



# Road Network Graph Representation for Traffic Analysis and Routing

Chiara Bachechi<sup>(✉)</sup>  and Laura Po 

“Enzo Ferrari” Engineering Department, University of Modena and Reggio Emilia,  
Modena, Italy  
{chiara.bachechi,laura.po}@unimore.it

**Abstract.** The road network is the infrastructure along which the mobility of users and goods takes place; the analysis of these networks in terms of spatial and graph theoretical approaches can provide insights to understand urban mobility, improve daily commuting, and reflect on new, more sustainable, scenarios. This paper presents an open-source framework to analyze the road network and investigate the relationship between its topology and traffic conditions. Open-source geographical data are stored in a graph database containing roads, junctions, and Points of Interest (POI), allowing importing of traffic data. The framework includes routing algorithms to obtain the optimal path based on different aspects such as distance, traffic volume, and the number of traversed junctions; furthermore, it allows simulating road closures to observe how they affect road viability. The framework was tested in the use case of the city of Modena (Italy) providing promising results.

**Keywords:** Road network · Graph database · Routing · Road traffic · Smart city · Network modelling

## 1 Introduction

At least once in a lifetime, everyone has been stuck in traffic. In recent decades, researchers have studied the topology and characteristics of the road network to understand, plan and optimize traffic management, improve commuting time and reduce emissions. In particular, when it comes to road networks that are complex non-planar spatial networks, weighted multi-digraphs play a massive role in the analysis and understanding of their properties. Two main approaches to road traffic modeling are suggested in the literature: a primal approach described in [26] and a dual approach presented and exploited in [14, 25]. The primal approach generates a graph where each road intersection (junction) is a node and lanes or road sections connect the junctions with relationships: a *primal graph*. The primal graph is necessary for routing purposes and provides an in-depth representation of the road network. The dual approach instead is inverse: it considers roads as nodes, and the connections between them are created when they meet at a junction. The obtained graph is a *dual graph*. Most studies adopt the primal

approach because, in the dual graph, the information regarding the geo-location and the geometry of the roads may be lost. However, both representations can be useful to unveil different aspects of the road network topology: the dual graph can help to study the connections between roads and easily determine the ones where traffic flows converge. For this reason, we realized an open-source framework available on git repository<sup>1</sup> that allows directly generating a graph instance from Open Street Map (OSM)<sup>2</sup> data composed of two graph representations: a primal and a dual graph. The two representations are generated together and added to the same graph database, keeping a connection between the node representation of the road in the *dual graph* and its geometry memorized in the *primal graph*. This combined approach allows exploiting the advantages of a simplified representation of the relations between roads without losing the complexity of their geometry construction.

The obtained graph model also includes the Points Of Interest (POI) in the geographical area and supports the integration of mobility information, such as traffic volume data. Moreover, the graph can be changed by closing or opening roads to investigate alternative routes in the different road closure scenarios. The analysis of the road network can be performed by applying centrality and community detection algorithms. This paper highlights the importance of integrating traffic information directly into the graph and taking into account the distance, the volume of traffic and the number of nodes crossed when evaluating the shortest path. For this reason, three different routing methods have been compared. Since the framework imports data from OpenStreetMap, it enables the easy generation of a digital twin of the road network of any city, area, or region; also, routing can be performed in very few steps. We strongly believe that this solution will simplify the modeling, analysis, and improvement of road networks in different cities. We present a use case analysis conducted on the road network of the city of Modena (Italy) that shows how traffic information can be generated, integrated, and exploited to gain additional insight into the traffic conditions of the urban area.

The rest of the paper is organized as follows. Related work is presented in Sect. 2. The generation of the graph model is described in Sect. 3. Section 4 explains how traffic data can be generated; while, Sect. 5 describes the methodology adopted to integrate traffic data in the existing road graph model. Furthermore, Sect. 6 presents three routing approaches and the management of road closures. In Sect. 7, the results of the analysis performed in the city of Modena are presented. Finally, conclusions are discussed in Sect. 8.

## 2 Related Work

There are many examples of the profitable use of graphs for representing road networks [1, 13, 20, 29]. In [21], several modelling solutions that can be employed to represent and simplify a road network are presented. A data-driven graph

<sup>1</sup> <https://github.com/ChiaraBachechi/roadRouting/>.

