electronics

MDPI

Article

A Centralized Route-Management Solution for Autonomous Vehicles in Urban Areas

Jorge Luis Zambrano-Martinez [1,*], Carlos T. Calafate [1], David Soler [2], Lenin-Guillermo Lemus-Zúñiga [3], Juan-Carlos Cano [1], Pietro Manzoni [1] and Thierry Gayraud [4]

[1] Department of Computer Engineering (DISCA), Universitat Politècnica de València, 46022 Valencia, Spain
[2] Institute of Multidisciplinary Mathematics (IMM), Universitat Politècnica de València, 46022 Valencia, Spain
[3] ITACA Institute. Universitat Politècnica de València, Camino de Vera S/N, 46022 Valencia, Spain
[4] LAAS/CNRS—Groupe OLC, Université Paul Sabatier-Toulouse III, 31031 Toulouse, France
* Correspondence: jorzamma@doctor.upv.es; Tel.: +34-96-387-7007

Received: 23 May 2019; Accepted: 20 June 2019; Published: 26 June 2019

check for updates

Abstract: Currently, one of the main challenges that large metropolitan areas must face is traffic congestion. To address this problem, it becomes necessary to implement an efficient solution to control traffic that generates benefits for citizens, such as reducing vehicle journey times and, consequently, environmental pollution. By properly analyzing traffic demand, it is possible to predict future traffic conditions, using this information for the optimization of the routes taken by vehicles. Such an approach becomes especially effective if applied in the context of autonomous vehicles, which have a more predictable behavior, thus enabling city management entities to mitigate the effects of traffic congestion and pollution, thereby improving the traffic flow in a city in a fully centralized manner. This paper represents a step forward towards this novel traffic management paradigm by proposing a route server capable of handling all the traffic in a city, and balancing traffic flows by accounting for present and future traffic congestion conditions. We perform a simulation study using real data of traffic congestion in the city of Valencia, Spain, to demonstrate how the traffic flow in a typical day can be improved using our proposed solution. Experimental results show that our proposed traffic prediction equation, combined with frequent updating of traffic conditions on the route server, can achieve substantial improvements in terms of average travel speeds and travel times, both indicators of lower degrees of congestion and improved traffic fluidity.

Keywords: autonomous vehicle; traffic prediction; SUMO; route server; DFROUTER; intelligent transportation system

1. Introduction

A serious problem in urban areas is the high population density, which leads to frequent traffic congestion conditions in critical areas of a city, increasing the travel time of vehicles accordingly. Such increase is directly associated with noise, accidents, unwanted delays, unnecessary fuel consumption, and, therefore, an increase in carbon dioxide (CO_2) emissions, all critical issues for both citizens and city authorities [1].

In the last few years, we have seen how a novel mobility paradigm focused on automated vehicles has emerged, and it is steadily gaining interest. As autonomous vehicles gradually become ubiquitous in coming years, they also present new opportunities to improve traffic by endowing traffic managers with more intelligent ways to regulate traffic when compared with the usual strategies, i.e., traffic light synchronization, or the deployment of on-site traffic agents. This way, the centralized administration of routes emerges as an approach with the potential to offer authorities total control of the traffic flow within their control domain. This new way of handling vehicular traffic can optimize traffic flows with high effectiveness by determining the route of each particular vehicle. Notice that such a centralized approach is much more beneficial than relying on the vehicle itself to decide the best route based on knowledge of the traffic congestion status beforehand, as the latter approach is prone to cause intermittent congestion problems in the different streets of a city. Therefore, current vehicular route servers rely on locally stored static information which is used to calculate the requested routes. There are several commercial and free solutions including TomTom and Google Maps Navigator. The first deficiency of these systems is that they are based on static information, or on real-time feedback in the best case, but they do not predict future traffic conditions, nor do they to perform any sort of load balancing.

When moving towards new traffic management paradigms, knowledge about the behavior of street segments of a city becomes mandatory in terms of the number of vehicles traveling in a segment, along with their corresponding travel time. Such knowledge is essential to efficiently model a city, and a prerequisite for traffic management systems.

In a previous work [2] we created realistic traffic models that reliably describe vehicular behavior in the city of Valencia, and the characterization of the traffic conditions of the different streets during rush hours, by predicting the impact of traffic conditions on travel times [3]. We started by using public data associated with induction loop measurements that are made available by the City Hall of Valencia [4]. We then relied on the DFROUTER tool [5], along with an iterative heuristic that refines the output produced by this tool, to obtain the Origin-Destination (O-D) traffic matrix that represents the internal trips inside the target area, and it resembles the real traffic distribution in the city [6]. The next step consisted of the characterization of travel times on a per-segment basis by applying a logistic regression [7].

The main objective of our work is to propose and implement a centralized route manager for autonomous vehicles with the ability to optimize and balance traffic flows by accounting for present and future traffic congestion conditions in terms of travel time depending on the vehicle load, thereby allowing us to predict the traffic distribution. Hence, in this work, we extend our previous contribution by performing an experimental study where we evaluate the effectiveness of our route server, named *Automatic Balancing of Traffic through the Integration of Smartphones with vehicles* (ABATIS), at mitigating traffic congestion. To this aim, we focus on the traffic management during peak hours for the city of Valencia, using the reference traffic conditions, obtained as described above, as input for the Simulation tool of Urban MObility (SUMO) tool [8], and coupling it with the Objective Modular Network Testbed in C++ (OMNeT++) network simulator [9]. By deriving an equation that is able to establish the relationship between the street occupation along each possible route and the predicted travel time delay, our route server is able to provide routes to all vehicles in the city, and minimize travel times by balancing traffic throughout the city. Our experiments show that ABATIS can improve traffic fluidity by increasing travel speeds in the range from 16 to 35 km/h, thereby reducing travel times by 8% on average. As an additional advantage, ABATIS works with real maps, meaning that it can directly act as route server for real vehicles without requiring further changes to the software itself.

The rest of this paper is organized as follows: Section 2 presents some related works addressing traffic flow improvements using different techniques. Section 3 provides information about the SUMO and DFROUTER tools, along with the iterative heuristic used to mimic real traffic traces, and it also includes details about the route server used. Section 4 describes the methodology used to improve traffic flow using an interface that establishes a communication link between the two simulators and the route server. In Section 5, we analyze the performance of our solution by comparing the improved traffic flow with respect to the original traffic flow, evaluating different metrics such as average speed, average travel time, arrival times for the vehicles, total distance between routes and number of vehicles that end their routes. Finally, Section 6 concludes the paper and discusses the relevance of the results obtained, together with their potential regarding future traffic management systems.

2. Related Works

The prediction of traffic flows, and the adoption of traffic management strategies, are considered to be key elements for Intelligent Traffic Systems, which are usually developed from simple traffic models or algorithms. Among the many prediction models available in the literature, Mir et al. [10] consider that those based on Kalman filters are able to optimize speed predictions by minimizing the variance between the measured real-time speed and its estimation. Similarly, Abidin et al. [11] used a Kalman filter model to predict the arrival time of public transport vehicles using information captured from social networks. However, making Kalman filter-based estimations is often complex, and requires a high computational power to predict traffic, in conjunction with specialized time series analysis software [12].

Other methods for predicting traffic are the use of neural networks applied to short-term time periods. For instance, Ma et al. [13] proposed a novel architecture called Long Short-Term Neural Network to estimate dynamic traffic effectively. Similarly, Tian et al. [14] propose a model, called Long Short-Term Memory Recurrent Neural Network (LSTM RNN), that captures the randomness of the traffic flow and corrects problems of back-propagated error through the memorization of historical data to determine the optimal times of travel delay. In general, the greatest drawback of neural networks is the long training time to have a high prediction accuracy.

In [15], the authors proposed a traffic flow prediction approach that is based on deep learning as the stacked autoencoder model to learn and train the characteristics of the generic traffic, and thus obtain the prediction of traffic. In [16] authors propose a constrained optimization method for vehicles that leverages historical data to improve the performance of a network of self-driving taxis; this method computes a probability distribution over future demand based on historical data. Basically, this method can assign thousands of requests to thousands of autonomous vehicles using a probability distribution for request considering specific origin and destination region. This proposal manages only historical data of taxis, and not the overall traffic behavior for a city.

In the same way, there are random ways of predicting the best routes to decongest the traffic as in [17] where the authors present a delegated multi-agent system for anticipatory vehicle routing to avoid traffic congestion where vehicles issue light-weight agents for exploring alternatives routes in the environment. Thus, based on the evaluation of the alternative routes, vehicles issue a second type of light-weight agent for allocating roads segments, spreading the vehicles' intentions, and coordinating their behavior. This propose is related to ant colonies.

Zheng et al. [18] proposed an adaptive control model in real time for street intersections regulated by semaphores. In addition, they incorporated a prediction model which estimates the future vehicles arrival rates based on the available information and the semaphore synchronization plan. Gartner et al. [19] presented the operational characteristics of an adaptive algorithm, and its implementation in the field of real-time adaptive traffic control systems. Lin et al. [20] presented a traffic controller structured throughout the vehicular network using Model Predictive Control theory, which can control urban traffic, specifically situations where traffic flows are not uniformly distributed throughout the network.

In [21] the main goal is to devise a routing and rebalancing algorithm for autonomous vehicles fleets to minimize congestion. The customers are routed to their destination using the shortest-time path as computed by the A* algorithm, and the travel time along each edge is computed using a heuristic delay function that is related to the volume of traffic on each edge. Also, they obtain a set of rebalancing flows that is then decomposed into a set of rebalancing paths via a flow decomposition algorithm, and each rebalancing path connects one origin region with one destination region. However, the congestion model is simpler and less accurate than typical congestion models used in the transportation community. Agafonov et al. [22] compare the efficiency of a vehicle-routing algorithm based on the shortest path. All route calculations rely on a centralized traffic management agent, as well as the rerouting procedure. They used the Greenberg model as the routing algorithm in a time-dependent network. Therefore, they perform an obvious modification of the routing algorithm adapted to a time-dependent network that can rebuild the route during a predeterminate time interval for routing. During the rerouting procedure, the authors consider that the order of choosing the vehicle is important and they rely on ordering based on distance to the destination in priority for short routes and priority for long routes.

