Future Generation Computer Systems 114 (2021) 649–660

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Traffic flow reconstruction by solving indeterminacy on traffic distribution at junctions

Stefano Bilotta [1], Paolo Nesi [*,1]

DISIT Lab, Department of Information Engineering, Università degli Studi di Firenze, Firenze, Italy

ARTICLE INFO

Article history:
Received 5 March 2020
Received in revised form 2 July 2020
Accepted 14 August 2020
Available online 27 August 2020

Keywords:
Smart city
Reconstruction algorithm
Traffic flow
Parallel computing approach
GPUs

ABSTRACT

The knowledge of the real time traffic flow status in each segment of a whole road network in a city or area is becoming fundamental for a large number of smart services such as: routing, planning, dynamic tuning services, healthy walk, etc. Rescue teams, police department, and ambulances need to know with high precision the status of the network in real time. On the other hand, the costs to obtain this information either with direct measures meant to add instruments on the whole network or acquiring data from international providers such as Google, TomTom, etc. is very high. The traditional modeling and computing approaches are not satisfactory since they are based on many assumptions that typically are doomed to change over time, as it occurs with traffic distribution at junctions; in short they cannot cover the whole network with the needed precision. In this paper, the above problem has been addressed providing a solution granting any traffic flow reconstruction with high precision and solving the indeterminacy of traffic distribution at junctions for large networks. The identified solution can be classified as a stochastic relaxation technique and resulted affordable on a parallel architecture based on GPU. The result has been obtained in the framework of the Sii-Mobility national project on smart city transport systems in Italy, a very large research project, and it is at present exploited in a number of cities/regions across Europe and by a number of research projects (Snap4City, TRAFAIR) of the European Commission.

1. Introduction

Often, the traffic flow estimation is related to the monitored area in terms of short-term traffic flow prediction on fixed points and thus no information is provided in the roads where sensors have not been located [1–5]. The usage of large number of traffic sensors can help on estimating directly the traffic flow in the whole city, but costs would not be affordable. On the other hand, the knowledge of traffic flow/density in real time and along almost all roads can be crucial for: (i) route planning for private services, rescue teams, ambulances, fire brigades, etc.; (ii) decision taking to support intelligent transport system for traffic reshaping, with the aim of preventing and/or reducing congestions, regulating traffic lights at intersections, etc.; (iii) prediction and analysis about the production of pollutants such as NOx [6]; (iv) the definition of sustainable mobility strategies, planning urban mobility services, etc.

The *Traffic Flow Reconstruction* is the process to produce a value of traffic density (flow) – e.g., vehicle per meter (vehicles per second) – for each road (or road segment, or a large number of road segments) by starting from a limited number of traffic sensors measuring traffic density (flow) in the road. The measures of traffic density are typically obtained by stationary sensors on strategic positions of different kinds. For example, at the intersections and/or on the road far from the intersection when vehicles are moving with a regular velocity.

A different approach can be based on extracting traffic density from the velocity of the moving vehicles (private cars, public buses, taxi, etc.). In fact, the OBU (on board units) and/or mobile devices can be endowed of a Global Positioning System (GPS) [7]. Similar data could be obtained by insurance assistance boxes. They can be regarded as mobile sensors, see [8,9], where Kalman filtering has been used to produce a more regular solution reducing noise and discontinuities. In most cases, the user engagement (at least to signify his/her assent) is needed to collect vehicles trajectories and velocity data from moving vehicles, as in [10]. Thus, once trajectories are known, it becomes also possible to compute origin–destination (OD) matrices [11]. Similarly, typical vehicle trajectories and velocity data can be obtained from the navigators installed on vehicles (if they are connected to the network to get updates and also for communication purposes) or on mobile Apps (e.g., TomTom, Google map, Waze, Here). Solutions aimed at fusing the different kinds of data sources have

* Corresponding author.
E-mail addresses: stefano.bilotta@unifi.it (S. Bilotta), paolo.nesi@unifi.it (P. Nesi).

[1] https://www.disit.org, https://www.km4city.org, https://www.snap4city.org

https://doi.org/10.1016/j.future.2020.08.017

been proposed, as well [12]. Mobility strategists of cities acquire traffic flow data from companies and insurances, which provide typical traffic flow data in fixed daily time slots for all the major road segments of the city. Those data are in most cases computed as an average of the values in the time slot, because the data coming from navigators are also scattered and affected by errors due to a personal driving style and individual needs.

In general, to set up a network of traffic flow sensors in a city drastically avoids the costs of taking updated data from third parties. For the sustainability, the number of deployed sensors has to be limited and thus to exploit traffic flow reconstruction solutions to obtain the value of traffic flow in the unmeasured road segments is mandatory to have a global view of traffic city conditions. Thus, in certain cases, a network area as freeways or rings, where incoming and outcoming roads have a limited number of elements, is monitored by consider traffic flow sensors as in [13–15]. When traffic flow sensors are available, traffic flow reconstruction is performed by using different computational approaches; typically by adding a number of additional constraints to produce realistic solutions from very scattered data sources, producing noisy data with discontinuous data in space and time.

The problem of the *traffic flow reconstruction* described above is also regarded as the solution of the LWR (Lighthill–Whitham–Richards) model [16,17]; which is modeling the traffic density in terms of Partial Differential Equation (PDE). The solution of the LWR model is not a trivial matter for large networks due to its computational complexity and constraints [18–20]. The estimation of the traffic distribution on junctions plays a crucial role on the effectiveness of the LWR model application in real context. Many studies have focused on the mathematical aspects concerning the solution at the junctions in the theory of LWR model [18,21,22]. On the other hand, the current literature is quite poor about the parameter estimation applied to the traffic junctions in terms of percentage of vehicles getting out each outcoming road with respect to those getting in each incoming road, so as to simulate realistic scenarios by using LWR model. In general, when a model has been applied on large road network, such a percentage at the junctions has been assumed constant or predefined [23,24].

Actually, vehicle behavior changes along the day and the effectiveness of any applied model can be negatively influenced by assuming as fixed the parameters involving traffic distribution at the junctions. In order to compute more precise models, the traffic distribution at the junctions should be not assumed as constant or predefined. A large part of literature has studied only small road-segments where the behavior is known at a signalized traffic intersection, since a control sensor is installed per arm and the estimation of the distribution is given [25–27]. The most widespread solutions for macroscopic models are: the Cell Transmission Model (CTM) [23], and approaches based on queues and on the principle of vehicle conservation [24]. As stated in [27], the CTM is likely to be the most precise, but problematic also for its complexity on large traffic networks, because the junction has to be divided into a large grid of cells. The distribution at junction is part of the wider problem of traffic flow reconstruction by using a limited number of control sensors. The problem becomes more difficult when such sensors are installed far from the traffic intersection as it occurs in the cases this paper copes with. This problem is also wider and more complex mathematically and computationally. Therefore, a parallel solution could make it easier to compute for the LWR model applied to a large urban network where parameter estimation on traffic distribution at the junctions is needed. In the literature, several partitioning algorithms, along with their strengths and weaknesses for various PDE applications have been reviewed in [28]. For example, some preliminary studies can be seen in [29,30] where concepts are applied to a small number of roads (or a single road), to make it computationally affordable. In a recent study [31], a deep learning neural model has been presented for the prediction of traffic conditions from probe vehicles within car-navigator involving the implementation of parallel structures by innovative instruments like TensorFlow. Also, some studies have been carried out in [32] where a (unreal-world) simplified urban-road network is developed for Parallel-transportation management system applications.

1.1. Aim of the paper and its overview

In this paper, a solution is presented for computing traffic flow reconstruction by solving the indeterminacies about traffic flow distribution at junctions. The solution we proposed addresses at the same time (i) the traffic flow reconstruction by solving PDE; (ii) the estimation of traffic flow distribution at junctions with a solution that can be classified as a stochastic relaxation technique by means of a meta-heuristic approach; (iii) the real time computation of (i) and (ii) in large road networks exploiting tensor flow modeling and GPU; (iv) the identification of the error function in traffic flow reconstruction, thus demonstrating its relation with road network features. The result has been produced in the framework of Sii-Mobility, the Italian national research project on mobility and transport for smart city funded by the national Ministry of research (http://www.sii-mobility.org). This is a 4 years national project involving research centers and industries. The algorithms have been put in execution by exploiting the Km4City data aggregator and its semantic model (https://www.km4city.org) [33,34]. The solution is at present exploited in a number of research projects such as Snap4City [35], TRAFAIR [6], RESOLUTE [36], and in a number of cities: Firenze, Pisa, Livorno, Modena, Santiago di Compostela, see the dashboard accessible online on the Snap4City open service https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc5NQ==

The paper is structured as follows. In Section 2, the mathematical model for traffic flow is presented, addressing road network and traffic junctions. Section 3 presents the solution for traffic flow reconstruction with first approximation on junction distribution. In Section 4, the complete computational solution is presented, while addressing also the computing of the so-called Traffic Distribution Matrices that model the distribution of traffic flow at junctions. In the same section details about the adopted parallelization approach are reported. Section 5 presents the error analysis which allowed us to assess the solution precision and to identify the dependency of the error estimation on the road network topology in terms of road network betweenness and eccentricity. In Section 6, such experimental results have been reported, thus providing evidence of a real time performance obtained in computing the solution. The above reported URL to dashboard is accessible and allows a real-time verification on the solution validity. Conclusions are drawn in Section 7.