<sup>2</sup> <https://wiki.openstreetmap.org/>.

model generated from traffic sensors observations based on the primal approach is described in [16]. They model the road network as an undirected graph weighted according to the spatial correlation among adjacent traffic intersections; however, to adopt this solution, traffic sensors are needed at every road intersection. In [15], a tool that allows studying efficiency in a road network is described. They employed a PostgreSQL database with pgRouting<sup>3</sup> and PostGIS extension to convert OSM ways into geometric features. Graph algorithms, however, are generally performing better when executed on a graph database as discussed in [7, 22]. For this reason, we decide to employ the Neo4j graph data platform<sup>4</sup> for our framework. In [5], a library to convert the OSM road network into a road graph is presented. This tool allows downloading and analyzing the road network as a graph. We rely on this library for the generation of the initial *primal graph*, but we enriched the obtained graph with properties and relations extracted from traffic data, POI nodes and their connections, additional properties related to the status of the street, and we generate a *dual graph*. Moreover, our developed framework allows closing and opening streets and generating different routes according to the new road status. Several studies discussed the use of centrality measures to explain traffic flows in urban areas [11, 12] investigating the correlation among different centrality measures and the corresponding simulated or real traffic flows. The studies are mainly focused on the primal approach, we further investigate this correlation in this paper considering also the *dual graph*. Pathfinding algorithms have been widely studied to determine the shortest path between a source and a destination. In [30], the labels associated with each road are taken into account to find label-constrained paths, employing index-based techniques and decomposing the road network into a tree-like structure. Moreover, in [27], the authors suggest that the users are looking for the fastest path rather than the shortest path; thus, they include traffic influence factors when evaluating the shortest path and develop ‘Trafforithm’ a traffic-aware shortest path algorithm.

### 3 Road Network Graph Modelling

To ensure that this methodology can be easily applied to any urban environment, OSM is the main open data source. OSM provides very high-resolution data regarding road networks collected through the collaboration of a big community of users worldwide. The topological data structure has two main elements: nodes and ways. Nodes represent map features without a size that can be approximated as points. While, ways are lists of nodes representing polylines and polygons. The more intuitive conversion of this structure in a graph model is to generate graph nodes from OSM nodes and relationships from ways that connect them. As a result, a *primal graph* is obtained. This graph can be used for routing purpose since each node maintains a reference to the real point in space where the junction is located. However, the number of nodes and relationships is very high, while the density is very low. A simplified representation of the road network

<sup>3</sup> <https://pgrouting.org/>.

<sup>4</sup> <https://neo4j.com/>.

can be generated by the dual approach where the graph nodes are roads (each identified by their OSM identifier). The resulting *dual graph* is a modified version of the named street graph (described in [8]) that assumes as street names their OSM identifiers. The *dual graph*'s analysis highlights the interactions between roads. These two representations of the road network are regenerated in the same graph database instance maintaining a connection between them to exploit the advantages of both approaches.

### 3.1 Primal Graph

The *primal graph* represent point-to-point relationships between junctions. The OSMnx Python package [5] was employed to construct the graph database instance directly from OSM data. Given a point and a radius, the data in the circular area are converted into a 'graphml' format; then, a Neo4j instance is generated through a query in Neo4j proprietary language: Cypher. To do that we employed the Awesome Procedure On Cypher (APOC)<sup>5</sup> library. Since our framework is devoted to vehicular traffic, we decide to consider only the roads where vehicles can travel. We modify the structure of the graph obtained directly with OSMnx: we changed the label of OSM nodes to 'Node', and we insert a spatial property containing the node's location (latitude and longitude). Moreover, relationships between the junction nodes are called 'ROUTE' and contain information about the type of highway, the street name, the OSM identifier, the length of the road segment, and the status. The status property has also been added and is 'active' when the road is open to vehicular traffic, or 'closed' if not. A relationship is generated for each travel direction.

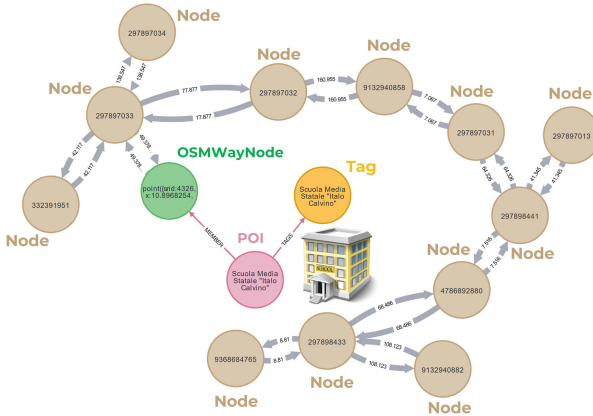
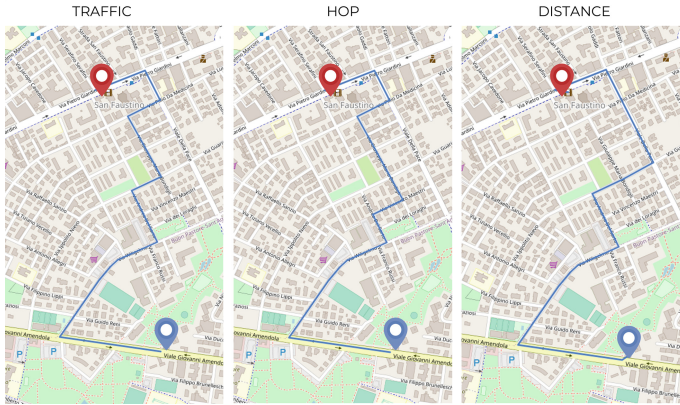


Fig. 1. An example of the primal graph structure.