Ren et al. [23] proposed a mobility-based and stability-based clustering algorithm for urban scenarios that takes into account the movement of the vehicle, its relative position, the estimation of the useful link life between vehicles, and the traffic flow rate. Similarly, Nafi et al. [24] proposed a predictive road traffic management system based on a Vehicular Ad-hoc Network architecture. Their solution relies on a communications scheme to estimate the intensity of future traffic at different intersections based on a linear predictive algorithm that accounts for maximum speed, scenario, and routes. Wan et al. [25] proposed a method to use data derived from connected vehicles to increase transport efficiency, such as estimating and predicting travel times, so that drivers avoid congestion dynamically. Liebig et al. [26] presented a system to plan trips individually by calculating the future conditions of traffic through a random space-time field based on the readings collected from sensors. Likewise, Kamal et al. [27] proposed a predictive control model that improves the flow of traffic as it regulates the safe distance between vehicles, and the bounded driving torque condition to predict traffic while avoiding traffic jams. Furthermore, the traffic flow prediction can be presented by computational models, such as the Local Space-Time AutoRegressive approach proposed by Chen et al. [28]. However, all these previous authors did not experiment with vehicle mobility data in the proposed scenario, comparing their approach with other theoretical models, but not with a real traffic model.

This paper differs from previous works by performing an experimental study of traffic flows in the city of Valencia using real rush hour traffic. By analyzing traffic demand in real time, and by relying on our Traffic Prediction Equation, we can predict future traffic conditions, and thereby our proposal provides optimized routes to all vehicles in the city using our proposed route server. Also, differently from previous works, our route server can be directly deployed to serve routes to real vehicles without requiring any software modifications.

3. Overview of the Simulation Tools Adopted

In this section, we provide some details about the mobility simulator used, called SUMO. Then, we will introduce the DFROUTER tool, together with our proposed iterative heuristic to generate the O-D matrix based on induction loop data; such data were retrieved from real traffic traces during the rush hour for a working day in Valencia (8 to 9 a.m.). Lastly, we will describe the functionality of our route server (ABATIS).

3.1. SUMO

Commonly, vehicle traffic models consist of obtaining several variables that help to understand traffic behavior and to improve traffic management, detailing times of departure and arrival, speeds, routes followed by the vehicles injected into the network, and the streets that the vehicles have passed through, with the corresponding travel time.

The SUMO [29] is a tool that performs simulations of vehicular mobility through detailed microscopic modeling of cities and vehicles. In fact, this simulator is constantly improving, as it is open source and widely accepted and validated by the scientific community. Among its features it includes the simulation of multimodal traffic and traffic scheduling, and it supports different map formats, such as OpenStreetMap, importing the road network in multiple formats, and generating routes with different sources. Additionally, it offers high performance simulations through the Traffic Control Interface (TraCI) [30], allowing interactive access to the simulation of route traffic with many more features, e.g., manipulating their behavior online when coupled with another simulation tool such as OMNeT++ [31]. Multimodal simulation support in SUMO is one of the main advantages, as it not only includes the movement of vehicles in the city, but it also includes the public transport system, the rail networks, and even the pedestrian routes. This means that traffic mobility can be described by multiple routes, being that each of them can be composed of sub-routes. In SUMO, the vehicular traffic flow is simulated microscopically, meaning that vehicles are simulated individually within the network. This characteristic allows us to determine parameters such as their acceleration, speed, location, time of arrival, and time of departure. Moreover, it allows a discrete simulation of continuous mobility in space, where each time step is one second. In addition, the simulations consider the different attributes of streets, such as maximum speed, priority rules to the right, and realistic driver models. Although SUMO was not originally designed to address autonomous vehicles, it is a very flexible tool which can include different driver models. In fact, SUMO has been recently updated to model autonomous vehicles, and there are several scientific publications that use this mobility simulator for that purpose [32,33].

3.2. Extending DFROUTER with an Iterative Heuristic

Within the packages included in the SUMO distribution version 0.32.0, the DFROUTER [5] tool is useful when attempting to obtain an O-D matrix. This tool allows us to determine the inflow and outflow of the roads, starting from induction loop counts for the different streets within a city, thereby achieving an estimation of the possible vehicles routes that match such input. Likewise, this tool can reconstruct the number of vehicles and routes to be injected into the road network simulator, based on the data extracted from the induction loops such as vehicle speeds, flows, and vehicle count, to achieve the desired O-D traffic matrix [34].

In a previous work [6] we used induction loop data provided by the City Council of Valencia, Spain, as input to DFROUTER. Those data sets were created by 520 induction loop groups distributed throughout the city, and correspond to the rush hour between 8:00 am and 9:00 am on a typical Monday in November. In that preliminary work we noticed that there was a significant discrepancy between the traffic generated by the tool and the original data. To address this issue, we proposed an iterative heuristic to compensate for that error through a refinement of the output provided by the DFROUTER tool. This allowed us to achieve an O-D matrix that closely resembles the real traffic distribution. In fact, the output of the DFROUTER using our iterative heuristic achieves a high level of matching when compared against the reference data, reducing the error to just 0.0001% (see Figure 1).

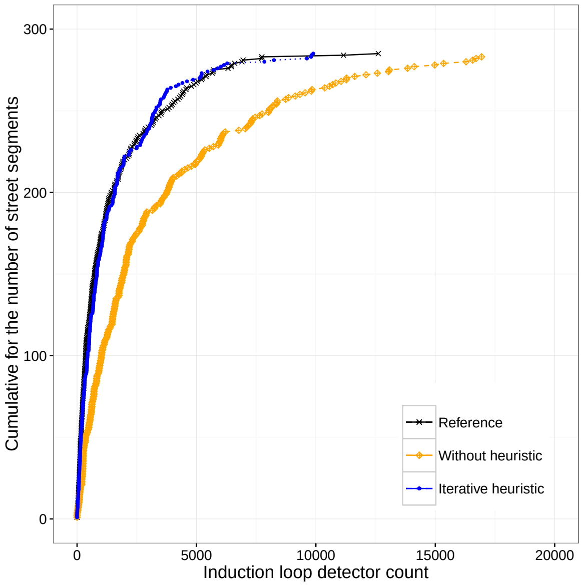


Figure 1. Traffic flow modeling for Valencia. Results with and without our iterative heuristic [6].

3.3. The ABATIS Route Server

Presently, vehicular route servers basically rely on static information locally stored, and based on which new routes are calculated. There are several similar systems, including both commercial

(TomTom) and free solutions (Google Maps Navigator). The first deficiency of those systems is that they combine historical data with data sent by drivers' smartphones, portable browsers, collaborative social networks, or any reporting application, meaning that users must constantly send reports about where they are, and how long it takes to get from one place to another. Such approaches are not effective if the users do not share this information to protect their privacy, or due to regulatory conditions such as the General Data Protection Regulation (GDPR). Besides, the second deficiency is that both Google Maps and TomTom perform only short-term predictions based on past and current status, but do not act as centralized controllers for all the traffic in a city. Thus, they cannot perform medium-term predictions nor balance traffic as we do. Therefore, we aim at a solution that is powered by the data coming from the induction loops deployed on the streets of Valencia, and made available by the City Hall, to avoid relying on any smart device as the primary data collection device, as in the case of Google Maps or TomTom.

To meet the aforementioned requirements, a route planning server called ABATIS [4] was devised in order to provide route recommendations through a client-server interaction, and whose main objective is to offer the best possible routes, taking into account the cost of traveling on each street segment. ABATIS includes a characterization of the different streets of a city in terms of travel time depending on the vehicle load, thus allowing us to predict the traffic distribution. The server itself responds to route requests via an HTTP connection, which includes the listening port for the requests, the geographical coordinates for the starting and destination points, the type of service offered, and the output format. The answer returned by ABATIS is in the JSON format, and contains information about the route that a vehicle will follow as a sequence of geographical coordinates, along with parameters such as: status of the request, maneuver, duration, driving side, name of the street, and street length. Likewise, for improving the performance of the query, and for a live update of the traffic data on the route server, a modification of Dijkstra's algorithm called Multilevel Dijkstra (MLD) is used. Notice that our modified algorithm is aware of this multilevel partitioning by using precomputed overlay cells to heuristically speed up the computation [35].

For ABATIS to be fully updateable when performing traffic routing within any city, the following steps are followed:

1. Extract a city graph from the OpenStreetMap map database into a normalized format, and then add routing metadata.
2. Partition the city graph recursively into cells to improve route computation.
3. Customize those cells to calculate all possible routes, together with additional information such as traffic updates, in a periodic manner.
4. Store results in shared memory, thus allowing data sharing between different map updating processes to take place.
5. Run the route server through a known IP and port to attend any incoming requests.

In addition, to achieve load balancing, it becomes necessary to provide ABATIS with updates regarding predicted average speed on a per-street basis, in step 3; this way, the route server will understand that there is a traffic jam in those streets, and it will be able to provide different alternative routes to the vehicle that made the request. Therefore, steps 3 and 4 must be repeated each time the traffic is updated without having to restart the server. Notice that the limitations of the proposed centralized architecture are closely associated with the characteristics of the server used for that task. In our experiments, we have used a PC with modest features—Intel Core i5 quad-core processor with 16 GB of RAM—finding that it is possible to serve up to 2000 vehicles per second without using parallel processing, and with an average CPU consumption of only 10%. Also, each response from our server takes between 0.5 ms to 4 ms, depending on the length of the route requested by the vehicle. Thus, we do not foresee any scalability problems for our solution.

4. Methodology

In this section, we describe the procedure followed to improve the traffic flow in the city of Valencia, Spain. In particular, we will focus on achieving the load balancing of vehicles by relying on Equation (1) presented in a previous work, which is able to adequately predict travel times depending on the degree of traffic congestion on a per-segment basis as shown in [7].

$$f(x) = \frac{a}{1 + e^{b - \frac{x}{c}}} - \frac{a}{1 + e^{b}} + t_{ff} \qquad (1)$$

In this Equation, variable x represents the number of vehicles in the street segment, t_{ff} is the free-flow travel time in the segment, b is a parameter to adapt the curve of the formula to the travel time in free-flow conditions (zero vehicles ahead), a is the actual maximum value for the travel times measured, and parameter c determines its corresponding displacement in the abscissa axis.