2. Modeling traffic flow

The traffic flow in road networks can be modeled by using a fluid dynamics model of vehicular traffic as in [18]. It is the so called LWR PDE model and equation. The nonlinear model is based on the conservation of vehicles described by the following scalar hyperbolic conservation law:

$$\frac{\partial \rho\,(t,x)}{\partial t} + \frac{\partial f\,(\rho\,(t,x))}{\partial x} = 0, \tag{1}$$

where: $\rho\,(t,x)$ is the traffic density of vehicles, which admits values from 0 to $\rho_{\max}$, where $\rho_{\max} > 0$ is the maximal traffic density; $f\,(\rho(t,x))$ function is the vehicular flux which is defined by means

of the product $\rho(t,x)\, v(t,x)$, where $v(t,x)$ is the vehicle speed; and boundary conditions $\rho(t,a) = \rho_a(t)$, $\rho(t,b) = \rho_b(t)$, initial values $\rho(0,x) = \rho_0(x)$, with $x \in (a,b)$. In the case of first order approximation, we assume that $v(t,x)$ is a decreasing function, depending on the density, then the corresponding flux is a concave function. Thus, we consider the local speed of the vehicles as $v(\rho) = v_{max}(1 - \frac{\rho}{\rho_{max}})$ and then $f(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)\rho$, where v_{max} is the limit speed on a given road segment.

Eq. (1) can be considered valid only when the hypothesis of the conservation of flow is satisfied at each time instant. The conservation can be verified by counting the inflow and outflow vehicles of the considered area in the time slice, for example in 15 min. Thus, the condition can be also generally applied, by knowing the value of the flow at the border, see for example [8]. And the flow at the border of the area can be measured by traffic flow sensors. So that, the traffic flow at a given time instant of a road network consists of the values of the traffic density in each point of the network, which is obtained by solving equation (1), that is the so called *traffic flow reconstruction*.

2.1. Solving traffic flow equation

Let u be the solution of Eq. (1). The discretization of (1) at finite difference for the evolution of the traffic density in each road which is composed by a time–space region (h,m), can be formalized as:

$$u_m^{h+1} = u_m^h - \frac{\Delta t}{\Delta x}\left(F\left(u_m^h, u_{m+1}^h\right) - F\left(u_{m-1}^h, u_m^h\right)\right), \tag{2}$$

where F denotes the flux, which is computed by taking into account the physical constraints of the selected road and its connections to the neighbor road-segments according to:

$$F(w,z) = \begin{cases} \min\left(f(w), f(z)\right), & w \le z \\ f(w), & z < w < \rho_c \\ f_{max}, & z < \rho_c < w \\ f(z), & \rho_c < z < w \end{cases}$$

where f_{max} denotes the maximum value for the vehicular flux permitted in the selected road and such a value is considered when the vehicular density assumes its critical value ρ_c which depends on the capacity of the given road.

In order to set up the finite difference scheme in a complex road network, boundary conditions at the border of the area and conditions at the junctions regarding traffic distribution have to be imposed. That is, provided that the conservation of flow and boundaries conditions are satisfied, the solution of Eq. (1) on large network of roads is not a trivial matter due to the presence of (i) several and topologically diverse intersections/junctions with unknown behavior (distribution of flow over time), (ii) unknown in the data regarding flow density in all segments of the network, high noise and discontinuities in the measures, thus resulting in a (iii) high computational complexity. In fact, several thousands of road segments and several hundreds of junctions can be present even in small cities with 100.000 inhabitants. Therefore, most of the applications based on Eq. (1) are based on discrete numerical solutions at finite differences [19,20], and they are applied on specific configurations for junctions, small networks and dense data. In [8], Kalman filtering has been used to cope with noise modeling/reduction and the presence of small discontinuities due to the lack of data/measures coming from sensors.

2.2. Road network and junctions

A **road network** is modeled as an oriented graph composed by segments connected at nodes; nodes may be junctions where more than 2 roads meet. In a discretized version of the model, a road between two junctions can be regarded as composed by a series of units/segments of size Δx. Similarly, the time can be divided in time slots, of Δt. The time–space region bounded within a duration h and unit m is referred to as a *cell* and it is denoted as (h,m); and the number of vehicles contained in unit m at the end of duration h (at a given time instant) is denoted as $n(t_h, x_m) = n(h\Delta t, m\Delta x)$. We also use notation u_m^h for $u(t_h, x_m)$ when u is a function on (t,x) plane. Moreover, the relationship $\frac{\Delta x}{\Delta t} > v_{max}$ is required in the discretization scheme, that is, a vehicle should not cross more than one unit Δx within a time step Δt. On the other hand, this is not a strong restriction if the time and space slots are too small with respect to the vehicle velocity and size.

In order to model the traffic distribution at **junctions**, a distribution matrix can be used to describe the percentage of vehicles getting out each outcoming road with respect to those getting in each incoming road. Thus, the *Traffic Distribution Matrix* (TDM) can be defined as $TDM = \{w_{ji}\}_{j=n+1,\ldots,n+m, i=1,\ldots,n}$ so that $0 < w_{ji} < 1$ and $\sum_{j=n+1}^{n+m} w_{ji} = 1$, for $i = 1,\ldots,n$ and $j = n+1,\ldots,n+m$, where w_{ji} is the percentage of vehicles arriving from the ith incoming road and taking the jth outcoming road (assuming that, on each junction, the incoming flux coincides with the outcoming flux). The real values of w_{ji} may depend on the time of the day, on the road size, cross light settings, etc., and thus, it is unknown a priori. In the following, w_{ji} coefficients are called *weights*.

$TDM(t)$ over time is unknown, since the traffic flow is not measured in each inflow/outflow road of all junctions, in each time slot t. A first approximation of the $TDM(t)$ could be the typical values for $TDM(t)$ in the day, for each day of the week. These trends are an approximation, since the $TDM(t)$ is conceptually changing at each time slot. Thus, the first approximation can be produced by means of a computation of typical trends at the junctions over time, in the last period (for example, the last two months).

2.3. Boundary conditions vs junctions

The boundary conditions occur at the border of the road network where some road segments have one of their nodes not connected to any junctions. For this purpose, considering the model of Eq. (1) without loss of generality, we assigned a condition at the incoming boundary for $x = 0$ as $\rho(t,0) = \rho_b^{inc}(t)$ and thus computing in that point the solutions of Eq. (1) only for $x > 0$. In practice, we proceed by inserting an "incoming ghost cell" and the discretization becomes:

$$u_0^{h+1} = u_0^h - \frac{\Delta t}{\Delta x}\left(F\left(u_0^h, u_1^h\right) - F\left(\rho_{(inc)}^h, u_0^h\right)\right) \tag{3}$$

where:

$$\rho_{(inc)}^h = \frac{1}{\Delta t}\int_{t_h}^{t_h+1} \rho_b^{inc}(t)dt,$$

replaces the "*ghost value*" u_{-1}^h. The same equation can be used for segments where the measure of flow density is known. For example, where sensors are located.