For routing purposes, also POIs are of key importance as sources and targets of vehicles' routes. For this reason, the Overpass API<sup>6</sup> is queried to get data

<sup>5</sup> <https://github.com/neo4j-contrib/neo4j-apoc-procedures>.

<sup>6</sup> <http://overpass-api.de/>.



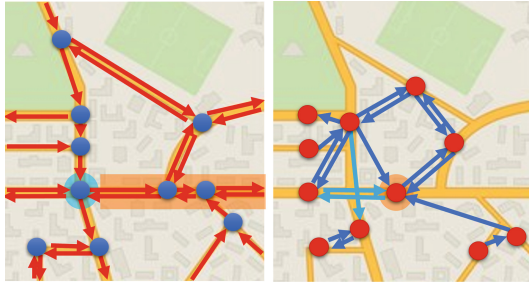
**Fig. 2.** Comparison between the paths evaluated with the three approaches.

regarding amenities (e.g. restaurants, bars, pubs, and schools) and insert POI in the already existing graph as new nodes. The OSM’s flexible structure enables the association of a variable number and different type of tags to each element (way or node) that describes its properties. Thus, the POI nodes are labelled as ‘PointOfInterest’ with a connected ‘Tag’ node containing all the additional properties of the POI retrieved from OSM. Each POI is connected to a node labelled as ‘OSMWayNode’ with a ‘MEMBER’ relationship and then connected to the node of the road network nearest to the exact geographic position of the POI. When the POI is represented as a point (OSM node), the ‘OSMWayNode’ is a single node. While, when the POI has a complex geometry (e.g. polygon) and is represented as a way in OSM, it is represented as a collection of connected ‘OSMWayNodes’. Figure 1 shows an example of a POI node connected to: a Tag node with additional information, and an OSMWayNode connected to the other nodes of the road network. The Gray relationships are of ‘ROUTE’ type and the value displayed is the distance in meters. Exploiting the framework for our use case, we generate a *primal graph* for the city of Modena containing 19,607 junction nodes, 841 POIs nodes, and 33,571 ‘ROUTE’ relationships. The average un-directed degree of nodes is 3.41: the incoming and outgoing average degree are very similar and around 1.71. The graph has a very low density as expected for a road network ( $8 \times 10^{-5}$ ).

### 3.2 Dual Graph

The *primal graph* is a point-to-point oriented graph where the relationship that connects the nodes are segments of road lanes and to represent a single road you may need more than one relationship. For example, in Fig. 3, all the nodes and relationships highlighted in orange in the *primal graph* correspond to the same road. For this reason, we decide to generate a simplified version of the graph that is derived from the *primal graph*, and thus we will refer to it as

*dual graph*. In this graph, each road is represented by a node ‘RoadOsm’, and the relationships ‘CONNECTED’ are generated between connected roads. Two roads are connected if there is a junction that allows driving from the source road to the target road. Thus, each ‘CONNECTED’ relationship is associated with a junction ‘Node’ in the *primal graph*. However, for each node in the *primal graph* there could be several ‘CONNECTED’ relationships in the *dual graph*. This version of the graph is distant from the real street map since the nodes do not have a spatial reference. For example, in Fig. 3, the node highlighted in light blue in the *primal graph* corresponds to the three relationships in the same color in the *dual graph*. This alternative representation can be useful to better understand the relationship between roads and identify the ones that are more important in the road network. Our framework automatically generate this simplified *dual graph* as an additional layer over the *primal graph*. Moreover, the graph contains only the roads in the ‘active’ status: open to vehicular circulation. In this way, the user can generate alternative *dual graphs* in different scenarios of road closures and study the different relationships’ contexts that will emerge. The *dual graph* generated for the city of Modena contains 4421 nodes and 15037 relationships, the average undirected degree of nodes is 6.8 (the incoming and the outgoing average degrees are very similar and around 3.4), and a decimal order higher density compared with the *primal graph* ( $7 \times 10^{-4}$ ).