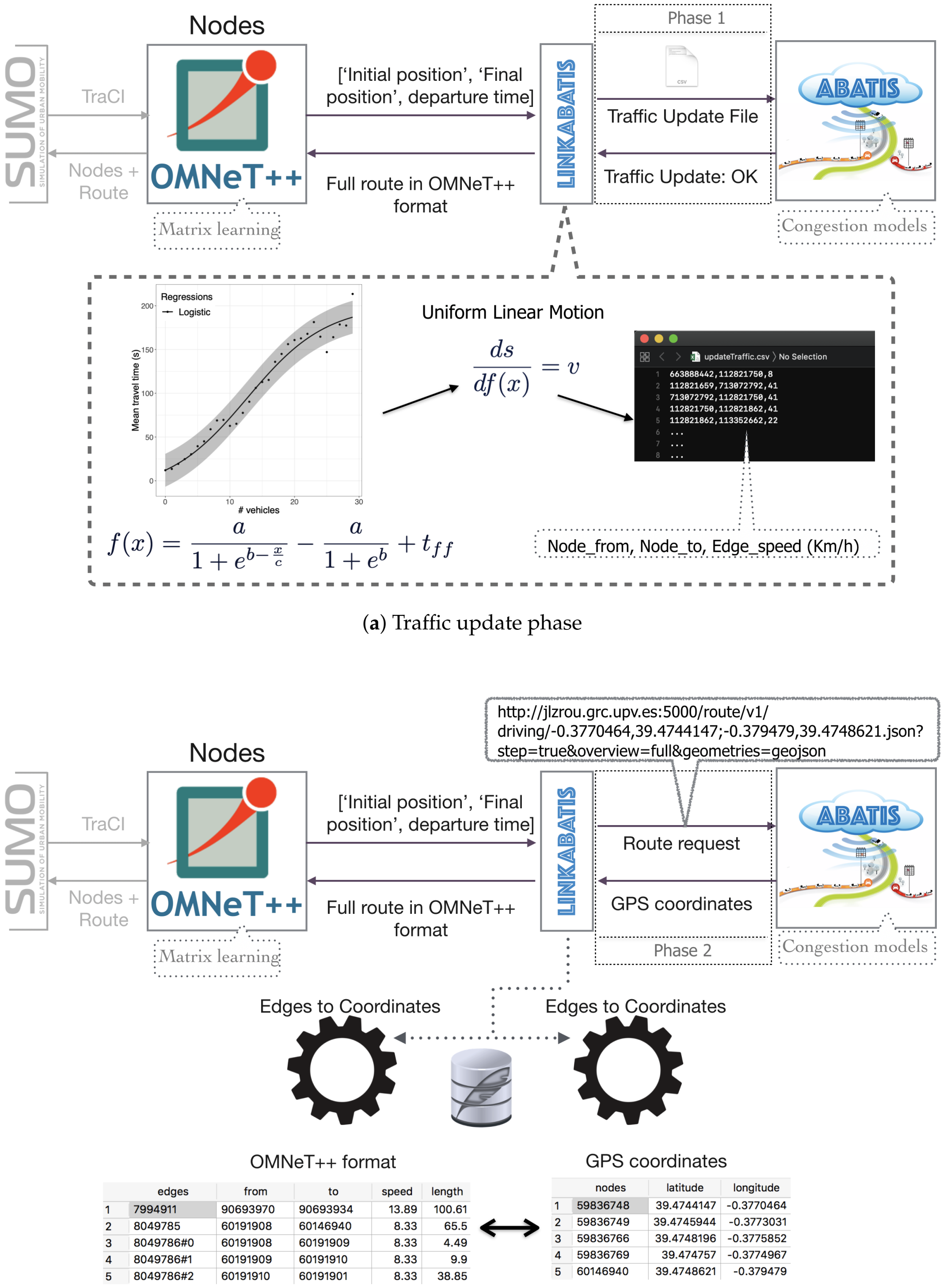
In particular, our objective is to demonstrate the validity of the proposed characterization and traffic prediction for this city. With this purpose in mind, we will compare the original traffic conditions with our proposed approach, which is able to balance traffic by exploiting different alternate routes to decongest traffic in critical areas. To achieve this goal, we have connected our vehicular simulators (SUMO and OMNeT++) with the ABATIS route server. However, since these simulators and the route server use different formats for geopositioning data, they cannot communicate directly. Hence, we developed an interface that performs the necessary calculations to perform a conversion between both formats. In the following subsection we describe in more detail how this interface works.

4.1. The linkABATIS Interface

The existing interface between the SUMO traffic simulator and the OMNeT++ network simulator is called TraCI [30], and it allows us to investigate how vehicles move within urban areas under various traffic patterns, and using real-life driving models.

The TraCI interface connecting SUMO and OMNeT++ is open source, and it allows full control of the vehicle mobility, and of the attributes of each simulated vehicle. We modified this architecture to obtain more information about simulated vehicles, such as the segments that a vehicle has passed, the distance traveled by a vehicle, the average speed per segment, and for how long a vehicle has been in each segment. We then integrated such information within our process for characterizing and predicting the traffic in the city of Valencia.

One of the objectives to be achieved when developing the linkABATIS interface was to solve the traffic congestion problems detected in some parts of the city in the best way possible. Specifically, it will enable vehicles to use alternative routes to balance the traffic, thus improving traffic flow during rush hours. To accomplish this goal, it is necessary to have an individual characterization of street segments in terms of average travel times experienced by vehicles for different degrees of congestion, whose estimation is based on the number of vehicles found ahead of a vehicle that just enters a segment. Then, the linkABATIS interface processes and sends the necessary information to the ABATIS route server so that it updates the weights of the street segments by accounting for the expected segment occupation to estimate speed levels. The speeds for each segment are calculated using the Uniform Linear Motion equation ($\frac{ds}{df(x)} = v$). Notice that beforehand, we have the average travel time estimation, as provided by Equation (1), along with the length of each street in the city. The scheme detailing the first phase of our intelligent interface, is shown in Figure 2a.



$$\frac{ds}{df(x)} = v$$

$$f(x) = \frac{a}{1+e^{b-\frac{x}{c}}} - \frac{a}{1+e^{b}} + t_{ff}$$

updateTraffic.csv 〉 No Selection

```
663888442,112821750,8
112821659,713072792,41
713072792,112821750,41
112821750,112821862,41
112821862,113352662,22
...
...
...
```

(**a**) Traffic update phase

http://jlzrou.grc.upv.es:5000/route/v1/driving/-0.3770464,39.4744147;-0.379479,39.4748621.json?step=true&overview=full&geometries=geojson

OMNeT++ format

	edges	from	to	speed	length
1	7994911	90693970	90693934	13.89	100.61
2	8049785	60191908	60146940	8.33	65.5
3	8049786#0	60191908	60191909	8.33	4.49
4	8049786#1	60191909	60191910	8.33	9.9
5	8049786#2	60191910	60191901	8.33	38.85

GPS coordinates

	nodes	latitude	longitude
1	59836748	39.4744147	-0.3770464
2	59836749	39.4745944	-0.3773031
3	59836766	39.4748196	-0.3775852
4	59836769	39.474757	-0.3774967
5	60146940	39.4748621	-0.379479

(**b**) Route request phase

Figure 2. The different phases associated with our smart Interface.

In a second phase, the linkABATIS interface is used to query for the route of a vehicle accounting for the updated traffic. To send this query to ABATIS, it is necessary to know the initial geographic location and the target geographic location of the vehicle, together with the vehicle time of departure. Based on this input, ABATIS will return the best complete route as a sequence of geographic coordinates. However, the format of the route proposed by ABATIS is not compatible with the format accepted by SUMO and OMNeT++. To address this incompatibility, it is necessary to create a translator function that converts the geographical coordinates into codes that are compatible with both the network and the traffic simulator, as well as the inverse conversion to support bidirectional communications.

To improve the speed of processing and search in the conversion of geographic coordinates from one format to the other, we store the information relevant to all nodes, such as their ID code and geographic coordinates. Also, we store the "edges", which are representations of streets or roads in the context of SUMO. These edges contain the identification code of the corresponding nodes, edge type information, the edge length, the vehicular speed associated with that edge, and the different street segments that have been unified, as presented in a previous work [7]. Figure 2b represents the scheme of the second phase of our smart interface.

Due to the format differences between both systems, ABATIS, as a route provider, is not 100% accurate; for instance, if we have a long street within a route, this route server would choose to deliver the geographical coordinates associated with the beginning and end of the street, but not its intermediate points, which are at times required. For this reason, it is necessary to make certain corrections so that the information sent by ABATIS can be understood and processed by OMNeT++ and SUMO. To this aim, we perform searches among the Edge IDs, and match the missing Edge ID through the identification code of the nodes, and this way it corrects the gaps in the route to be used by the simulators. Thus, in Algorithm 1 we detail the process followed when a vehicle requests a new route; notice that it updated traffic knowing the conditions of future traffic. By using this strategy, it is possible to improve the travel time of the vehicles in the city of Valencia, as simulation experiments will show, which indirectly implies a reduction in both fuel consumption and CO_2 emissions.