Of course, let $x < N$, for the outcoming boundary we can consider a similar procedure leading to:

$$u_N^{h+1} = u_N^h - \frac{\Delta t}{\Delta x}\left(F\left(u_N^h, \rho_{(out)}^h\right) - F\left(u_{N-1}^h, u_N^h\right)\right). \tag{4}$$

As to the conditions at the road junctions and thus on TDM aspects, the following discretization for roads connected to a junction at the right and left endpoint is modeled, respectively:

$$u_N^{h+1} = u_N^h - \frac{\Delta t}{\Delta x}\left(\gamma_i - F\left(u_{N-1}^h, u_N^h\right)\right) \tag{5}$$

$$u_0^{h+1} = u_0^h - \frac{\Delta t}{\Delta x}\left(F\left(u_0^h, u_1^h\right) - \gamma_j\right), \tag{6}$$

where: γ_i, γ_j are the incoming/outcoming flows such that $\gamma_j \simeq w_{ji}\gamma_i$ in accordance with the definition of the Traffic Distribution Matrix. And thus, only if the traffic flow distribution at junctions is known ($TDM(t)$), then a solution could be computed.

3. Traffic flow reconstruction

The traffic flow reconstruction consists of the estimation of traffic density in each road segment of the network. This also implies the knowledge of the $TDM(t)$ at junctions over time. In order to produce the traffic flow reconstruction, the model formalized in the previous Section 2 has to be solved. The solution of Eq. (1) on a real road network with several junctions presents as unknowns both: (i) the traffic density in each road segment and (ii) the $TDM(t)$ weights for the junctions at each time slot. In this section, we focused on solutions (1) to estimate a traffic density (i) with some simple assumptions on (ii). On the other hand, such simple assumptions could be improved, whenever a more precise estimation of $TDM(t)$ weights is addressed as computational problem in Section 4. Please note that, the estimation of $TDM(t)$ weights in a road network on the basis of the actual traffic has a value useful to understand traffic distribution, thus tuning intersection red lights.

In addition, the solution has to take into account:

- **road network details** such as: (i) traffic restriction determined by vehicle kind for each road due to restricted traffic zones, high speed roads; (ii) road flow direction; (iii) road size in terms of lanes; (iv) road segment speed limit; (v) constraints on U turns; etc. Those aspects mainly change the capacity of the road segment.
- **traffic density status**, which may not be available in all road segments. In fact, sensors may provide measures on scattered points, as well as measures of velocity provided by OBU and/or navigator, which may be affected by personal needs and/or driving styles.
- **boundary conditions** in the computation of the density.
- **conservation of flow** in the general model and data, with respect to road network area considered.

Please remind that, the solution of Eq. (1) should:

- allow to know the status of the traffic flow in all road segments, with the above constraints, and despite the fact that the measured traffic flow is only performed in a limited number of points and with noisy conditions.
- be produced in real time; that means with a very small delay after the last measure of the traffic flow sensors.

The traffic flow reconstruction in all road-segments is useful for: route planning, traffic regulation, public service, anomaly detection, prediction of certain pollutants. Thus, only by taking into account of all the above-mentioned conditions the results can be realistic with respect to the actual traffic experiences by the users on the road.

Due to the above requirements, unknowns and constraints the solution is highly computationally intensive both for a wide area and a small city where it is easy to have some tens of thousands of road segments and several hundreds of junctions.

Therefore, new method and solution have been identified and initially implemented on a traditional multi-thread CPU based computer, later a GPU based solution has been developed to strongly reduce the execution time, as described in this paper.

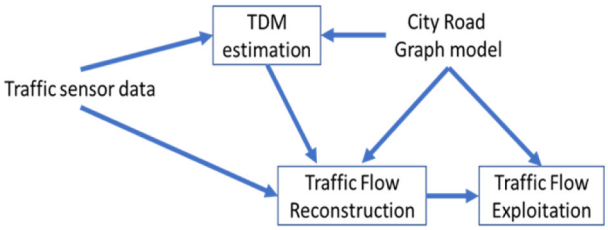

Fig. 1. Schema of the computational approach.

In the present section, we describe the main computational steps involving the traffic flow reconstruction model: in part A, the parameters occurring in the TDM computation are presented, part B describes how the traffic sensors data are inserted in the model, part C is focused on algorithm, part D describes the network graph structure and the traffic density computation on the basis of equations of Section 2, part E is devoted to the graphical representation of the resulting traffic flow. Fig. 1 describes the schema of the computation.

3.1. Traffic distribution matrix, TDM

When the traffic density is unknown and also the $TDM(t)$ at junctions is unknown, Eq. (1) is under-constrained, and may become solvable by using a large number of traffic density measures over time. Moreover, Eq. (1) is computable as stated in Section 2 having at least a first approximation of $TDM(t)$. To this end, the weights of $TDM(t)$ can be produced/guessed by: (i) taking into account the relevance/size of the roads at the junctions, road-segment type (motorway, trunk, primary, secondary, tertiary and residential), road-segments' lane number, maximum flux permitted in road-segments (by size), turn restrictions, restricted traffic zone, speed limit in road-segment, road-segment access restriction, bidirectional traffic road-segments, presence of designated lanes for public transport, etc., (ii) taking into account the census data, as well as the attractions (monuments, events, etc.) in the area, (iii) measuring the typical trends of the traffic distribution at the junctions in the last period (for example, for the last two months).

3.2. Traffic sensors' modeling

TV Cameras and spire road sensors are typically located on: (i) highway/main roads in the country side, (ii) main roads arriving/departing to/from a city, as well as in the main road leading to the city, (iii) main junctions (recently this kind of sensors are less used since the traffic flow is influenced by red lights), (iv) main gates at Restricted Traffic Zone access points. Each sensor gives the state of the traffic by counting the number of vehicles (for example, by vehicle kind) which pass through the supervised area and the data are simultaneously updated every *time slot* (for example every 5–10 min). Each traffic flow sensor presents static information related to its geographical coordinates and assessed flow direction, number of observed lanes, etc. The integration of traffic sensors in the road network graph corresponds to its association to road segment. So that, a tv-cam sensor may assess traffic flows in both directions and on multiple lanes of that road segment. For each time slot t, the traffic sensors can measure traffic density in the road segment where they are located. In particular, the measured traffic density $\rho_M(t)$ is the value in the outcoming roads of such segment. If the initial node of a road-segment corresponds to the location of a sensor, then the problem related to the integration of data sensor in the graph is solved by inserting the measured data $\rho_M(t)$ (to be properly distributed if

the node corresponds to a junction) in the corresponding road-segment according to Eq. (3) where $\rho(t, 0) = \rho_M(t)$. In particular, a smoothing solution is given and the density value in the associated segment passes from $\rho_M(t)$ to $\rho_M(t')$, being t the previous time slot.

The traffic flow sensors can be considered a more precise estimation of vehicular flow and traffic density with respect to the measures produced by OBU or navigators which are mobile measures and they can be taken into account in [8] in the LWR. Traffic sensors are less affected by errors due to the higher number of vehicles used for the measures. For example, a navigator could provide instantaneous velocity of a single vehicle which can go spontaneously slow even being the only one in the road. Thus, producing a wrong traffic density.

3.3. Basic computational approach

Therefore, if exploiting equation (1), and taking into account the values of a relevant number of traffic flow sensors, it is possible to infer the distribution of traffic in the rest of the network and the weights of the $TDM(t)$. Thus, having a road network graph for a given area (including, road details, junctions, sensors, and $TDM(t)$) becomes possible by means of a relaxation method to estimate the traffic density in real-time, according to the following steps:

Load road network and details.
For each *Time t*:
  Get traffic sensors' values
  Load *TDM(t) and weights*
  For *h: 1 → H*
    Compute the model for each sensor' location to the sensors' value
    Compute the model for all traffic distribution in each junction
    Compute the model for all traffic density in each road-segments
  End For h
  Compute graphics representation
End For each Time

For each time slot t, after H iterations all the road-segments in the road network have an estimated traffic density value. In the following, details of the computational steps of the mentioned algorithm are described, including the computing of the number of H iterations which is most suitable to minimize errors, see Section 5.

3.4. Traffic density computing

Each road segment with its attributes (such as: number of lanes, restrictions, kind of road, etc.), provides a max limit of traffic density (and flux), that is defined in terms of vehicular capacity. According to Section 2.2, a bi-dimensional *density array* of size $h * s$ is associated with each road having s units and the value in the (h, m)-*th* cell representing the estimated traffic density in the mth unit of a road-segment at the duration h, (time instant $h\Delta t$) with $1 \leq m \leq s$. The density array is partitioned in L units according to the length of the road-segment itself. The value in each unit is computed at the finite difference according to Section 2. In the following, we give some details about such a procedure which depends on both the position the unit assumes in the density array and the role played by the associated road-segment in the road network.

Every time, measures collected by sensors are updated /considered, the traffic density value is propagated from the road-segments where sensors are located, by applying to the first unit of the associated density arrays the procedure described in the part B of the present section.

