly the approach we propose reduces the granularity of the data and the precision of calculus, namely over an exact path we can commit huge errors. However, if an average traveling time between two zones is considered, then aggregation allows to simplify the problem while preserving a sufficient accuracy. This allows to consider the traffic evolution from an aggregated point of view and simplifies the analysis of complex traffic patterns over days and weeks.

As already mentioned before, the natural extension of this work is to exploit the proposed technique to predict travel times: the idea we aim to pursue is to define typical patterns for aggregated travel time and compare the present traffic status with those historic *clusters*. Then, by merging the actual state with the patterns defined, it would be possible to predict the evolution of the aggregated travel time. Another possibility would be to exploit the aggregation in order to reconstruct density by zones: indeed in classic large scale frameworks, density sensors are much more sparse then velocity sensors. Thus density has to be reconstructed from velocity.

References

C. Canudas de Wit, ERC: Scale-Free Control for Complex Physical Network Systems, http://scale-freeback.eu .

C. Canudas De Wit, F. Morbidi, L. L. Ojeda, A. Y. Kibangou, I. Bellicot and P. Bellemain, "Grenoble traffic lab: An experimental platform for advanced traffic monitoring and forecasting", IEEE Control Systems 35 (3), pp. 23–39, 2014.

A. Ladino, A.Y. Kibangou, C. Canudas de Wit, H. Fourati, "A real time forecasting tool for dynamic travel time from clustered time series", Transportation Research Part C, 80, pp. 216–238, 2017.