Algorithm 1 Traffic update and optimal route request.

```
Require: Road Network, Vehicles files, Edges files, Reunified Segment file, Segment-info files, and Statistical learning by segment file
Ensure: Traffic Update file, Optimal Route
firstTimeInABATIS = true
while ABATIS server route is activated do
    if firstTimeInABATIS is true then
        $Extract\ module$ ← extract a graph out of the OpenStreetMap base map
        $Partition\ module$ ← partition the graph recursively into cells
        $Customize\ module$ ← customize the cells with default weights for all cells
        $Datastore\ module$ ← load the data into shared memory for sharing among different processes (requests)
        $Routed\ module$ ← Run the route server engine that will hot-swap to the new dataset
        firstTimeInABATIS = false
    else
        for edge in Edges file do
            $edgeIDInitial[]$ ← vector that stores all initial edge IDs where the vehicles will be injected into the road network as starting points
        end for
        for vehicles in Vehicles file do
            if $(firstEdge\_Vehicle = edgeIDInitial[])$ then
                $vehicle\_info[]$ ← vector that stores all information about vehicles together with the time that the vehicle will be injected into the road network
            end if
        end for
        for vehicles in Statistical learning by segment file do
            if $(vehicle[ID] = vehicle\_info[vehicleID])$ then
                $v[]$ ← vector that stores all numbers of vehicles in the segment before a new vehicle enters the segment
            end if
        end for
        for segment in Statistical learning by segment file do
            if ((vehicle_info[segmentID] = segment[ID]) and (vehicle_info[segmentID] not partitioned)) then
                v = segment[speed] ← gets the same original speed of the non-reunified segment
            else
                $a, b, c$=segment[ID][a][b][c] ← get the three parameters that characterize the traffic using the logistic function
                $t_{ff}$=segment[ID][t_ff] ← get the free-flow travel time (zero vehicles ahead) in the segment
                $f(v) = \frac{a}{1+e^{b-\frac{v}{c}}} - \frac{a}{1+e^{b}} + t_{ff}$ ← get the mean travel time through the Traffic Prediction Equation knowing numbers of vehicles ahead
                v=$\frac{ds}{df(v)}$ ← get the speed through the Linear Motion Equation knowing the mean travel time and the segment length
            end if
            Traffic Update file ← stores all nodes that compose the edges that will change their weight in the routes server together with their new speed (v)
        end for
        ABATIS server route ← receives the Traffic Update file that specifies edge weight updates
        $Customize\ module$ ← customize the cells by calculating routing weights for all cells through Traffic Update File
        $Datastore\ module$ ← load the data updated into shared memory for sharing among number of processes (requests)
        linkABATIS → transform initial and final position from simulators format to GPS coordinates
        ABATIS server route ← receives vehicle information including ID, initial and final position in GPS
        $Routed\ module$ → return the optimal route according to the traffic updated in GPS using MLD Algorithm
        linkABATIS → transforms the optimal route from GPS to simulators format for $vehicle[ID]$
    end if
end while
```

5. Experimental Results

As explained in the previous section, we have interconnected the SUMO mobility simulator and the OMNeT++ network simulator through an interface called TraCI. Additionally, these two simulators are linked to the ABATIS route server through a smart interface, called linkABATIS, that we developed. Our main purpose is to solve congestion issues arising in some areas of the city by using alternative routes, thereby achieving distributed traffic load balancing. This was made possible due to our prior efforts regarding the characterization and prediction of traffic congestion using our proposed traffic congestion Equation (1).

We detail two scenarios for experimentation, studying on each of them the impact of traffic load balancing on the different metrics of interest, including average vehicle speed, average travel time, vehicle arrival times, the different routes that are achieved when applying our equation, and heat maps that indicate the areas with the highest traffic congestion levels for both the real traffic (reference) and the improved traffic.

5.1. Uniform Traffic Load Balancing

In this section, we study the impact of load balancing traffic throughout the whole city of Valencia (excluding suburban areas), with a size of 77.42 km^2. To achieve this goal, the reference traffic scenario for the city of Valencia consists of injecting a total of 34,065 vehicles corresponding to real traces at the rush hour [2], and for a period of 900 seconds.This scenario allows us to check the effectiveness of our traffic congestion equation for a city, reducing congestion, and balancing the flow of traffic globally.

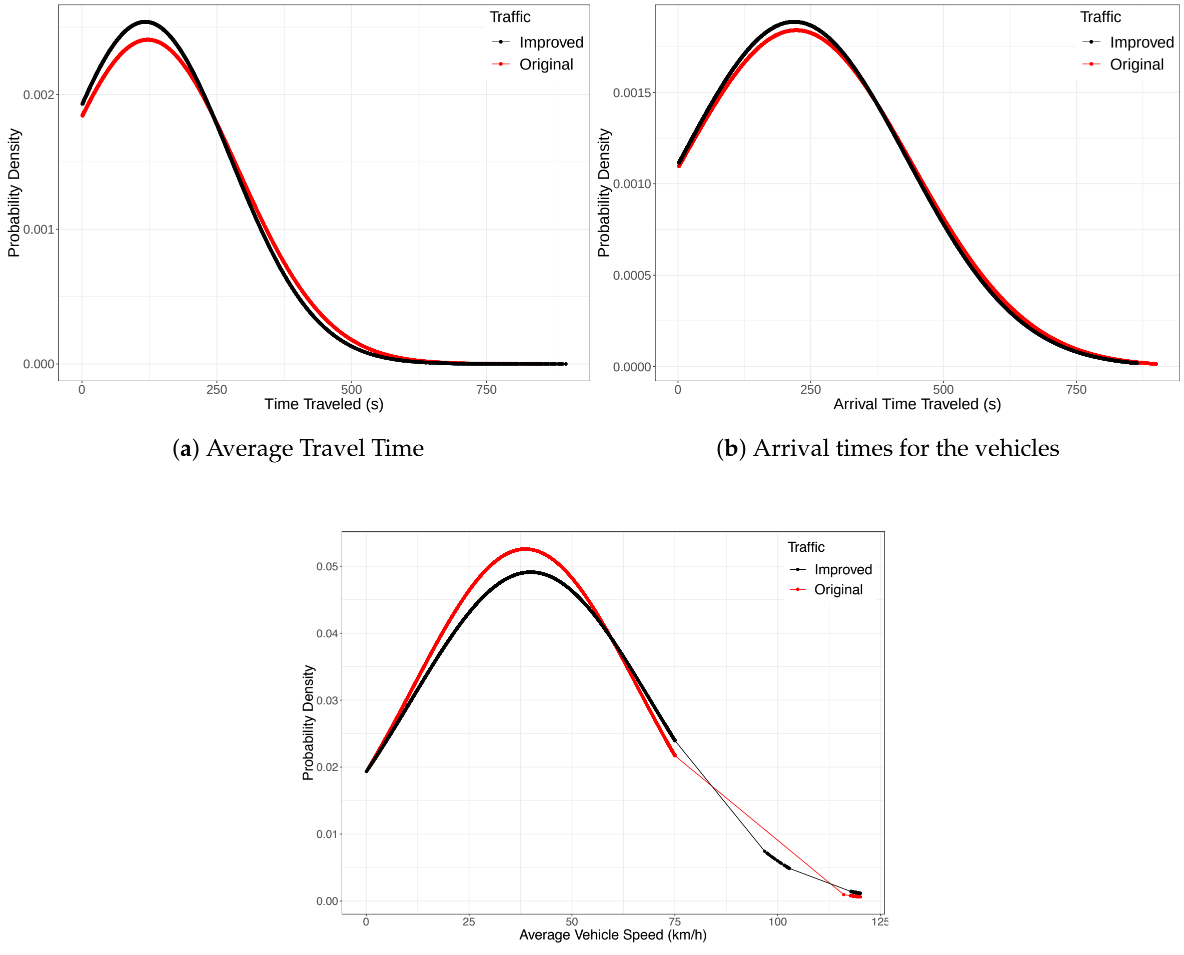
Figure 3a shows the improvements that have been obtained in terms of average travel times for the entire city, demonstrating that our traffic prediction was effective. In particular, we find that the prediction of traffic reduces the time the vehicles require to arrive to their destinations. We can also see that the average number of vehicles traveling on the streets and falling within the Gauss bell is approximately 5%, and there are very few vehicles whose travel time is high compared to the vehicles in the reference traffic.

Likewise, in Figure 3b we observe that vehicles tend to arrive much earlier when our load-balancing technique is used compared to the normal behavior of the reference traffic; in fact, the standard deviation is reduced even taking into account that the scenario encompasses the whole city.

Another metric used to assess vehicular traffic behavior is the average speed that vehicles have in each of their routes. Thus, in Figure 3c we observe that when doing traffic load balancing, vehicles move faster than under reference traffic conditions. This is mainly due to the fact that when applying our strategy, a vehicle has several routes to reach its destination, and the less congested route is chosen for the vehicle to arrive at its destination.

Likewise, it can be observed that in the improved traffic results, our algorithm is able to reduce congestion in some of the streets, allowing vehicles to increase their speed progressively and efficiently, verifying the correct balancing of traffic introduced by our algorithm.

To compare the routes taken by the vehicles within the reference traffic scenario against the scenario with a balanced load, we have chosen some examples of the longest routes, along with their main metrics, for comparison purposes. As shown in Figure 4, the route chosen by our algorithm is the most optimal between the different route options, introducing a lower travel time, and being less congested at the time the vehicle reaches each of the street segments along the proposed path.



(**a**) Average Travel Time

(**b**) Arrival times for the vehicles

(**c**) Average vehicle speed

Figure 3. Improvements our proposed approach in the city of Valencia.