If a road-segment is located on the **border of the road network**, then we distinguish two cases depending on the direction of the road-segment with respect to the graph. If the road-segment represents an incoming edge of the graph, then the first unit of the associated density array is computed according to Eq. (3) where a bounding value or "ghost-value" is introduced (such value can be interpreted as a "ghost-sensors" in the graph). Otherwise, if the road-segment represents an outcoming edge of the graph, then the last unit of the associated density array is computed according to Eq. (4).

Generic road-segments are not placed in the graph border and they do not host any sensor, so they represent both an incoming road-segment in its end node and an outcoming road-segment from its initial node. Therefore, the first unit of the associated density value is computed according to Eq. (6) while the last one is computed according to Eq. (5). Of course, such computation depends strongly on weight assignment since it determines the turn percentage and related traffic (flux) distribution on the junction.

Finally, for each **road-segment having more than 2 units**, all the units which differ from the first and the last are computed according to Eq. (2).

At the next time slot, a new measurement is obtained and the described procedure of Section 3.3 is repeated for a number H of iterations (a method to determine H is described in the sequel) in order to evolve the system.

3.5. Computing graphical representation

The traffic density of each segment can be represented on the map with a line for each road segment. To this end, specific GPS coordinates for each unit have to be maintained. Typically, the density network is depicted by using a color coding for traffic density ranges. Then, the graphical representation of the road network on the map is depicted (see Fig. 2 that is accessible online as https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc5NQ== for the graphical representation of the model running in real-time in some of the cities/areas where it has been applied), and updated at each new value of the traffic sensors, namely every 10 min in this case.

4. Estimating traffic distribution matrixes

In Section 3, the computational model to perform the traffic flow reconstruction has been presented, by assuming the knowledge of junction weights (addressing both those with and without traffic lights).

The values of junction weights should be estimated on the basis of actual traffic behavior in the city since, any approximated values, computed as described in Section 3.1, may be affected by relevant errors. A solution for their computation has been proposed in [37] and [38], where the estimation has been performed by reaching a general error of 25% and using a stochastic method as described hereafter. The aim was to estimate the $TDM(t)$ weights for a given time slot t. The weight assignment contains all the weights w_{ji} at time t of the junctions in the road network. The approach for weight estimation was iterative and started from arbitrary values minimizing the error percentage of traffic density in selected points where such density is known, as they are the sensor locations. If the number of sensors is high, the approach is viable, while the reference sensors should be carefully chosen. In [38], the identified sensors have been inside the related area of interest of the road network. In fact, selecting sensors along the road network border for validation purposes would not be a good

choice since they are subject to boundary conditions, thus making the relaxation of computation process not effective in those cases. Moreover, as discussed in Section 5, error distribution for general Traffic Flow Reconstruction depends on the topological structure of the road network; on such grounds, the selection of reference sensors is very responsive, and not only the internal sensors in the road network could be eligible for error minima.

The valuable part of the solution in [38] included a computational approach based on stochastic relaxation similar to a simulated annealing with random production of values, to reduce the probability of an immediate stop in a local minimum. The procedure identifies a number of road-junction-weights of $TDM(t)$ (as described above or in the whole network) and randomly assigns values to them in a reasonable range depending on road type (residential, tertiary, secondary, primary, trunk and motorway). Then, at each iteration, if the local error is lower than the previous one, then the assigned weights are confirmed. The procedure continues in assigning random values on other road-junction weights until a general error is minimized when no other improvement can be obtained despite a relevant number of new iterations; the procedure converged on reasonable configurations [38]. The ranges assigned to the different road type were not overlapped. The residential road type had a range from 5 to 14, the tertiary road type from 15 to 29, the secondary road type from 30 to 49, the primary road type from 50 to 69, etc. In the present work such ranges admit overlapping in order to extend their cardinalities and effective changes in the road kinds according to their effective usage.

Such procedure can compute the weights for a number of T periods of the day taking into account the actual values of traffic flow sensors in the time slots. We define the *weight assignment* $TDM_b(T)$ as the weight assignment giving the lower mean error in the period T. Then, in real-time prediction we assign $TDM\,(t) = TDM_b(T)$ for each t in T.

Despite the effective results obtained for the estimation of $TDM(t)$, the solution is very computationally heavy, and thus the procedure can be executed only sporadically for the estimation of weights, while reducing as much as possible the number of time slots and cases which $TDM(t)$ weights are computed for. For example, on computer with 20 cores at 2.20 GHz, the estimation for a road network with 31 217 road elements (units Δx) takes about 20 days for the relaxation process of the weight estimation (60 s for the error estimation of only one sensor setting $H = 100$, 50 controlled sensors, 600 weight assignments). On the other hand, the same process has to be performed at least (i) for each time slot of the day, (ii) for working and weekend days, since the traffic behavior is very different. Therefore, multiplying the computational time for 3 cases (working days, Saturday, Sunday) and 48 time slots of the day (every 30 min), about 2880 days should be needed to estimate weights.

For these reasons, the activity has been focused on reducing the execution time by using a parallel solution on GPU. It should be noted that the computation of $TDM(t)$ weights by means of the above described approach based on error verification implies the computation of the solution of Eq. (1) and thus we are presenting a complete parallel solution for solving Equation (1) and $TDM(t)$ weight estimation (see Section 6 for the experimental results and comparisons of the parallel solution in terms of computational time).

4.1. Parallel solution overview

In this section, we introduce an original method to improve the computational performance involving suitable data structures for road network representations. A such innovative parallelization approach is applied to both the calculus of the *density arrays* and the estimation of the $TDM(T)$. Please note that the $TDM(T)$ constitutes an input data for the estimation of *density arrays*, so they are not independent processes. On one hand, the computation of the density arrays seems to be quite straight forward being substantially a finite difference iterative solution of a PDE. On the other hand, the computing of $TDM(T)$ is much more complex since it is based on an iterative stochastic relaxation algorithm.

Table 1
Numerical computation with respect to the position of the units inside a density array.

Units' position	Numerical method
First unit at the sensors' location	Eq. (3) with sensor measurement
First unit at the bound of the graph	Eq. (3) with no sensor measurement
Last unit at the bound of the graph	Eq. (4)
First unit inside the graph	Eq. (6)
Last unit inside the graph	Eq. (5)
Internal units	Eq. (2)

A GPU based parallel architecture has been selected for the execution of solutions such as a GPU NVIDIA Titan XP. It is a SIMD machine that can execute the same instructions on multiple processing units at the same time. The two cases calculus of the *density arrays* and the estimation of the $TDM(T)$ have to be threaded separately.

4.2. Parallel computation of density arrays

To this aim, the data model has to be prepared in a suitable manner, thus allowing the calculus of the density arrays equations simultaneously at each iteration and time stamp, since they are independent each other. In fact, the equations from (2) to (6) have to be applied according to different road network element configuration as summarized in Table 1, where at each equation a label has been assigned; this label has been used to execute different equation in different processing units at the same time. It seems to be like a sort of WHERE construct of many parallel programming languages.

4.3. Data model for TDM computation

The description of each junction configuration represents a static information. The density values involving the traffic distribution admit a fixed placement and they are computed with Eqs. (5) and (6), respectively. Events involving the distribution at the nodes are necessarily independent one another. The junction configuration can be very different one another. Let I and O be maximal number of incoming and outcoming road-segments, then it is possible to create a comprehensive TDM data model maximizing the model and creating V TDM matrices of dimension $O \times I$, where V is the number of junctions in the network. If a junction presents a lower number of in/outs, the non-considered will be zeroed. Analogously, the other objects involving the calculus of the traffic distribution at the junctions can be similarly defined, obtaining V independent computations, called *slides*. The basis of the parallelization approach is grounded on the simultaneously computing of such slides.