R. Herring, A. Hofleitner, S. Amin, T. Nasr, A. Khalek, P. Abbeel, and A. Bayen, "Using mobile phones to forecast arterial traffic through statistical learning", Proc. 89th Transportation Research Board Annual Meeting, Washington D.C.,USA, 2010.
A. Bayen, J. Butler and A. Patire, "Mobile Millennium,CCIT Research Report, University of California", Berkeley, Institute of Transportation Studies (ITS), UCB-ITS-CWP-2011-6, 2011.
C. Peng, X. Jin, K. Wong, M. Shi, and P. Lio, "Collective human mobility pattern from taxi trips in urban area", PLOS ONE 7, e34487, 2012.
C. de Fabritiis, R. Ragona and G. Valenti, "Traffic Estimation And Prediction Based On Real Time Floating Car Data", Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, 2008.
A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing", Proc. Of Int. Conf. Information Processing in Sensor Networks (IPSN), St.Louis, USA, 2008.
H. Liu, and W. Ma, "A virtual vehicle probe model for time-dependent travel time estimation on signalized arterials", Transportation Research Part C,17(1), pp.11-26, 2009.
J.A. Deri, F. Franchetti, and J. M .F. Moura, "Big Data Computation of Taxi Movement in New York City", IEEE International Conf. Big Data, 2016.
B. Donovan and D.B. Work, "Using coarse GPS data to quantify city-scale transportation system resilience to extreme events", Transp. Res. Board 94th Annual Meeting (arXiv:1507.06011 [physics.soc-ph]), 2015.
P.S. Castro, D. Zhang, and S. Li, Urban traffic modelling and prediction using large scale taxi GPS traces, in Pervasive Computing, pp. 5772. Springer, 2012.
Y. Han and F. Moutarde, "Statistical Traffic State Analysis in Large-scale Transportation Networks Using Locality-Preserving Non-negative Matrix Factorization", IET Intelligent Transport Systems 7 (3), pp. 283-295, 2012.
H. Wang, Z. Li Y. Kuo and D. Kifer, "A Simple Baseline for Travel Time Estimation using Large-Scale Trip Data", arXiv:1512.08580, 2015.
H. Etemadnia, K. Abdelghany, A. Hassan, "A network partitioning methodology for distributed traffic management applications", Transportmetrica A: Transport Science 10, pp.518-532, 2014.
Y. Ji and N. Geroliminis, "On the spatial partitioning of urban transportation networks", Transportation Research Part B: Methodological 46, pp. 1639-1656, 2012.
M. Saeedmanesh and N. Geroliminis, "Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks", Transportation Research: Part B, a 23, pp. 962-979, 2017.
K. An, Y. Chiu, X. Hu and X Chen, "A Network Partitioning Algorithmic Approach for Macroscopic Fundamental Diagram-Based Hierarchical Traffic Network Management", IEEE Trans. on Intelligent Transportation Systems, 19(4), 2018.
N. Geroliminis and C. F. Daganzo, "Macroscopic modeling of traffic in cities", Transportation Research Board 86th Annual Meeting, 2012.
N. Geroliminis, C. F. Daganzo, "Existence of urban-scale macroscopic fundamental diagrams: some experimental findings", Transport. Res. Part B: Methodol. 42, 759–770, 2012.
H. Bar-Gera, S. Ahn, "Empirical macroscopic evaluation of freeway merge-ratios", Transport. Res. Part C: Emerg. Technol. 18, 457–470, 2010.
C. Buisson and C. Ladier, "Exploring the impact of homogeneity of traffic measurements on the existence of macroscopic fundamental diagrams", Transport. Res. Rec.: J. Transport. Res. Board 2124, 127–136, 2009.
Z. Zhang, B. Wolshon and V. V. Dixit, "Integration of a cell transmission model and macroscopic fundamental diagram: Network aggregation for dynamic traffic models", Transportation Research Part C, 55 (2015) 298–309, 2015.
L. Leclercq, N. Geroliminis, "Estimating MFDs in simple networks with route choice", Transport. Res. Part B: Methodol. 57, 468–484, 2013.
C. Lopez, L. Leclercq, P. Krishnakumari, N. Chiabaut and H. van Lint, "Revealing the day-to-day regularity of urban congestion patterns with 3D speed maps", Scientific Reports, 7, 2017.
C. Lopez, L. Leclercq, P. Krishnakumari, N. Chiabaut and H. van Lint, "Spatiotemporal Partitioning of Transportation Network Using Travel Time Data", Transportation Research Record: Journal of the Transportation Research Board, 2623, pp. 98–107, 2017.
S. Fortunato, "Community detection in graphs", arXiv:0906.0612v2 [physics.soc-ph], 25 Jan 2010.
F. D. Malliarosa and M. Vazirgiannisa, "Clustering and Community Detection in Directed Networks: A Survey", arXiv:1308.0971v1 [cs.SI], 5 Aug 2013.
G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, and A. Marchetti-Spaccamela, "Complexity and Approximation - Combinatorial Optimization Problems and Their Approximability Properties", Springer, 2nd edition, 2002.
D. Delling, R. Görke, C. Schulz and D.Wagner, "Orca Reduction and ContrAction Graph Clustering", AAIM 2009, LNCS 5564, pp. 152–165, 2009.
U. Brandes and T. Erlebach, editors, "Network Analysis: Methodological Foundations", LNCS, 3418, Springer, 2005.
S. M. van Dongen, "Graph Clustering by Flow Simulation", PhD thesis, University of Utrecht, 2000.
R. Kannan, S. Vempala, and A. Vetta, "On Clusterings: Good, Bad, Spectral", Journal of the ACM, 51(3), pp: 497–515, May 2004.
M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks", Physical Review E, 69 (026113), 2004.
R. Albert and A. L. Barabsi, "Statistical mechanics of complex networks", Rev. Mod. Phys, 74(1), pp. 47–97, 2002.
M. E. Newman, "The structure and function of complex networks", SIAM Rev 45(2), pp. 167–256, 2003.
D. Lia, B. Fua, Y. Wangc, G. Luc, Y. Berezind, H. E. Stanleye and S. Havlin, "Percolation transition in dynamical traffic network with evolving critical bottlenecks", PNAS, 112(3), pp. 669–672, 2015.

R. Albert and A. L. Barabsi, “Statistical mechanics of complex networks”, Review of Modern Physics, 74, pp. 47–97, 2002.
D. Li, B. Fu, Y. Wang, G. Lu, Y. Berezin, H. E. Stanley, S. Havlin, “Percolation transition in dynamical traffic network with evolving critical bottlenecks”, Proc. Natl Acad. Sci., 112, pp. 669-672, 2015.
M. L. Fredman, R. E. Tarjan, “Fibonacci heaps and their uses in improved network optimization algorithms”, 25th Annual Symposium on Foundations of Computer Science, pp. 338–346, 1984.