**Fig. 3.** Example of a *primal graph* (on the left) and the *dual graph* (on the right). (Color figure online)

## 4 Generation of Traffic Data

Traffic information can be generated in many ways from traffic sensors observations (e.g. induction loop detectors, cameras, Bluetooth sensors), GPS routes, and open data sources (e.g. Open Transport data<sup>7</sup>), OD matrices or through simulations. Each method can provide different data for each road lane such as vehicle count, speed, type of vehicles, traffic flow (Veh/hour), etc. We are interested in integrating the annual average daily traffic volume (AADT) for each road lane in the road network. AADT is a traffic volume metric defined as the

<sup>7</sup> <http://opentransportmap.info/>.

average daily traffic volume at a given road lane over a full 365 days/year [10]. In the following, we described the solution adopted to generate traffic data and calculate AADT in our use case.

About 400 induction loop detectors are placed under the surface of the street in the urban area of the city of Modena; these sensors provide a value of vehicle count and average speed every minute. Traffic flow data have been collected from November 2018 to April 2021, cleaned (as described in [4]), and used to feed a micro-simulation traffic model: SUMO (Simulation for Urban MObility) [17]. The traffic sensor data of each day are the input of the SUMO model that simulates the traffic in all the main streets of the urban area. This simulation allows predicting the traffic flow in all the road segments of the road network, and it is better described in [2,3,24]. Considering all the simulations of the year 2019, for each road  $r$ , AADT was evaluated as:

$$AADT_r = \frac{\sum_{i=0}^N \sum_{j=0}^{24} flow_{r,j,i}}{N}$$

where  $N$  is the number of simulated days in the given year, and  $flow_{r,j,i}$  is the observed traffic flow (veh/hour) in the road lane  $r$  for the  $j^{th}$  hour of the  $i^{th}$  day of the year. The traffic data generated by the traffic model for the city of Modena are visualized in a dashboard<sup>8</sup> and available as open data<sup>9</sup>. Moreover, the CSV formatted file used in our use case is available for testing in the git repository.

## 5 Traffic Data Integration

The framework allows integrating traffic data directly in the graph, through a python script. This traffic data need to be formatted as a CSV file containing the OSM id of the starting OSM node, and the ending OSM node between which the traffic is measured. A new relationship named ‘AADT’ is inserted in the *primal graph* between the source and the target node; then, the traffic volume is added as a property of this new relation. A new relationship is needed because the *primal graph* and the road network the traffic data refers to can be different. In particular, in our use case, traffic data are lane-based; thus, in the same direction there could be more lanes and the *primal graph* has a single ROUTE relation. Additionally, a new property is added to the ‘ROUTE’ relation: the average traffic volume in each direction. Moreover, traffic data may not be complete: they can cover only a reduced part of the total roads. Where traffic data are not provided, in order to exploit as much as possible the available traffic information, two main approximations are adopted:

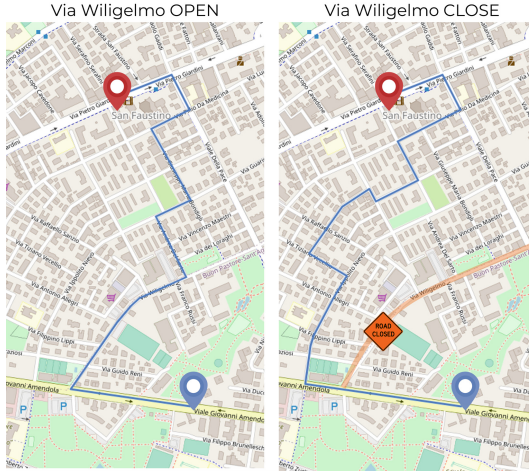
- the traffic volume is evaluated as the average traffic volume of all the observed relationships in the same direction between 1 to 5-order neighbors,

<sup>8</sup> <https://trafair.eu/trafficflow/annual-average-traffic-volume>.

<sup>9</sup> [https://dati.emilia-romagna.it/dataset?organization=comune-di-modena&tags=features&res\\_format=WFS&res\\_format\\_limit=0&page=2](https://dati.emilia-romagna.it/dataset?organization=comune-di-modena&tags=features&res_format=WFS&res_format_limit=0&page=2).

- if there are no neighboring traffic relationships, the traffic volume is evaluated as the average AADT of all the roads of the same type (e.g. highway, primary, secondary, and residential).

Moreover, in the *dual graph*, a new ‘traffic’ property is added to each road node. Since longer roads are supposed to have a higher traffic volume, the traffic property is evaluated as the ratio between the average AADT and the total distance of all the ROUTE relationships that correspond to the given road.