4.4. Parallel computing of TDM, junction weights

In the proposed parallel solution, the stochastic approach for computing TDM weights is based on the computation of the model excluding the data from a number of selected sensors at each error computation. The solution can be classified as a stochastic relaxation technique and differs from the one proposed in [38] thanks to parallel implementation and error computing,

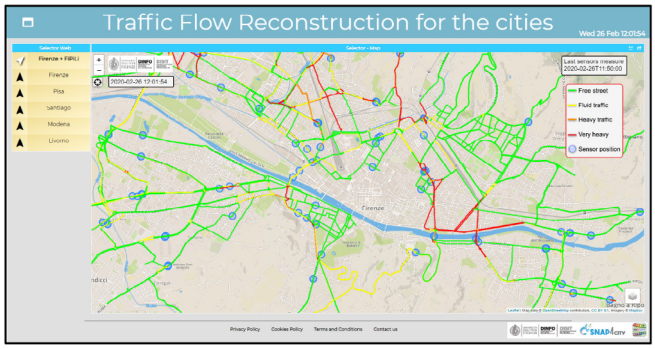

Fig. 2. Graphical representation of the real time traffic flow reconstruction in different cities and areas (on the top left corner the cities/areas selection is provided). Blue points mark the position of the sensors and the state of traffic in each street is represented by different colors. The reconstruction is based on the solution proposed in this paper. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 3. Control Dashboard of the real time conservation of traffic flow in Florence taking into account border sensors. In the top, the first and the second trends represent the hourly behavior of incoming and outcoming vehicle flow, respectively. In the bottom, the monthly trends of the incoming/outcoming behaviors are depicted.

thus producing significantly better results as highlighted in Section 5. More precisely, for each time t the error is computed on the basis of the calculated traffic density $\rho_c(t)$ in those sensor positions with respect to the traffic density value $\rho_M(t)$ measured by the sensors. The formal model of the error is reported in Section 5 where an Error Analysis of the whole solution is carried out.

The computation of the $TDM(T)$ is performed computing a value every 30 min in the day, for the 7 days of the week.

The effectiveness of the proposed computational model is related to the number of iterations H computed by the algorithm. In particular, the number of iterations has to be sufficient to allow the iteration process to minimize errors of estimation in the destination arrays. An insufficient number of iterations may lead to an inaccurate propagation of the traffic flow in the city graph. When the same computation is used for producing the traffic flow reconstruction in real time, the execution of the H iterations has to be completed before the arrival of the next traffic flow measures from sensors.

5. Traffic flow reconstruction error analysis

A deeper analysis of the results to be achieved by the solution we presented can be obtained by assessing the resulted traffic flow reconstruction during the real-time execution in order to understand: (i) identification of the most suitable number of iterations H, (ii) solution accuracy, (iii) if the error in reconstruction depends on structural parameters of the urban network.

The assessment has been performed verifying in real-time the conservation of the flow in the area. As depicted in Fig. 2, sensors are located also at the border and thus change in the conservation can be compensated by any knowledge of the flow rebalancing the difference as in [8]. Fig. 3 reports the real-time dashboard for controlling the conservation of flow. This Dashboard belongs to the Smart City Control Room dashboard for Florence Smart City [35]. Please note also the assessment of sensor network coverage on the right side, bottom corner, accessible to public from https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc2MQ== , the Fig. 3 reports only a part of it.

For this purpose, we instrumented the above-presented process to assess in real-time its precision and its trend. The integrated solution starts from an assigned $TDM(t)$ to perform the traffic flow reconstruction in the road network and verify the RMSE (root-mean-square-error) with respect to actual values in sensor locations. This is performed by computing the solution which excludes data from each different sensor (all of them), so as to estimate the deviation from the calculated traffic density $\rho_c(t)$ in the road where the selected sensor is located, with respect to the density $\rho_M(t)$ measured by the sensor, for each time t. At a given location the RMSE is estimated as $\sqrt{\frac{\Sigma_{t=1}^{T}(\rho_c(t)-\rho_M(t))^2}{T}}$, where T is the total number of observations. The RMSE is used to measure error values when the perfect fit with the data is indicated by 0. Therefore, in the present context 0.5 represents a small error being a fraction of a vehicle in the space of 20 m (it is an absolute value). For each round, the stochastic relaxation produced a minimum of the RMSE that has been taken as a reference together with the produced $TDM(t)$.

The accuracy of the solution primarily depends on the number of iterations H which are applied to the execution. At each iteration the RMSE for each sensor has been measured and also the so-called *system RMSE*, which is the average value of the measured RMSE of all the sensors. Fig. 4 shows the trend of *system RMSE* with respect to number of iterations H. The computing has been based on data observed every 10 min during the weeks from January 20th (Monday) to February 9th (Sunday), i.e., 24 (h) per 21 (days) to calculate the deviation of each computed traffic density with respect to the measured density by sensors.

The System RMSE obtains the minimum error for $H = 250$. In Fig. 4, also the trends of RMSE of internal and external (boundary) sensors (about the 74% and 26% of them, respectively) have been reported. The RMSE of the internal sensor points continues to reduce with the number of iterations, while the RMSE for sensors at the boundaries presents a minimum in the range of H from 150 to 250 depending on the sensors, this effect is due to the boundary conditions and related discontinuities at the external border of the road network. In those cases, the propagation of correct values can arrive only from internal values not having other sensors far away (they are the external sensors in the network). Therefore, a compromise number of $H = 250$ has been taken, as obtained by minimizing System RMSE. Observing the data, the RMSE turned out to be not uniform for all the locations of city sensors, as expected. Thus, other criteria could be also taken, for example the minimization of the errors in specific

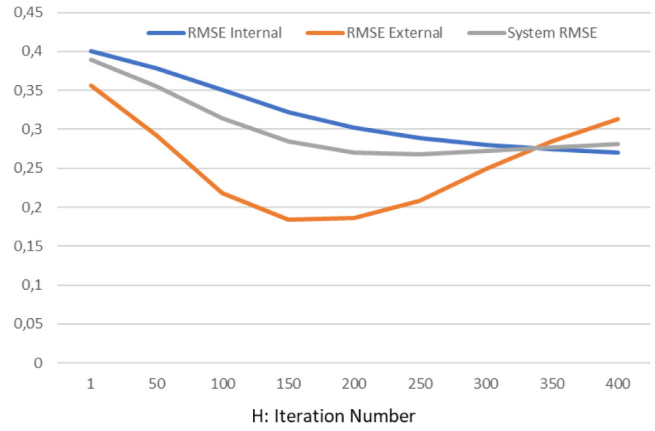

Fig. 4. The RMSE trends with respect to the iterations number H in the traffic flow reconstruction are shown. The average RMSE trend of the internal sensors is represented by the blue line, the average RMSE trend of the external sensors is represented by the orange line. In gray is reported the System RMSE having its minimum when $H = 250$. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

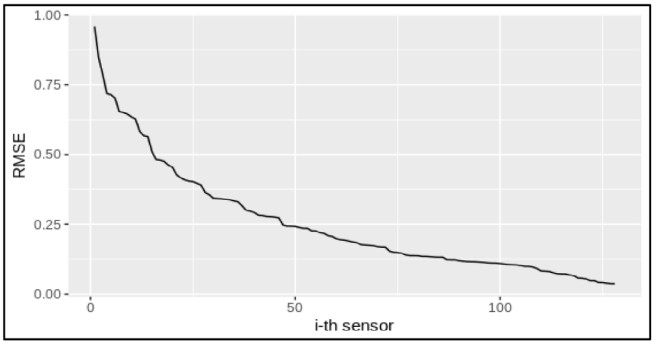

Fig. 5. Distribution of RMSE estimations for each sensor location assuming that the iteration number H is equal to 250.

points of the road network, or even accepting a different number of iterations for different parts of the city. This last solution is particularly complex since would implies to stop the relaxation of the solution, and not always produce better results.

In Fig. 5, the distribution of RMSE in the whole set of traffic sensor road segments is reported for $H = 250$ (that is 130 sensors in Florence area, which has 2730 road-segments and 1390 junctions). Please note that the 90% of the total number of points co-located on actual sensors have an RMSE value less than 0.5 (vehicles per 20 m). In the next part A of this Section, an analytical study on the network properties of the urban graph is conducted in order to give the evidence how the error is related to the topological aspects of the network.

Moreover, the RMSE values are related to traffic conditions and, in general, a greater RMSE value has been registered for high traffic levels in the road network. Such relationship is depicted in Fig. 6 where actual values from the sensors (in terms of number of vehicles per 20 m) have been grouped to make a distribution and to compute their mean RMSE value and the corresponding confidence interval. The graph has been estimated by using 81500 measures performed on sensors during several weeks. The gray band represents the confidence interval of RMSE values. Please note that a larger band has been registered for critical values of traffic density when some congestion situations are occurring. Such trend seems to be related with the data coming from the observed sensors' behavior and their well-known *fundamental diagrams* where congestions are examined.