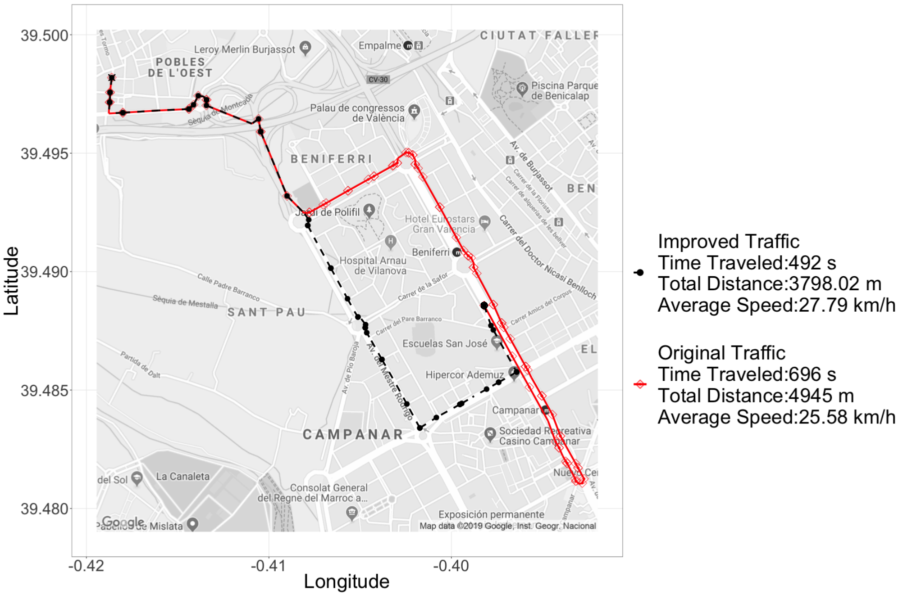


Figure 4. *Cont.*

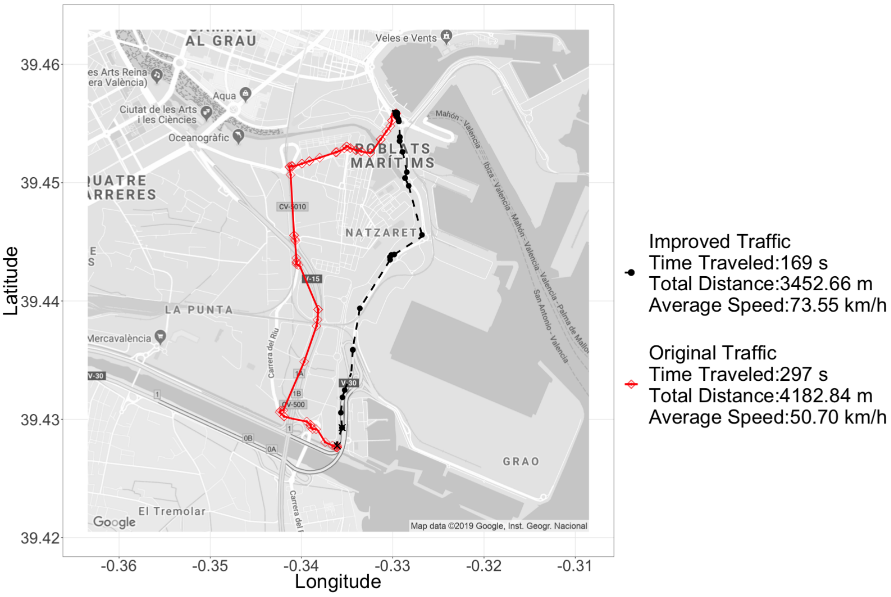
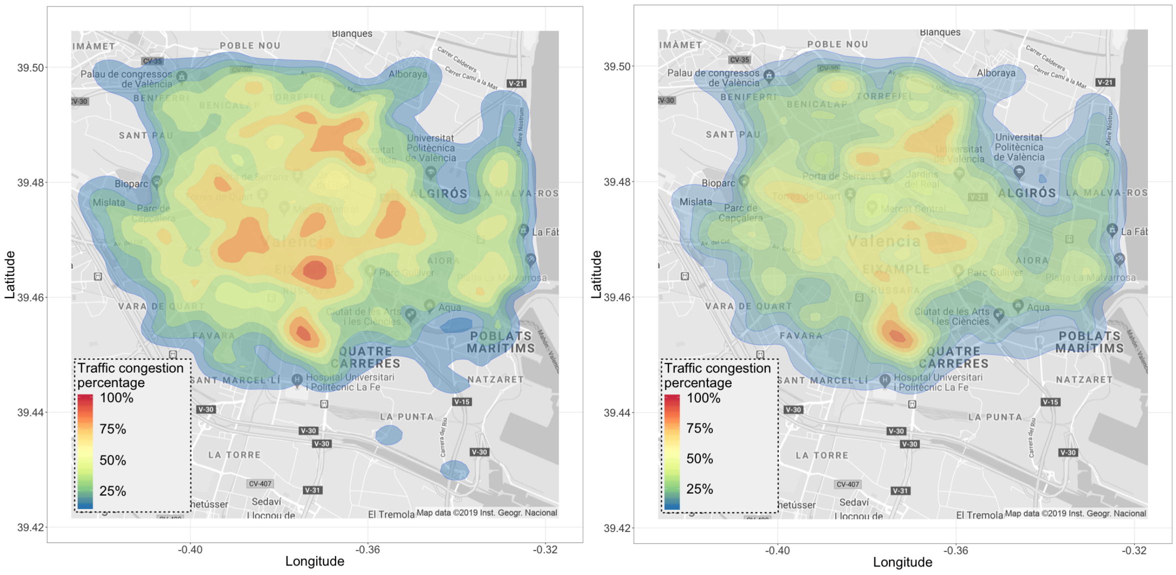


Figure 4. Long routes examples of the city of Valencia.

Figure 5 provides a broader perspective of the areas that tend to present traffic congestion in the city of Valencia, with a tendency towards the red color. Thus, we observe that the reference traffic in Figure 5a has more congested areas; by applying our traffic load-balancing algorithm, the amount of congested areas is reduced, greatly alleviating traffic, as shown in Figure 5b.



(**a**) Original Traffic

(**b**) Improved Traffic

Figure 5. Heatmap of traffic congestion for the city of Valencia.

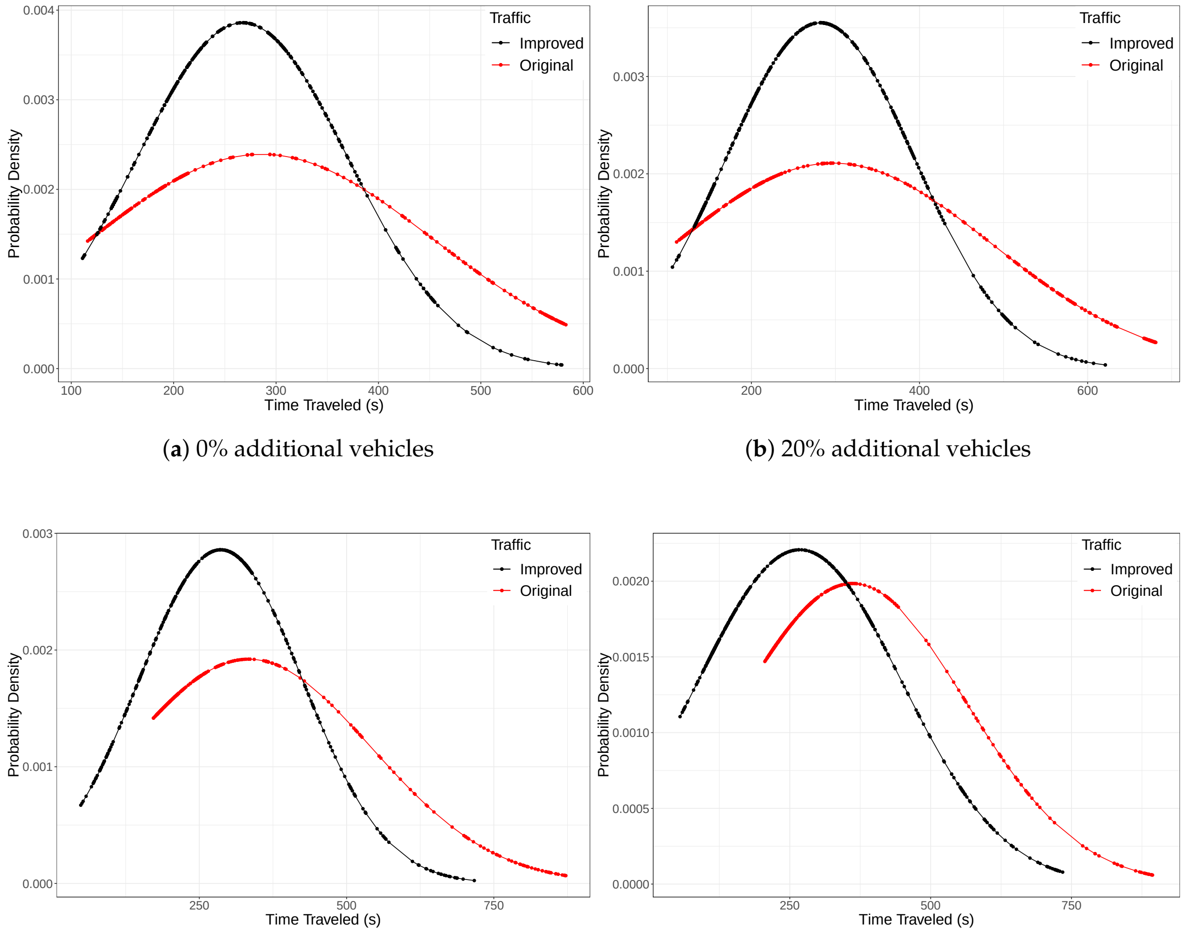
5.2. Hostpot-Based Traffic Load Balancing

For our next set of experiments, we have chosen the Ruzafa neighborhood within the city of Valencia, whose area is 2150 × 500 m, with 300 vehicles moving around during the rush hour. The real trace corresponds to a Monday in November (8:00 a.m.–9:00 a.m.) [2] within a maximum period of 900 s, and whose O-D matrix is within the selected zone. This particular neighborhood offers several alternative routes to gain further insight into the effectiveness of our traffic congestion Equation (1) when used to balance traffic flows and reduce congestion. In addition, we have performed tests by varying the total number of additional vehicles injected in this scenario up to twice the number of vehicles of the real trace. For this purpose we relied on Equation (2), presented in a previous work [6], thus ensuring that the additional load is distributed between the different street segments involved, and we assess the behavior for different numbers of additional vehicles.

$$\tau_{s,n} = \frac{\sigma_s}{\omega_s} \cdot \varphi_n + \lambda \tag{2}$$

In Equation (2), σ_s is the smallest number of vehicles detected among all segments of the street s, variable ω_s is the number of segments that compose street s, φ_n is the traffic flow adjustment factor for each street segment, and n is the number of iterations. In addition, the term λ is used to regulate the number of vehicles injected into the scenario to generate different degrees of congestion, and thus offers the required flexibility to regulate congestion in the target area according to any criteria.

Figure 6 shows the improvements achieved in terms of vehicle travel times. We can see that indeed, our traffic predictions are effective, and that they allow us to significantly reduce the time required for vehicles to reach their destinations. In particular, we observe that the average number of vehicles traveling on the streets within this area falls within a Gauss bell, and that the average travel time improvement achieved is approximately 8% compared to the default travel time behavior (no load balancing).