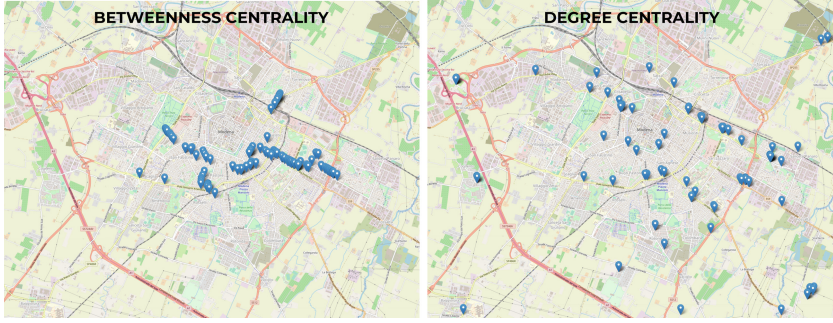


**Fig. 4.** Comparison between the shortest paths before and after via Wiligelmo closure.

## 6 Routing

Detecting the optimal route between two points in a network based on different traffic conditions is of key importance for traffic management and analysis. Moreover, routing algorithms can help in generating realistic random routes to feed simulation models. Pathfinding algorithms find the best path between two nodes in a graph, comparing all the possible paths based on their cost. The cost can be evaluated in very different ways: summing the values of a property of the relation that associates the two nodes (used as weight), or counting the number of traversed nodes in the path. We explored three main approaches for shortest path evaluation based on: the number of traversed nodes, the distance, and the traffic volume. The shortest path based on the number of traversed nodes was evaluated with the fast algorithm provided by Neo4j. This solution is unweighted and only needs to search for the path; thus, the implementation is based on the fast bidirectional breadth-first search algorithm [23]. However, when considering the ‘distance’ attribute that corresponds to the length of the road segment that connects the two nodes, we need to employ a more complex algorithm: the A\*





**Fig. 5.** Map of the 100 junctions with the highest betweenness centrality (on the left) and the highest degree centrality (on the right) in the city of Modena.

informed search algorithm that uses a heuristic function. This heuristic function is the Haversine distance between two geo-located points on the earth sphere [18]. For this reason, each node contains information regarding its position (the coordinates). Moreover, the distance between the nodes in a road network can be significantly different from the real distance to travel; thus, the A\* algorithm use as weight the length of the road segment. Finally, if traffic data are available, the optimal path that considers also the amount of traffic between each node can be evaluated. In a first attempt, we try to employ the A\* algorithm considering the traffic volume as weight; however, since there are several rural roads outside the urban area where traffic volume is low, the obtained path was very long and not realistic. For this reason, we decide to define a new relationship property based on both distance and traffic volume. The value of this property is estimated as:

$$w_x = 0.5 * \frac{d_x - \min_d}{\max_d - \min_d} + 0.5 * \frac{AADT_x - \min_{AADT}}{\max_{AADT} - \min_{AADT}}$$

where  $d_x$  is the road length corresponding to the 'ROUTE' relation and  $AADT_x$  its average traffic volume,  $\min_d$  and  $\max_d$  are the minimum and maximum distance values computed on all the road network, and  $\min_{AADT}$  and  $\max_{AADT}$  the minimum and maximum AADT values on all the road network. As a result, the weight of each relationship is the equal-weighted sum of the normalized values of distance and AADT. This solution allows finding the optimal path, considering both the traffic conditions and the length of the resulting path. In Fig. 2, the three paths obtained applying the routing procedure between the same source and the same target node but with different approaches are displayed. We can observe that the three resulting paths are slightly different.

## 6.1 Managing Road Closures

For an event or the presence of maintenance work, a street may needs to be closed for a short period. In this case, the traffic manager need to know an alternative path that avoids traveling through certain streets. To enable this

functionality, our street graph can be dynamically modified, setting the status of a road (identified by its road name) to ‘close’ or ‘open’ to traffic. All the ‘ROUTE’ relations in our road network are characterized by the ‘status’ property automatically set to ‘open’ when the *primal graph* is generated; however, the user can easily change this status by running our ‘ChangeStreetStatus’ script (available in the git repository). This script takes as input the name of the street (e.g. Via Wiligelmo), the new status, and the parameters needed to establish the connection with the Neo4j instance. The status of each ‘ROUTE’ relation that involves the road with the given name is updated and, to allow the correct execution of the routing procedure, the road’s connected POIs are detached and new relationships are established with the other roads in 100m from each POI. When the road is re-opened, the original relationships with the POIs are re-established.

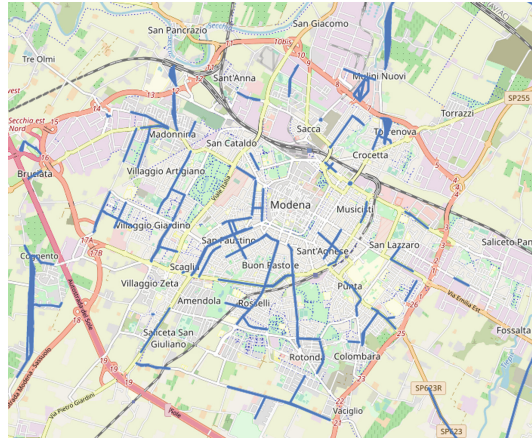
Figure 4 compares the shortest paths obtained considering the traffic volume when the road via Wiligelmo is open or closed. Our routing algorithm finds the best path that avoids passing through the closed street.