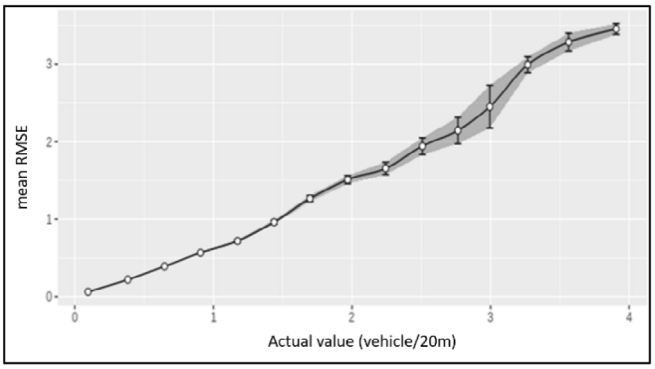

Fig. 6. The relationship of the mean RMSE over the whole road network with respect to actual values obtained by the sensors.

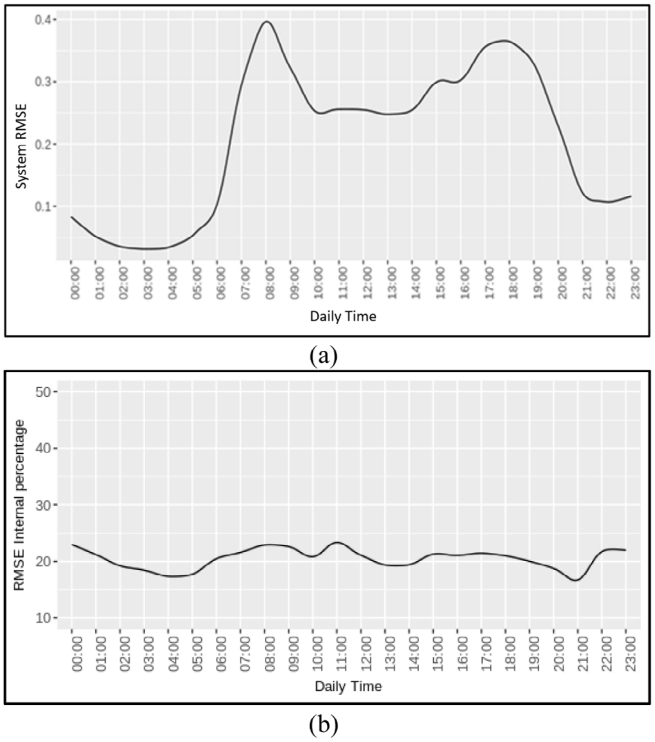

Fig. 7. (a) system RMSE over the daily time, it reflects the behavior in Fig. 3 where the hourly incoming/outcoming vehicle flow is depicted; (b) RMSE Internal percentage (estimated as the RMSE of Internal sensors with respect to the corresponding traffic density over time). This trend is coherent with the linear trend described in Fig. 6.

Since traffic congestion in the city is typically related to the city incoming/outcoming flow according to the working activities of citizens, then also the RMSE value is affected to such behavior in the day. In fact, Fig. 7a shows how the system RMSE depends on daily time, and it reflects the typical behavior in terms of incoming/outcoming flow in the whole city, see Fig. 3. This also means that the traffic flow estimation at system level presents a small error: in most cases lower than 0.2, that is a fraction of a vehicle in the space of 20 m.

Please note that the RMSE is an absolute error measure with respect to the traffic density. As depicted in Fig. 6, the RMSE is higher when the traffic density is high, while the ratio from the RMSE and the traffic density (actual values) is almost constant in fact it can be regarded an approximated straight line. This fact

can be also observed in Fig. 7b where only internal sensors are taken into account.

5.1. Error estimation with respect to the road network topology

This section reports the analysis performed to understand how the error in the traffic flow reconstruction is related to the road network topology. Once a topological relationship is identified, it could be possible to infer that a similar error would be produced in other locations with the same topology and use. This fact can be exploited for a better placement of the sensors and for a general assessment of the data measured and represented in the dashboards and thus in the applications showing the traffic flow reconstruction results.

Each node of the road network may have a specific characteristic in terms of relevance in the graph, regardless if it has been used or not for the location of a sensor. The above presented analysis has shown that the RMSE has a certain non-uniform distribution, see Figs. 4 and 7, and a clear dependency on traffic volume. Therefore, in order to characterize the error function over the city structure and map, many features could be taken into consideration, for example: the type of roads, the structure of the city, the points of attractions, etc. After a long data analysis, we discovered that error behavior is related to the topological characteristics of the road network, as described below.

The traffic flow reconstruction depends on the constraints related to the road network which is modeled as a directed graph. The network analysis tools can help to describe topological features. We tested a large number of them, and in particular two of them turned out to be related with the error behavior of sensors, that is, *betweenness* and *eccentricity* [39].

The *vertex betweenness* (also known as *betweenness centrality*) of a node v is the number of shortest paths which pass through v, formally we have

$$b(v) = \sum_{i \neq j, i \neq v, j \neq v} g_{ivj}/g_{ij}$$

where g_{ij} is the total number of the shortest paths from node i to node j and g_{ivj} is the number of those paths passing-through v. The vertex betweenness represents the degree to which nodes stand between each other and it measures the extent to which a vertex lies on paths between other vertices. Nodes having high betweenness may have considerable influence within a road network by virtue of their control over traffic data passing between others. Such nodes are also the ones whose removal from the network will most disrupt communications between other vertices, because they lie on the largest number of paths inside the network. The betweenness centrality has been demonstrated to be related to the traffic flow in [40] with an R squared of 0.68 and yet such capability has not been exploited in relationship with the error of traffic flow reconstruction. The correctness of a given information which goes through the nodes of a network is also related to other issues related to node properties. An example is given by the *eccentricity* of a vertex v, denoted by $e(v)$, which is defined as the shortest path distance a given vertex has from the farthest other node in the graph.

In order to understand the error behavior in the road network, the values of betweenness and eccentricity have been computed for each node of the urban graph. On the other hand, if the topological indexes are computed on the whole road city graph, also restricted traffic zones and marginally used roads should be taken into account. Therefore, those indexes would not describe the actual traffic network of the city. Thus, we only considered the nodes and arcs involved in the traffic flow reconstruction. Fig. 8 depicts the position of the computed highest values for

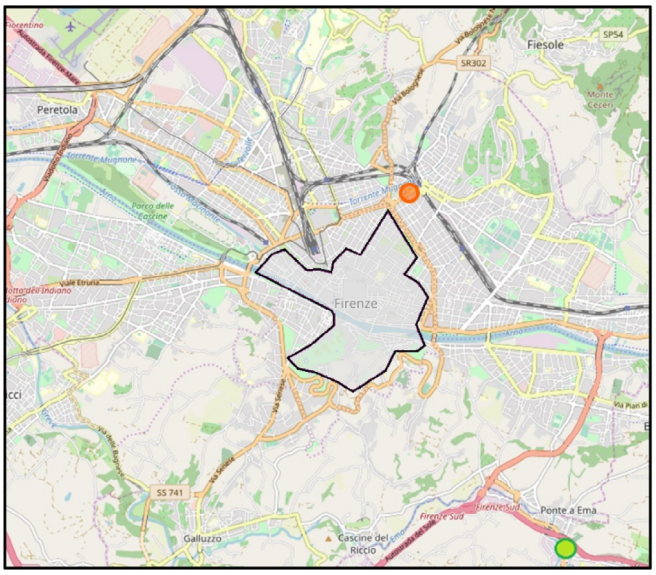

Fig. 8. In orange the node having the maximum betweenness value, while in green the node having the maximum eccentricity value. The main restricted traffic zone is depicted in the center of the city in gray. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Multilinear regression results.

Coefficient	Estimate		Std. Error	t-value	p-value
betweenness	β	0.80224	0.13097	6.125	<0.05
eccentricity	γ	0.23256	0.02657	8.752	<0.05

Residual standard error: 0.1806.
Multiple R-squared: 0.6836.
Adjusted R-squared: 0.6786.
F-statistic: 136.1.
p-value: $<$ 2.2e−16.

betweenness and *eccentricity* metrics, in orange and green, respectively. Please note that *betweenness* is located in proximity of one of the typical areas where traffic congestion often occurs. On the other hand, nodes having high eccentricity are located in the decentralized zones of the urban graph admitting more distance from the other side of the network.

Thus, a multilinear regression model has been conceived to verify the presence of an effective relationship between the RMSE and metrics reported in Table 2, and to identify a model that can put in relationship the identified feature with the error function as RMSE. The parameters of the resulting model have been reported in Table 2.