(**a**) 0% additional vehicles

(**b**) 20% additional vehicles

(**c**) 80% additional vehicles

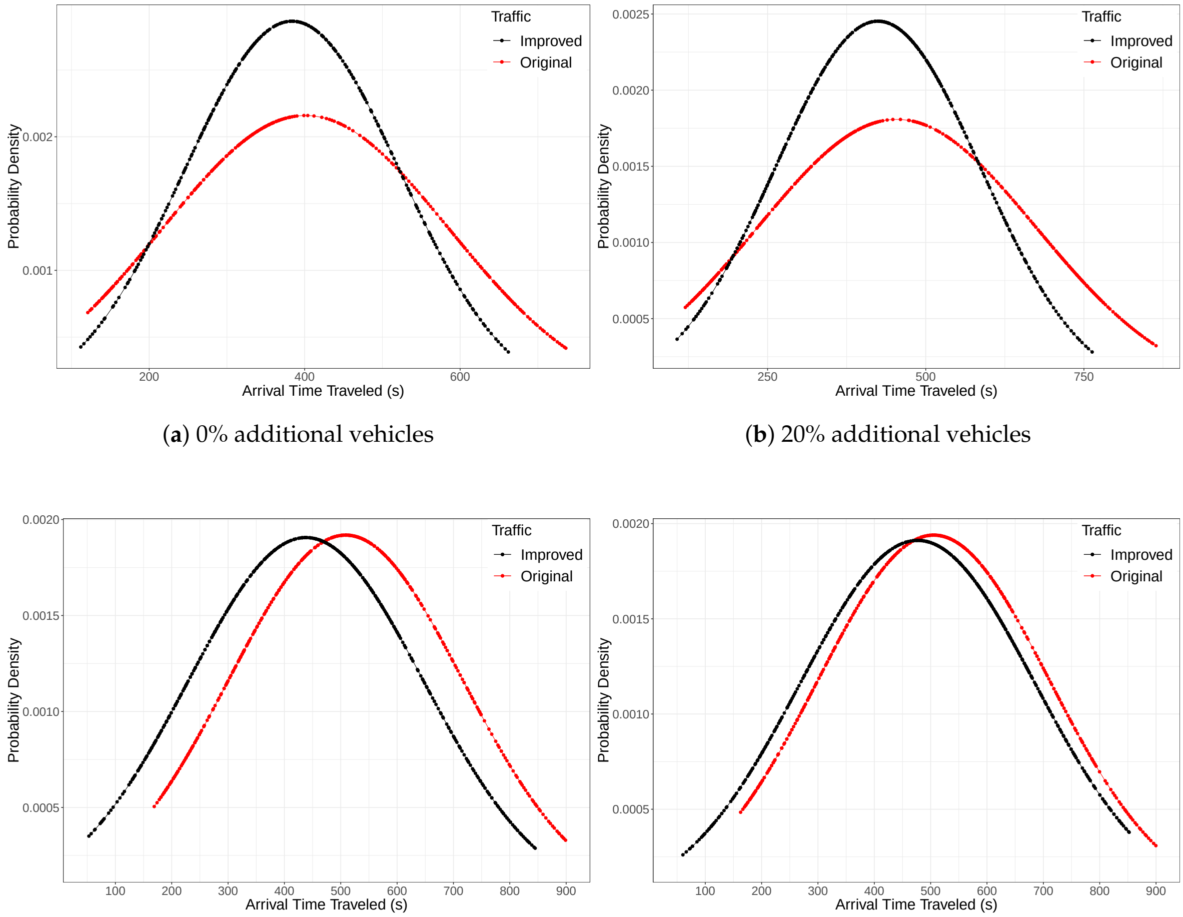
(**d**) 100% additional vehicles

Figure 6. Average Travel Time under different saturation levels.

Likewise, we can observe in Figure 7 that the time of arrival to the destination for the vehicles that benefit from our load-balancing technique is also decreased when compared to the original traffic used as reference, and the standard deviation for this metric is also reduced.

Another parameter providing insight on how the traffic flow is improved is the average vehicle speed within the scenario. Figure 8 shows that vehicles can drive faster in the scenario where our load-balancing strategy is active. In particular, we observe that the normal distribution characterizing vehicular speed is shifted to the right, which indicates that the average speed of the original traffic is slower. Notice that lower vehicle speeds are closely related to higher traffic congestion conditions, impeding vehicles to advance. Therefore, we find that our improved solution achieves 16% more fluid traffic conditions compared to the default, non-balanced situation.

In Figure 9, we can also observe the general statistics when gradually increasing the vehicles in the Ruzafa area. Therefore, we observe in Figure 9a how the average travel time increases when injecting additional vehicles. However, when applying our algorithm, travel times do not increase as abruptly as in the reference scenario, meaning that our algorithm assigns vehicles the best routes, achieving traffic balancing at all times. Consequently, we also observe the same behavior in Figure 9b, since the vehicles injected in the traffic scenario that benefits our load-balancing strategy reach their destination faster than the vehicles in the reference traffic scenario.



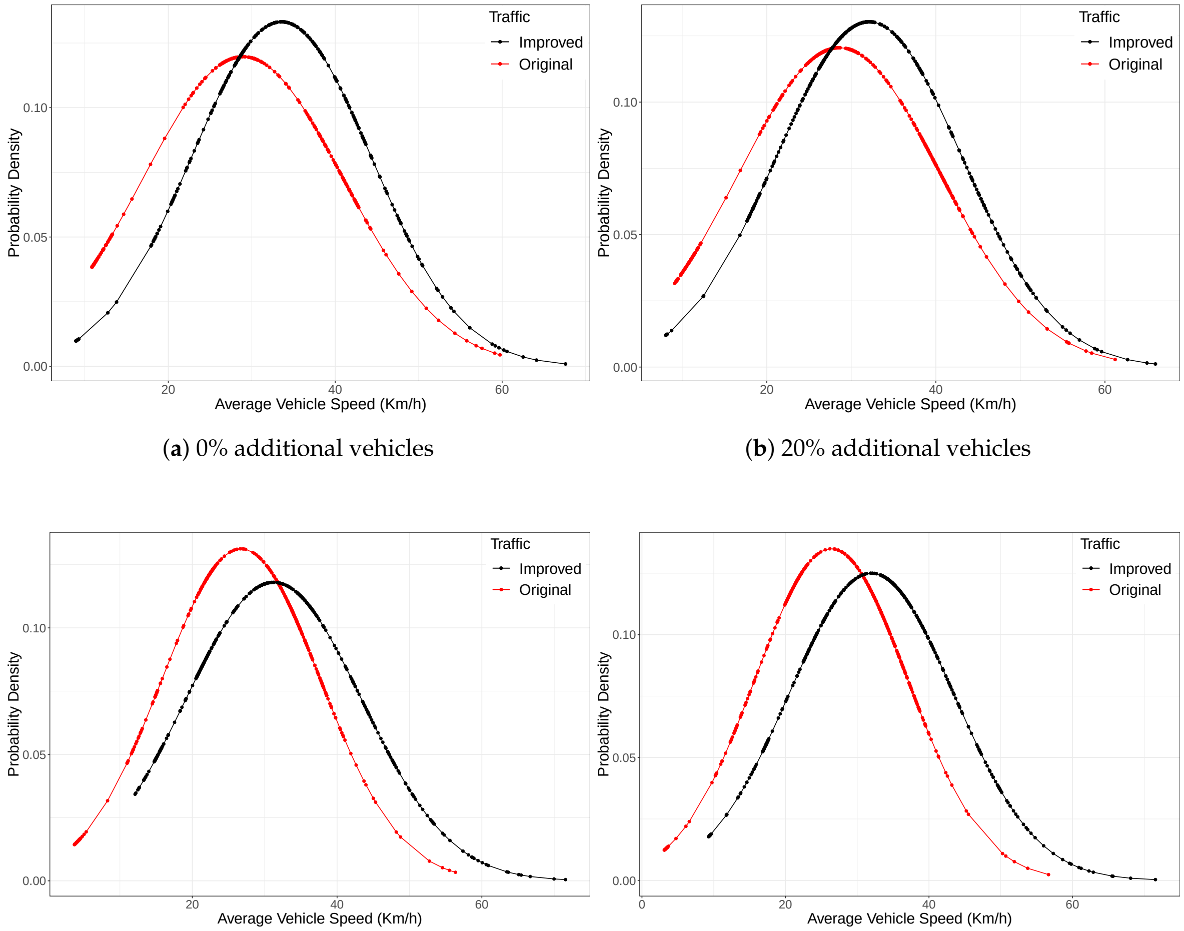
(**a**) 0% additional vehicles

(**b**) 20% additional vehicles

(**c**) 80% additional vehicles

(**d**) 100% additional vehicles

Figure 7. Arrival times for the vehicles under different saturation levels.



(**a**) 0% additional vehicles

(**b**) 20% additional vehicles

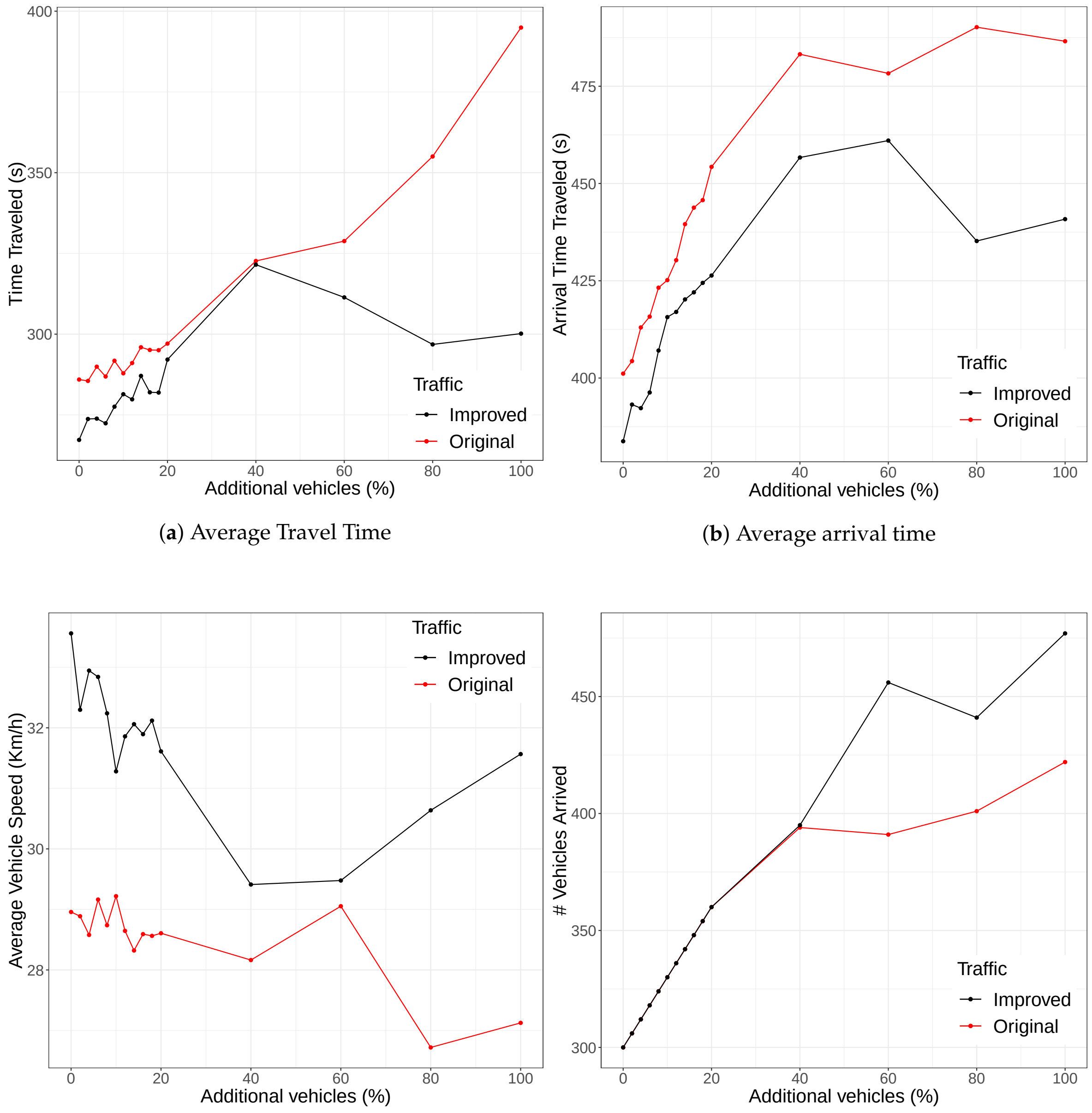
(**c**) 80% additional vehicles

(**d**) 100% additional vehicles

Figure 8. Average vehicle speed under different saturation levels.