## 7 Road Network Analysis

Now that we have converted the road network into a graph, and we have integrated traffic information, we can employ several graph-based algorithms to investigate the graph structure and its relation to traffic. For doing this, we rely on the Graph Data Science library of Neo4j. In order to identify the junction that, given the graph topology, are involved in the majority of the shortest paths, we employed the Betweenness Centrality (BC) algorithm on the *primal graph*. The shortest paths connect two points passing through the minimum possible number of relations. The BC evaluates the shortest paths between all the couples of nodes in the network and then associates to each node a score [9]. This score depends on the number of the shortest paths crossing the node and is evaluated as:

$$score(n) = \sum_{s,t \in N} \frac{sp(s,t|n)}{path(s,t)}$$

where  $n$  is the actual node,  $N$  is the ensemble of all nodes in the graph,  $sp(s,t|n)$  is the number of the shortest paths between  $s$  and  $t$  that passes through  $n$ , and  $path(s,t)$  the total number of the shortest paths between  $s$  and  $t$ . In the city of Modena, we obtained a BC score between 0 and  $8 \times 10^7$ , with an average score of about  $3 \times 10^6$ . Figure 5 displays the first 100 junctions with the highest score. Moreover, we investigate if the BC score of the junction is correlated with the traffic in the incoming and outgoing roads by evaluating the correlation between the BC score and the sum of all the traffic volumes of the roads. Since the Person’s and Spearman’s coefficients were both lower than 0.2 (0.12 and 0.15 respectively), as already discussed and demonstrated in [11], we prove that, also in our use case, the BC score is not significantly correlated with the traffic volume. Then, we try to test the Harmonic Centrality (HC) algorithm:



**Fig. 6.** PR results for the city of Modena.

a modified version of Closeness Centrality that works better with unconnected graphs [19]. The HC score depends on the average shortest path between the node and all the other nodes in the graph. The average shortest path is evaluated summing the inverse of the distances between the given node and all the others. HC algorithm has been applied to the *primal graph* of the city of Modena obtaining a Spearman's correlation coefficient significantly higher (0.61). Similarly, we evaluated the HC score of the *dual graph* and compared it with the sum of all the traffic volumes along each road. We obtain a Spearman's correlation coefficient of 0.45. Thus, there is a relation between HC score and traffic volume. If we want to employ centrality to find the most congested junction, however, the best solution is to exploit the presence of traffic data in the properties of our graph and use the Degree Centrality (DC) algorithm. The DC algorithm is a weighted algorithm; thus, it can be weighted by the ratio between AADT and distance. In this way, the degree of each junction is evaluated as the sum of this ratio in all of the incoming routes. In the city of Modena, we obtained a DC between 0 and  $8 \times 10^3$ , with an average degree of 142. The Pearson's and Spearman's correlation coefficients with the traffic flow are obviously very high (0.95) because the traffic has been used as weight. In Fig. 5, the most important nodes evaluated with BC and DC are compared. We can observe that the result is very different since it conveys different information. The BC assumes that: the drivers are always choosing the shortest path, all the positions in the city have the same importance, and the same number of vehicles are driving through them. The DC, instead, considers the AADT traffic and shows the nodes with the highest incoming traffic: the most congested junctions. The application of centrality algorithms to the *primal graph* is an interesting solution to transfer the traffic information from roads to junctions. As a matter of fact, not only the roads with an high traffic volume are affected by slowdowns; all the vehicles driving through the roads incoming in a congested junction will have a

longer traversal times than expected, even if their route does not involve congested roads. Therefore, to investigate the roads prone to slowdowns, we set up a methodology that uses both the *primal* and the *dual graph*. Firstly, we employ DC on the *primal graph* in order to assign a score to each junction based on traffic. Then, the new ‘score’ property of the junction that connects the two roads is transferred in each relationship between two road sections in the *dual graph*. Finally, we decide to apply the PageRank algorithm (PR) [6] to the *dual graph*. We employ PR considering as weight the ‘score’ property evaluated with DC. In this way, a road can have a high rank if many roads are connected to it, or if some of these connections are through congested junctions. As can be seen in Fig. 6, the framework allows to automatically visualize the roads with a PR higher than the average rank plus two times the standard deviation. In the *dual graph* of the city of Modena, the roads’ ranks are between 0.15 and 7.28, with an average of 0.95; thus, the rank’s distribution is left-skewed with a low number of roads with a high page rank. The displayed road sections are 201. These are only some examples of the insights that can be obtained through the *primal graph* and *dual graph* analysis supported by our framework.