The identified model is $Y_i = \beta x_i + \gamma z_i$ where Y_i, x_i, z_i are the RMSE, betweenness and eccentricity, respectively. The β coefficient is the expected change in Y associated with a 1-unit increase in the value of x at the same value of z; the γ coefficient is the expected change in Y associated with a 1-unit increase in the value of z at the same value of x. The results show that both betweenness and eccentricity are statistically significant (with a p-value < 0.05). The R squared of the model is equal to 0.68, which means that the model explains 68% of the variability of the response data around its mean. When supposing to consider the betweenness as the unique metric of the model, then the R squared decreases to 0.48, so that the multilinear regression model is improved by means of the eccentricity values. The result presented in this section is strongly different from the one in [40] where only the relation between traffic flow and betweenness centrality is considered. Therefore, we have identified a relevant

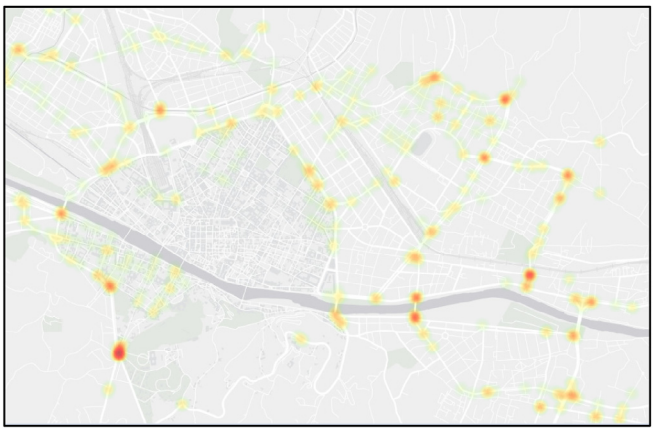

Fig. 9. A graphical representation of the error distribution is presented. An intense pigmentation denotes a higher error of the distribution of Fig. 5. The typical road segments with high traffic flow are represented in Fig. 2 with a snapshot of the actual service.

relationship between the RMSE and network topology. A general representation of the Y over the considered urban map is depicted in Fig. 9, where locations having an intense pigmentation are affected by a greater error model. It is easy to remark that urban zones having greater errors are not those where traffic is heavy. The typical road segments with high traffic flow are represented in Fig. 2 with a snapshot of the actual service. Thus, comparing Figs. 2 and 9, it is evident that the critical values of the error function are not specifically located on segments with high traffic, and neither on all cross points or close to the sensors. They are more related to the critical topological points of network, which are specific nodal cross points.

6. Experimental results

The computational speed of the solution depends on the dimension of the considered road network. In particular, as the computational complexity it is an $O(H(V+U))$ where: V is the number of nodes, U is the number of units and H is the number of iterations. Since U is larger than V, then we definitively have an $O(HU)$.

Concerning the validation of our model, we have considered an area of the metropolitan network of Florence constituted by 1390 nodes (or intersections), 2730 road-segments and 130 traffic sensors, as depicted in Fig. 2. The graph array structure has U = 31217 elements which represent the total amount of units occurring in the network, that is, the total number of units in density arrays.

For example, we can observe the results in Table 3 showing some execution times (in seconds) for different number of iterations related to the mentioned approaches. Each execution time has been obtained as mean value of results taken on 30 distinct executions in both cases: (CPU) 20 cores at 2.20 GHz, 128 GB Ram; (GPU) Graphics device GPU NVIDIA of 12 GB.

Of course, the use of such data structures requires a supplemental time for their wrapping/unwrapping which does not alter the above dissertation.

An additional analysis related to the mentioned approaches has been also conducted. So that, fixing the number of iterations needed for the model computation we have compared the computational speed with CPU and GPU via parallel structures. Some results concerning this topic can be sketched in Table 4 where execution times are presented adopting 1, 10, 20, 30 sensor nodes where the related errors are estimated, respectively, via parallel structures.

Table 3
Algorithm execution times (in seconds) according to different number of iterations. Such comparison is performed on a machine having 20 cores at 2.20 GHz 20, 128 GB Ram, Graphics device GPU NVIDIA Titan XP.

Execution time	# of iterations			
	10	100	200	300
CPU (in sec)	4.85	61.48	150.62	366.02
GPU (in sec)	0.63	3.33	6.46	9.56

Table 4
Algorithms' execution times (in seconds) using GPUs according to different sensors' numbers. A such comparison is performed on a machine having 20 cores at 2.20 GHz 20, 128 GB Ram, Graphics device GPU NVIDIA Titan XP.

Execution time	# of sensors			
	1	10	20	30
Using CPUs (in sec)	3.23	29.11	60.57	100.82
Using GPUs (in sec)	7.8	35.55	72.95	87.61

Due to this analysis, whenever a small dimension data structure is ingested in the GPUs architectures, then the computational speed does not improve; thus, the desired advantages are related to data structure having greater dimensions. Such behavior is a common trend in the context of parallel computing approach via GPUs where a breaking event appears (in our case, it is closed to a data structure having about 780 000 units). For our purpose, such breaking event validates that the computation of the graph array (having 31 217 units) in real-time applications is carried out via parallel approach using CPUs, while the computation of the analysis error (having 4 058 210 units for each time observation) is performed by GPUs architectures in order to improve the computational speed of the presented model.

Considering for example $H = 250$, the computation of the error analysis presented in Section 5 takes about few minutes via GPU parallel computation for each observed time. Since the data under observation for this analysis take into account thousands of samples, then the execution time needed for this analysis would require some years by using a sequential approach on CPU.

7. Conclusions

The real time status of road network in terms of traffic flow reconstruction is receiving a higher attention due to its relevance for a number of smart services such as: routing, planning, dynamic tuning services, healthy walk, etc. The precise traffic flow status may become fundamental to save life for rescue teams, police departments, fire brigades, and ambulances. Both costs and needed precision have also constrained local governments to exploit sensor networks, which are in any case limited by the new number of sensors that can be acquired and maintained. The solution presented in this paper is based on data coming from scattered traffic flow sensors, which can be placed at cross junctions, as well as in the road segments. It has turned out to produce traffic flow reconstructions with high precision, addressing at the same time (i) the traffic flow reconstruction problem by solving PDE; (ii) the estimation of traffic flow distribution at junctions with an approach that can be classified as a stochastic relaxation. In the paper, and with a real time accessible service, it has been also demonstrated that results can be obtained in real time for large road networks exploiting tensor flow modeling and GPU. Moreover, we have also characterized the error function for the traffic flow reconstruction, thus proving its relationships with road network features such as betweenness and eccentricity. The spatial distribution of errors has highlighted that greater error values are far from the most critical area where higher values of traffic are registered. The main exploitable results of the solution

proposed are mainly related to the access at a dense traffic flow reconstruction. It can be used at the basis of many routing and multimodal routing algorithm for final users, and for the operators (taxi, fire brigade, police, ambulance, etc.). In addition, a recent application exploited the traffic flow for predicting the distribution of pollutant in the city. In turn, the prediction of pollutant can be used to plan short walking with baby, training, and biking in the city.

However, the presented model assumes that the suitable data structure for the related urban network is unchanging, that is, a given road is also considered in the computation when the actually access is temporarily not permitted (for example, in case of road maintenance) and the related percentage of turn is null. A further research could investigate on a sort of dynamical road network structure for the model in order to improve its computational cost. Another line of research could take into account the integration of some additional data (if available) in the model, as for example: traffic accidents, traffic light waiting times, etc.

The results presented in the paper has been obtained in the context of Sii-Mobility national project on smart city transport systems in Italy, a very large research project, and it is at present exploited in a number of cities/regions across Europe (Firenze, Pisa, Livorno, Modena, Santiago de Compostela) and is at the basis of research projects (such as RESOLUTE, Snap4City, TRAFAIR of the European Commission).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the MIUR, the University of Florence and companies involved in the co-funding of Sii-Mobility national project on smart city mobility and transport. Km4City is an open technology and research of DISIT Lab. Sii-Mobility is grounded on and has contributed to Km4City open solution. Moreover, we gratefully acknowledge the support of NVIDIA Corporation which gave for free the Titan XP GPU used in the first trial of this research.