In terms of the average vehicle speed when injecting additional vehicles, Figure 9c shows that the speed of the vehicles is reduced, meaning that there are areas where the presence of many vehicles provokes congestion, reducing traffic fluidity. However, when traffic is managed by our route server, speed does not decrease as abruptly as in the default case, demonstrating the effectiveness of our algorithm at regulating traffic, especially in the presence of automated vehicles, as it is able to reduce the areas experiencing traffic congestion. This is also reflected by the number of vehicles able to fully complete their routes, as shown in Figure 9d. By applying our load-balancing algorithm to traffic, more vehicles arrive at their destination because they are assigned alternate routes with less traffic, thereby increasing the chances of completing their route within the simulation period tested.

As stated earlier, our algorithm can analyze different alternative routes to balance traffic, choosing the least congested route through the prediction of travel times depending on the degree of traffic congestion on a per-segment basis by using Equation (1). Thus, we observe in Figure 10 some examples of routes for this scenario, in which the reference traffic (see Figure 10a) always takes a single route to reach the destination without prior knowledge of the vehicular behavior at a given time, thus causing traffic congestion and increasing travel times, generating an unnecessary increase in terms of fuel consumption. However, when relying on our route server (see Figure 10b), our algorithm always offers n alternative routes to balance the traffic by having the knowledge of the predicted congestion conditions. For that reason, it determines the route offering a lower travel time each time an autonomous vehicle requests such information, and such optimal route will be frequently changing depending on updates regarding future traffic occupation.



(**a**) Average Travel Time

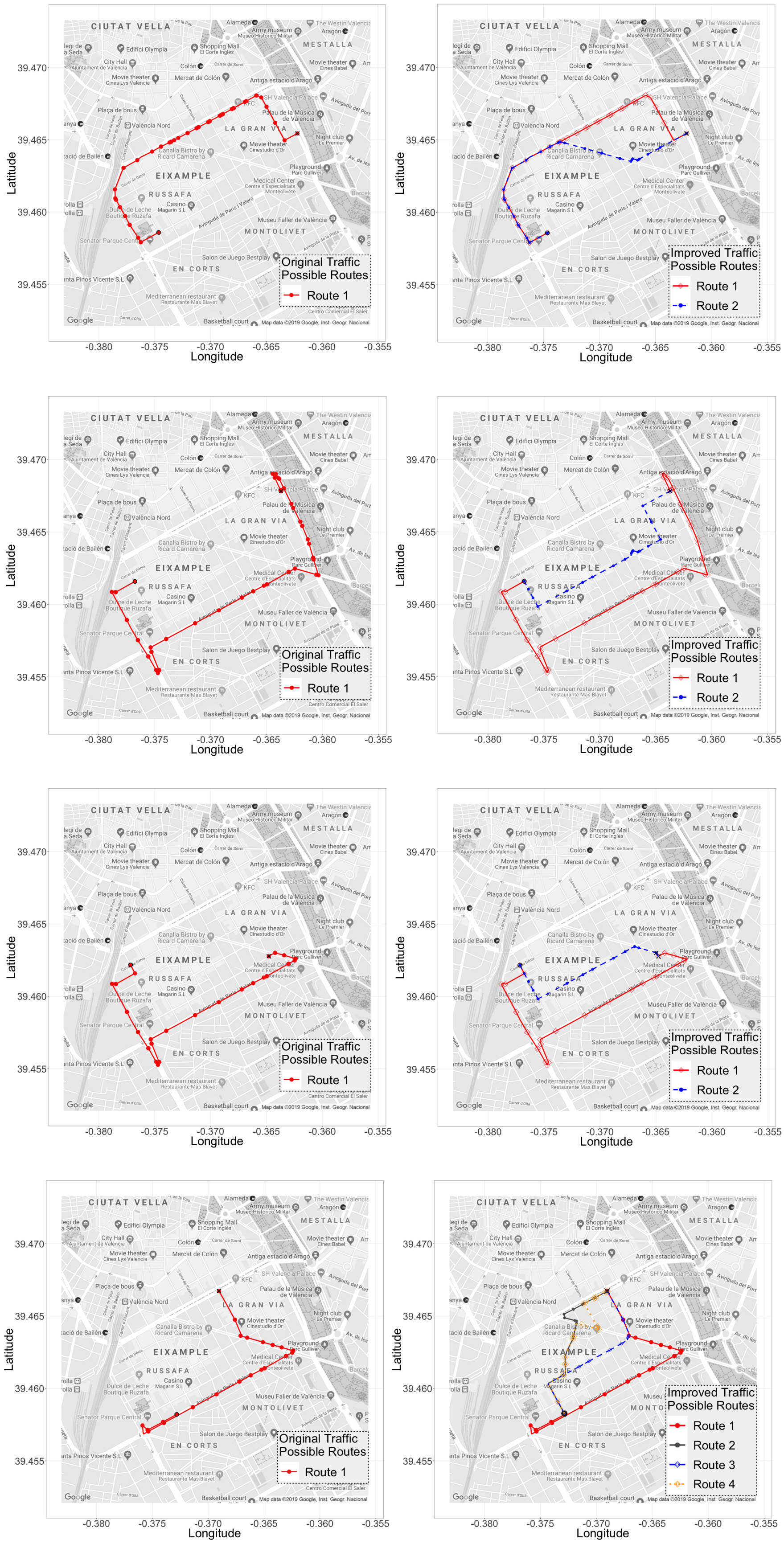
(**b**) Average arrival time

(**c**) Average vehicle speed

(**d**) # vehicles arrived

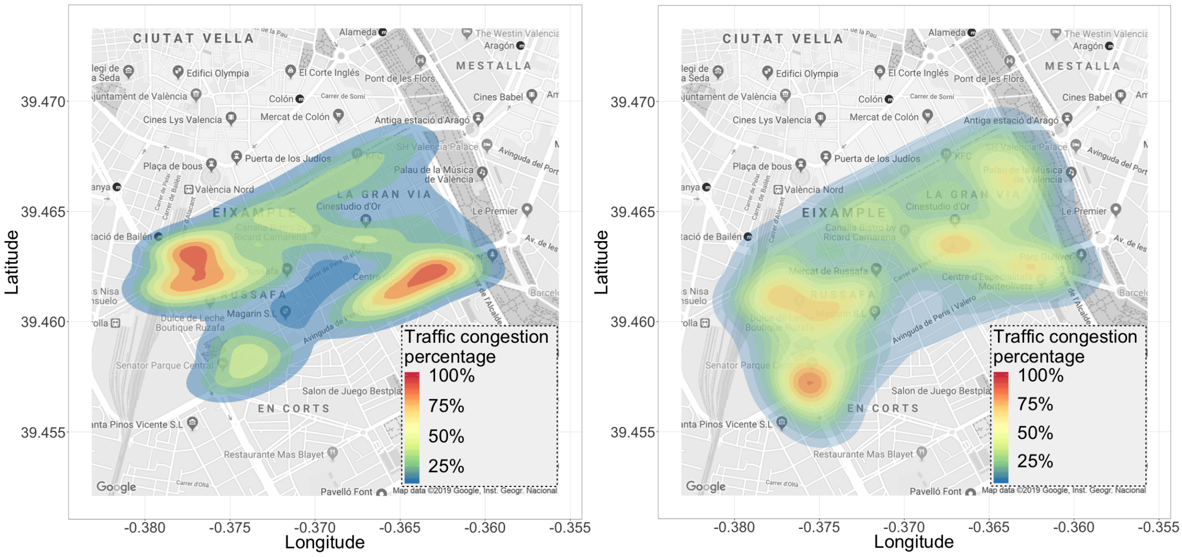
Figure 9. General Statistics at the Ruzafa neighborhood.

To complete our analysis, Figure 11 allows assessing the performance differences between the experiments in terms of heat maps that represent the vehicle density in the selected scenario. Figure 11a shows that the original traffic is distributed inefficiently, failing to occupy all the streets in this area. In fact, we can observe that traffic is mostly concentrated in a single area, where a high accumulation of vehicles provokes traffic congestion; this occurs because only a few routes are used to go from origin to destination, as occurs in real-life situations if relying on a standard navigation tool. Figure 11b shows that our solution generates a different behavior. Now, vehicles are assigned alternative routes when traffic congestion is predicted in the streets along the assigned route. In fact, we can clearly observe that more alternative routes are used to decongest traffic, improving traffic fluidity, and effectively balancing the traffic throughout the scenario. This way, only ephemeral congestion events take place near semaphores.



(**a**) Original traffic routes (**b**) Improved traffic routes

Figure 10. Routes Origin-Destination of the original traffic and our propose.



(**a**) Original Traffic (**b**) Improved Traffic

Figure 11. Heatmap of traffic congestion for the city of Valencia.

6. Conclusions

In the near future, an increasing number of autonomous vehicles is expected to be deployed in our cities. As we move towards this scenario, new requirements in terms of efficient traffic management emerge. In particular, different studies highlight that congestion problems can become more prevalent, as more vehicles use navigation systems to select their routes.