## 8 Conclusion

This paper presented an open-source framework to perform an analysis of the road network, investigate the relation between topology and traffic conditions, and exploit routing algorithms to obtain the optimal path based on different aspects such as distance, traffic volume, number of traversed junctions.

The proposed representation of road networks as the combination of *primal graph* and *dual graph* allows users to apply graph algorithms on cascade on both levels, offering the opportunity to analyze the relationship between the topology of the road network and the traffic distribution. The framework can be efficiently employed by users that are not aware of the graph theory or do not know how a graph database works; moreover, it can provide a good starting point for knowledgeable users that want to conduct deep analytic by applying graph data science and machine learning algorithms. To the best of our knowledge, there are no available open-source frameworks that allow generating both primal and dual graphs and integrating traffic data.

We also explained how we evaluate traffic in our use case and how we manage to evaluate the AADT on the roads where traffic data were missing. Thus, a user that has information about the traffic between a reduced number of nodes can employ our framework to estimate the traffic in the rest of the road network. In future work, the framework can be employed to study the robustness of the road network to different road closure scenarios [28] or compare the road network graphs of different cities considering their traffic.

## References

1. Ahmadzai, F., Rao, K., Ulfat, S.: Assessment and modelling of urban road networks using Integrated Graph of Natural Road Network (a GIS-based approach). *J. Urban Manag.* **8**(1), 109–125 (2019). <https://doi.org/10.1016/j.jum.2018.11.001>. <https://www.sciencedirect.com/science/article/pii/S2226585618301341>
2. Bachechi, C., Po, L.: Implementing an urban dynamic traffic model. In: *IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019*, Thessaloniki, Greece, 14–17 October 2019. ACM (2019)
3. Bachechi, C., Po, L.: Traffic analysis in a smart city. In: *Web4City, International IEEE/WIC/ACM Smart City Workshop: Web for Smart Cities*, Thessaloniki, Greece, 14–17 October 2019 (2019)
4. Bachechi, C., Rollo, F., Po, L.: Detection and classification of sensor anomalies for simulating urban traffic scenarios. *Clust. Comput.* **25**, 2793–2817 (2021). <https://doi.org/10.1007/s10586-021-03445-7>. <https://link.springer.com/article/10.1007/s10586-021-03445-7#citeas>
5. Boeing, G.: OSMnx: new methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Comput. Environ. Urban Syst.* **65**, 126–139 (2017). <https://doi.org/10.1016/j.compenvurbsys.2017.05.004>. <https://www.sciencedirect.com/science/article/pii/S0198971516303970>
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **30**(1), 107–117 (1998). [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X). <https://www.sciencedirect.com/science/article/pii/S016975529800110X>. Proceedings of the Seventh International World Wide Web Conference
7. Chen, J., Song, Q., Zhao, C., Li, Z.: Graph database and relational database performance comparison on a transportation network. In: Singh, M., Gupta, P.K., Tyagi, V., Flusser, J., Ören, T., Valentino, G. (eds.) *ICACDS 2020*. CCIS, vol. 1244, pp. 407–418. Springer, Singapore (2020). [https://doi.org/10.1007/978-981-15-6634-9\\_37](https://doi.org/10.1007/978-981-15-6634-9_37)
8. Claramunt, C., Winter, S.: Structural salience of elements of the city. *Environ. Plann. B. Plann. Des.* **34**, 1030–1050 (2007). <https://doi.org/10.1068/b32099>
9. Freeman, L.: A set of measures of centrality based on betweenness. *Sociometry* **40**, 35–41 (1977). <https://doi.org/10.2307/3033543>
10. Fu, M., Kelly, J., Clinch, J.P.: Estimating annual average daily traffic and transport emissions for a national road network: a bottom-up methodology for both nationally-aggregated and spatially-disaggregated results. *J. Transp. Geogr.* **58**, 186–195 (2017)
11. Gao, S., Wang, Y., Gao, Y., Liu, Y.: Understanding urban traffic-flow characteristics: a rethinking of betweenness centrality. *Environ. Plann. B. Plann. Des.* **40**, 135–153 (2013). <https://doi.org/10.1068/b38141>
12. Jayasinghe, A., Sano, K., Nishiuchi, H.: Explaining traffic flow patterns using centrality measures. *Int. J. Traffic Transp. Eng.* **5**, 134–149 (2015). [https://doi.org/10.7708/ijtpe.2015.5\(2\).05](https://doi.org/10.7708/ijtpe.2015.5(2).05)
13. Jiang, B., Zhao, S., Yin, J.: Self-organized natural roads for predicting traffic flow: a sensitivity study. *J. Stat. Mech. Theory Exp.* **2008** (2008). <https://doi.org/10.1088/1742-5468/2008/07/P07008>