References

[1] B.M. Williams, A.H. Lester, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, J. Transp. Eng. 129 (6) (2003) 664–672.
[2] Y. Kamarianakis, P. Poulicos, Space–time modeling of traffic flow, Comput. Geosci. 31 (2) (2005) 119–133.
[3] W. Zheng, D. Lee, Q. Shi, Short-term freeway traffic flow prediction: Bayesian combined neural network approach, J. Transp. Eng. 132 (2) (2006) 114–121.
[4] W. Huang, et al., Deep architecture for traffic flow prediction: deep belief networks with multitask learning, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201.
[5] Y. Lv, et al., Traffic flow prediction with big data: a deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 865–873.
[6] L. Po, et al., TRAFAIR: Understanding traffic flow to improve air quality, in: IEEE International Smart Cities Conference, 2019, pp. 730–737.
[7] S. Hu, et al., Smartroad: smartphone-based crowd sensing for traffic regulator detection and identification, ACM Trans. Sensor Netw. 11 (4) (2015) 55.
[8] J.C. Herrera, Alexandre M. Bayen, Traffic flow reconstruction using mobile sensors and loop detector data, 2007.
[9] D.B. Work, et al., An ensemble kalman filtering approach to highway traffic estimation using GPS enabled mobile devices, in: 2008 47th IEEE Conference on Decision and Control, IEEE, 2008.
[10] X. Ma, et al., Large-scale transportation network congestion evolution prediction using deep learning theory, PLoS One 10 (3) (2015) e0119044.
[11] A. Abadi, T. Rajabioun, P.A. Ioannou, Traffic-flow prediction for road transportation networks with limited traffic data, IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 653–662.
[12] E. Lovisari, Carlos Canudas de Wit, Alain Y. Kibangou, Data fusion algorithms for density reconstruction in road transportation networks, in: 2015 54th IEEE Conference on Decision and Control (CDC), IEEE, 2015.
[13] S. Yang, S. Shi, X. Hu, M. Wang, Spatiotemporal context awareness for urban traffic modelling and prediction: Sparse representation based variable selection, PLoS One 10 (10) (2015) e0141223.
[14] T.Q. Tang, W.S. Shi, X.B. Yang, Y.P. Wang, G.Q. Lu, A macro traffic flow model accounting for road capacity and reliability analysis, Phys. A 392 (2013) 6300–6306.
[15] Martin Treiber, Dirk Helbing, Reconstructing the spatio-temporal traffic dynamics from stationary detector data, Coop. Transp. Dyn. 1 (3) (2002) 3.1–3.24.
[16] M.J. Lighthill, G.B. Whitham, On kinematic waves II. A theory of traffic flow on long crowded roads, Proc. R. Soc. A 229 (1178) (1955) 317–345.
[17] P.I. Richards, Shock waves on the highway, Oper. Res. 4 (1956) 42–51.
[18] G. Bretti, R. Natalini, B. Piccoli, A fluid-dynamic traffic model on road networks, Arch. Comput. Methods Eng. 14 (2007) 139–172.
[19] S.K. Godunov, A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Math. Sb. 47 (1959) 271–290.
[20] P. Kachroo, S. Sastry, Travel time dynamics for intelligent transportation systems: theory and application, IEEE Trans. Intell. Transp. Syst. 17 (2016) 385–394.
[21] S. Blandin, D. Work, P. Goatin, B. Piccoli, A. Bayen, A general phase transition modelfor vehicular traffic, SIAM J. Appl. Math. 71 (2011) 107–127.
[22] G.M. Coclite, M. Garavello, B. Piccoli, Traffic flow on a road network, SIAM J. Math. Anal. 36 (2005) 1862–1886.
[23] C.F. Daganzo, The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory, Transp. Res. B 28 (4) (1994) 269–287.
[24] M. Van den Berg, A. Hegyi, B. De Schutter, J. Hellendoorn, Integrated traffic control for mixed urban and freeway networks: A model predictive control approach, Eur. J. Transp. Infrastruct. Res. 7 (3) (2007) 223–250.
[25] L.C. Zammit, S.G. Fabri, K. Scerri, Simultaneous traffic flow and macro model estimation for signalized junctions with multiple input lanes, in: Proc. 3rd Int. Conf. Vehicle Technol. Intell. Transp. Syst. Lisbon, Portugal, 2017, pp. 157–164.
[26] K. Aboudolas, M. Papageorgiou, E. Kosmatopoulos, Store-and forward based methods for the signal control problem in large-scale congested urban road networks, Transp. Res. C 17 (2) (2009) 163–174.
[27] L.C. Zammit, S.G. Fabri, K. Scerri, Real-time parametric modeling and estimation of urban traffic junctions, IEEE Trans. Intell. Transp. Syst. 20 (12) (2019) 4579–4589.
[28] A.M. Bruaset, A. Tveito, Numerical Solution on Partial Differential Equations on Parallel Computers, in: Lecture Notes in Computational Science and Engineering, Springer Science & Business Media, 2006.
[29] A.T. Chronopoulos, G. Wang, Parallel solution of a traffic flow simulation problem, Parallel Comput. 22 (1997) 1965–1983.
[30] D. Das, R. Misra, Parallel processing concept based on road traffic model, Proc. Technol. 4 (2012) 267–271.
[31] H. Yi, H. Jung, S. Bae, Deep neural networks for traffic flow prediction, in: Proceedings of 2017 IEEE International Conference on Big Data and SmartComputing (BigComp), 2017, pp. 328–331.
[32] Q. Kong, Y. Xu, S. Lin, D. Wen, F. Zhu, Y. Liu, UTN-model-based traffic flow prediction for parallel-transportation management systems, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1541–1547.
[33] P. Bellini, M. Benigni, R. Billero, P. Nesi, N. Rauch, Km4city ontology building vs data harvesting and cleaning for smart-city services, J. Vis. Lang. Comput. 25 (6) (2014) 827–839.
[34] P. Bellini, S. Bilotta, D. Cenni, P. Nesi, M. Paolucci, Mirco Soderi, Knowledge modelling and management for mobility and transportation applications, in: Proceedings of 1st International Workshop on Technology Convergence in Smart Cities, 2018, pp. 1–8.
[35] P. Bellini, D. Cenni, M. Marazzini, N. Mitolo, P. Nesi, M. Paolucci, Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support, J. Vis. Lang. Comput. http://dx.doi.org/10.18293/VLSS2018-030.
[36] E. Bellini, L. Coconea, P. Nesi, A functional resonance analysis method driven resilience quantification for socio-technical system, IEEE Syst. J. http://dx.doi.org/10.1109/JSYST.2019.2905713.
[37] P. Bellini, S. Bilotta, P. Nesi, M. Paolucci, M. Soderi, WiP: Traffic flow reconstruction from scattered data, in: Proceedings of 2018 IEEE International Conference on Smart Computing (SMARTCOMP), 2018, pp. 264–266.
[38] P. Bellini, S. Bilotta, P. Nesi, M. Paolucci, M. Soderi, Real-time traffic estimation of unmonitored roads, in: Proceedings of 2018 IEEE 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology, 2018, pp. 935–942.

[39] Mv. Steen, Graph Theory and Complex Networks: An Introduction, first ed., Maarten van Steen, 2010.
[40] Elise Henry, et al., Spatio-temporal correlations of betweenness centrality and traffic metrics, in: 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE, 2019.

Stefano Bilotta was born in Arezzo, Italy, in 1983. He received the Master degree in mathematics from the University of Siena, Italy, in 2009 and the Ph.D. degree in computer engineering and automation from the University of Florence, Italy, in 2013. From 2013 to 2016, he was a postdoctoral researcher at the Department of mathematics and computer science of the University of Florence. Since 2017, he has been a postdoctoral researcher of the DISIT Lab at the Department of information engineering of the University of Florence. His research interests include traffic flow reconstruction algorithms, parallel solution, dynamic systems, languages and coding theory. He has been involved in projects such as: Sii-Mobility and Trafair.

Paolo Nesi is a full professor at the University of Florence, Department of Information Engineering, chief of the Distributed Systems and Internet Technology, DISIT lab, and research group. His research interests include massive parallel and distributed systems, physical models, IOT, mobility, big data analytic, semantic computing, formal model, machine learning. He has been the general Chair of IEEE SC2, IEEE ICSM, IEEE ICECCS, DMS, etc., international conferences and program chair of several others. He is and has been the coordinator of several R&D multipartner international R&D projects of the European Commission such as Snap4City, RESOLUTE, ECLAP, AXMEDIS, WEDELMUSIC, MUSICNETWORK, MOODS and he has been involved in many other large projects on mobility and transport, smart city, such as Km4City, Sii-Mobility, Herit-Data, MobiMart, Trafair, Mosaic, REPLICATE, Weee, etc.