Thus, in this work, we took a first step to address this problem by proposing a centralized traffic manager able to balance the traffic in a city in a seamless and efficient manner. In particular, we started from a realistic traffic model for the city of Valencia developed in a previous work. Then, after analyzing and characterizing the congestion behavior for all streets and avenues in the city, we proposed an equation able to predict traffic congestion by relating travel times with the vehicular load on each street. By integrating the traffic equation characterizing each particular street segment in our ABATIS route server, we were able to accurately predict future traffic conditions in a dynamic manner, thus achieving load balancing.

Experimental results comparing the default traffic flow against our improved solution showed that the latter achieved significant benefits in terms of travel times and average travel speed of vehicles, reducing congestion and improving traffic fluidity. Our proposed solution was validated in large-scale scenarios, such as the city of Valencia, with an area of 77.43 km^2, and where the complexity increases due to the numerous routes that the vehicles requested to our centralized traffic manager. Our proposal improved travel times by 5%, and the average travel speed was 5% higher compared to the reference traffic for the entire city. We also focused on areas facing high congestion levels, such as the Ruzafa neighborhood in the city of Valencia, where we evaluated our proposed solution when injecting additional vehicles into this area to check the limits of our traffic balancing algorithm. In particular, our proposed solution was able to improve travel times by 8% compared to the default traffic conditions even under very high loads. In addition, the average travel speed with our proposed solution was 16% higher, meaning that vehicles arrived faster to their destinations.

As future work, we plan to conduct further experiments to determine the gain achieved by our approach in terms of CO$_2$ emissions and fuel consumption.

Author Contributions: Conceptualization, J.L.Z.-M. and C.T.C.; Formal analysis, J.L.Z.-M. and D.S.; Funding acquisition, J.-C.C.; Investigation, J.L.Z.-M.; Methodology, J.L.Z.-M. and C.T.C.; Project administration, C.T.C.; Resources, J.-C.C. and P.M.; Software, J.L.Z.-M.; Supervision, C.T.C., D.S., J.-C.C., P.M. and T.G.; Validation, J.L.Z.-M., C.T.C., D.S., J.-C.C. and P.M.; Writing—original draft, J.L.Z.-M.; Writing—review & editing, J.L.Z.-M., C.T.C., D.S., L.-G.L.-Z., J.-C.C., P.M. and T.G.

Funding: This work was partially supported by the "Ministerio de Ciencia, Innovación y Universidades, Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad, Proyectos I+D+I 2018", Spain, under Grant RTI2018-096384-B-I00, and by the "Programa de Becas SENESCYT" de la República del Ecuador.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sommer, C.; Krul, R.; German, R.; Dressler, F. Emissions vs. travel time: Simulative evaluation of the environmental impact of ITS. In Proceedings of the 2010 IEEE 71st Vehicular Technology Conference, Taipei, Taiwan, 16–19 May 2010 ; pp. 1–5. [CrossRef]
2. Zambrano, J.L.; Calafate, C.T.; Soler, D.; Cano, J.C.; Manzoni, P. Using real traffic data for its simulation: Procedure and validation. In Proceedings of the 2016 International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/ IoP/SmartWorld), Toulouse, France, 18–21 July 2016; pp. 161–170. [CrossRef]
3. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Cano, J.C.; Manzoni, P. Analysis and Classification of the Vehicular Traffic Distribution in an Urban Area. In *Ad-Hoc, Mobile, and Wireless Networks*; Puliafito, A., Bruneo, D., Distefano, S., Longo, F., Eds.; Springer: Cham, Switzerland, 2017; pp. 121–134.
4. Calafate, C.T.; Soler, D.; Cano, J.C.; Manzoni, P. Traffic management as a service: The traffic flow pattern classification problem. *Math. Probl. Eng.* **2015**, *2015*. [CrossRef]
5. Nguyen, T.V.; Krajzewicz, D.; Fullerton, M.; Nicolay, E. DFROUTER-Estimation of Vehicle Routes from Cross-Section Measurements. In *Modeling Mobility with Open Data*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 3–23. ISBN 978-3-319-15024-6.
6. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Cano, J.C. Towards realistic urban traffic experiments using DFROUTER: Heuristic, validation and extensions. *Sensors* **2017**, *17*, 2921. [CrossRef] [PubMed]
7. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Cano, J.C. Modeling and characterization of traffic flows in urban environments. *Sensors* **2018**, *18*, 2020. [CrossRef] [PubMed]
8. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; WieBner, E. Microscopic Traffic Simulation using SUMO. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
9. Varga, A. OMNeT++. In *Modeling and Tools for network Simulation*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 35–59.
10. Mir, Z.H.; Filali, F. An adaptive Kalman filter based traffic prediction algorithm for urban road network. In Proceedings of the 2016 12th International Conference on Innovations in Information Technology (IIT), Al-Ain, UAE, 28–30 November 2016; pp. 1–6.
11. Abidin, A.F.; Kolberg, M. Towards improved vehicle arrival time prediction in public transportation: Integrating SUMO and Kalman filter models. In Proceedings of the 2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim), Cambridge, UK, 25–27 March 2015; pp. 147–152. [CrossRef]
12. Kumar, S.V. Traffic flow prediction using Kalman filtering technique. *Procedia Eng.* **2017**, *187*, 582–587. [CrossRef]
13. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. C Emerg. Technol.* **2018**, *54*, 187–197. [CrossRef]
14. Tian, Y.; Pan, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 153–158. [CrossRef]
15. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp.* **2014**, *16*, 865–873. [CrossRef]
16. Alonso-Mora, J.; Wallar, A.; Rus, D. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 3583–3590.

17. Weyns, D.; Holvoet, T.; Helleboogh, A. Anticipatory vehicle routing using delegate multi-agent systems. In Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Seattle, WA, USA, 30 September–3 October 2007; pp. 87–93.
18. Zheng, X.; Recker, W.; Chu, L. Optimization of control parameters for adaptive traffic-actuated signal control. *J. Intell. Transp. Syst.* **2010**, *14*, 95–108. [CrossRef]
19. Gartner, N.H.; Pooran, F.J.; Andrews, C.M. Implementation of the OPAC adaptive control strategy in a traffic signal network. In Proceedings of the 2001 IEEE Intelligent Transportation Systems (Cat. No. 01TH8585), Oakland, CA, USA, 25–29 August 2011; pp. 195–200.
20. Lin, S.; De Schutter, B.; Xi, Y.; Hellendoorn, H. Efficient network-wide model-based predictive control for urban traffic networks. *Transp. Res. Part C Emerg. Technol.* **2012**, *24*, 122–140. [CrossRef]
21. Zhang, R.; Rossi, F.; Pavone, M. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. *Auton. Robots* **2018**, *42*, 1427–1442.
22. Agafonov, A.; Myasnikov, V. Efficiency comparison of the routing algorithms used in centralized traffic management systems. *Procedia Eng.* **2017**, *201*, 265–270. [CrossRef]
23. Ren, M.; Khoukhi, L.; Labiod, H.; Zhang, J.; Veque, V. A new mobility-based clustering algorithm for vehicular ad hoc networks (VANETs). In Proceedings of the NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; pp. 1203–1208.
24. Nafi, N.S.; Khan, R.H.; Khan, J.Y.; Gregory, M. A predictive road traffic management system based on vehicular ad-hoc network. In Proceedings of the 2014 Australasian Telecommunication Networks and Applications Conference (ATNAC), Southbank, Australia, 26–28 November 2017; pp. 135–140.
25. Wan, J.; Liu, J.; Shao, Z.; Vasilakos, A.; Imran, M.; Zhou, K. Mobile crowd sensing for traffic prediction in internet of vehicles. *Sensors* **2016**, *16*, 88. [CrossRef] [PubMed]
26. Liebig, T.; Piatkowski, N.; Bockermann, C.; Morik, K. Dynamic route planning with real-time traffic predictions. *Inf. Syst.* **2017**, *64*, 258–265. [CrossRef]
27. Kamal, M.A.S.; Imura, J.I.; Hayakawa, T.; Ohata, A.; Aihara, K. Smart driving of a vehicle using model predictive control for improving traffic flow. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 878–888. [CrossRef]
28. Chen, J.; Li, D.; Zhang, G.; Zhang, X. Localized space-time autoregressive parameters estimation for traffic flow prediction in urban road networks. *Appl. Sci.* **2018**, *8*, 277. [CrossRef]
29. Behrisch, M.; Bieker, L.; Erdmann, J.; Krajzewicz, D. SUMO—Simulation of urban mobility: An overview. In Proceedings of the Third International Conference on Advances in System Simulation. ThinkMind (SIMUL 2011), Barcelona, Spain, 23–28 October 2011; IARIA XPS Press: København, Denmark, 2011.
30. Wegener, A.; Piórkowski, M.; Raya, M.; Hellbrück, H.; Fischer, S.; Hubaux, J.P. TraCI: An interface for coupling road traffic and network simulators. In Proceedings of the 11th Communications and Networking Simulation Symposium, Ottawa, ON, Canada, 14–17 April 2008; pp. 155–163.
31. Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 3–7 March 2008.
32. Lindenberg, B.; Rieken, R.; Rummel, J.; Flötteröd, Y.-P.; Erdmann, J.; Schauer, O.; Parker, H.; Segata, M.; Portilla, C.; Espinosa-Oviedo, J.E.; et al. *SUMO 2017—Towards Simulation for Autonomous Mobility*; Institute of Transportation System (DLR): Berlin-Adlershof, Germany, 2017; Volume 31, p. 192.
33. Wießner, E.; Lücken, L.; Hilbrich, R.; Flötteröd, Y.P.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J. *SUMO 2018—Simulating Autonomous and Intermodal Transport Systems*; Institute of Transportation System (DLR): Berlin-Adlershof, Germany, 2018; Volume 2, p. 217.
34. Nguyen, T.V.; Krajzewicz, D.; Fullerton, M.; Mai, S.T. DFROUTER–Route estimate method based on detector data. In *SUMO2014—Modeling Mobility with Open Data*; Institute of Transportation System (DLR): Berlin-Adlershof, Germany, 2014.
35. Delling, D.; Werneck, R.F. Customizable point-of-interest queries in road networks. *IEEE Trans. Knowl. Data Eng.* **2014**, *27*, 686–698.