14. Jorge, A.A.S., Rossato, M., Bacelar, R.B., Santos, L.B.L.: A unified graph model for line and segment maps. In: Proceedings of the 10th International Space Syntax Symposium, pp. 146:1–146:11 (2015). <https://www.sss10.bartlett.ucl.ac.uk/wp-content/uploads/2015/07/SSS10Proceedings146.pdf>
15. Jorge, A.A.S., Rossato, M., Bacelar, R.B., Santos, L.B.L.: GIS4Graph: a tool for analyzing (geo) graphs applied to study efficiency in a street network. In: GEOINFO (2017)
16. Liu, T., Jiang, A., Miao, X., Tang, Y., Zhu, Y., Kwan, H.K.: Graph-based dynamic modeling and traffic prediction of urban road network. *IEEE Sens. J.* **21**(24), 28118–28130 (2021). <https://doi.org/10.1109/JSEN.2021.3124818>
17. López, P.Á., et al.: Microscopic traffic simulation using SUMO. In: 21st International Conference on Intelligent Transportation Systems, ITSC 2018, Maui, HI, USA, 4–7 November 2018, pp. 2575–2582. IEEE (2018)
18. Mahmoud, H., Akkari, N.: Shortest path calculation: a comparative study for location-based recommender system. In: 2016 World Symposium on Computer Applications Research (WSCAR), pp. 1–5 (2016). <https://doi.org/10.1109/WSCAR.2016.16>
19. Marchiori, M., Latora, V.: Harmony in the small-world. *Phys. A Stat. Mech. Appl.* **285**, 539–546 (2000). [https://doi.org/10.1016/S0378-4371\(00\)00311-3](https://doi.org/10.1016/S0378-4371(00)00311-3)
20. Marshall, S.: Line structure representation for road network analysis. *J. Transp. Land Use* **9**(1) (2015). <https://doi.org/10.5198/jtlu.2015.744>. <https://www.jtlu.org/index.php/jtlu/article/view/744>
21. Marshall, S., Gil, J., Kropf, K., Tomko, M., Figueiredo, L.: Street network studies: from networks to models and their representations. *Netw. Spat. Econ.* **18**, 1–15 (2018). <https://doi.org/10.1007/s11067-018-9427-9>
22. Miler, M., Odobašić, D., Medak, D.: The shortest path algorithm performance comparison in graph and relational database on a transportation network. *Promet-Traffic Transp.* **26**, 75–82 (2014). <https://doi.org/10.7307/ptt.v26i1.1268>
23. Needham, M., Hodler, A.: Graph Algorithms: Practical Examples in Apache Spark and Neo4j. O’Reilly Media (2019). <https://books.google.it/books?id=UwIevgEACAAJ>
24. Po, L., Rollo, F., Bachechi, C., Corni, A.: From sensors data to urban traffic flow analysis. In: 2019 IEEE International Smart Cities Conference, ISC2 2019, Casablanca, Morocco, 14–17 October 2019, pp. 478–485. IEEE (2019)
25. Porta, S., Crucitti, P., Latora, V.: The network analysis of urban streets: a dual approach. *Phys. A Stat. Mech. Appl.* **369**(2), 853–866 (2006). <https://doi.org/10.1016/j.physa.2005.12.063>. <https://www.sciencedirect.com/science/article/pii/S0378437106001282>
26. Porta, S., Crucitti, P., Latora, V.: The network analysis of urban streets: a primal approach. *Environ. Plann. B. Plann. Des.* **33**(5), 705–725 (2006). <https://doi.org/10.1068/b32045>
27. Qi, L., Schneider, M.: Trafforthm - a traffic-aware shortest path algorithm in real road networks with traffic influence factors. In: Proceedings of the 1st International Conference on Geographical Information Systems Theory, Applications and Management - GISTAM, pp. 105–112. INSTICC, SciTePress (2015). <https://doi.org/10.5220/0005350701050112>
28. Sohounou, P.Y., Christidis, P., Christodoulou, A., Neves, L.A., Presti, D.L.: Using a random road graph model to understand road networks robustness to link failures. *Int. J. Crit. Infrastruct. Prot.* **29**, 100353 (2020). <https://doi.org/10.1016/j.ijcip.2020.100353>. <https://www.sciencedirect.com/science/article/pii/S1874548220300172>

29. Wang, G.M., Li, Y.Q., Xu, M.: Integrating the management and design of urban road network to alleviate tide traffic\*. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 708–713 (2019). <https://doi.org/10.1109/ITSC.2019.8917083>
30. Zhang, J., Yuan, L., Li, W., Qin, L., Zhang, Y.: Efficient label-constrained shortest path queries on road networks: a tree decomposition approach. *Proc. VLDB Endow.* **15**(3), 686–698 (2021). <https://doi.org/10.14778/3494124.3494148>