Controlling road congestion via a low-complexity route reservation approach

C. Menelaou, P. Kolios, S. Timotheou, C.G. Panayiotou and M.P. Polycarpou[a,1]

[a]*C. Menelaou, P. Kolios, S. Timotheou, C.G. Panayiotou and M.P. Polycarpou are with the KIOS Research Center for Intelligent Systems and Networks, and the Department of Electrical and Computer Engineering, University of Cyprus,* {cmenel02, pkolios, timotheou.stelios, christosp, mpolycar}@ucy.ac.cy

Abstract

This work introduces a novel route reservation architecture to manage road traffic within an urban area. The developed routing architecture decomposes the road infrastructure into slots in the spatial and temporal domains and for every vehicle, it makes the appropriate route reservations to avoid traffic congestion while minimizing the traveling time. Under this architecture, any road segment is admissible to be traversed only during time-slots when the accumulated reservations do not exceed its critical density. A road-side unit keeps track of all reservations which are subsequently used to solve the routing problem for each vehicle. Through this routing mechanism, vehicles can either be delayed at their origin or are routed through longer but non-congested routes such that their traveling time is minimized. In this work, the proposed architecture is presented and the resulting route reservation problem is mathematically formulated. Through a complexity analysis of the routing problem, it is shown that for certain cases, the problem reduces to an NP-complete problem. A heuristic solution to the problem is also proposed and is used to conduct realistic simulations across a particular region of the San Francisco area, demonstrating the promising gains of the proposed solution to alleviate traffic congestion.

Keywords: Route Reservation Architecture, Congestion Avoidance, Traffic Control, Vehicle Routing, Communication Technologies

1. Introduction

Traffic congestion constitutes an ever growing problem in modern cities resulting in multiple adverse effects including driver frustration, environmental pollution while fuel cost and lost of productive hours are the economic downside. The leading cause of congestion is that during certain periods, the number of vehicles that request to simultaneously traverse specific road segments increases to the point where it approaches or even exceeds their critical capacity. This problem does not necessarily occur due to lack of overall network capacity but due to the absence of mechanisms that can achieve an efficient network utilization [1].

Interestingly, offering real-time traffic state information to drivers has shown to create additional side effects to the overall utilization since all rational drivers would try to follow less congested road segments instead of following the shortest distance paths. This selfish behaviour, as demonstrated in [2], gives rise to network state oscillations and exacerbates road congestion. Moreover, the problem becomes even greater when the unpredictability of driving behavior is taken into account. Thus, the objective of this work is to develop a novel architecture which will provide the means to better utilize the network capacity, both spatially and temporally such that the effects of congestion are minimized.

The Macroscopic Fundamental Diagram (MFD) [3] explains the macroscopic relationship between the three main mobility factors, i.e., speed, flow, and density. From studying the MFD two distinct regimes can be identified: 1) the free-flow regime and 2) the congested regime [4] which are separated according to each region's accumulated density. The MFD indicates that an increase in a region's density within the congested regime, results to a decrease in the

[1]This work was partially supported by the European Research Council (ERC) under the ERC Advanced Grant through the FAULT–ADAPTIVE Project.

vehicle speed with the possibility of a gridlock occurrence. On the other hand, the possibility of a gridlock diminishes within the free-flow regime where free-flow speed conditions are observed and both driver and network dynamics are well approximated [5]. These macroscopic relations are also present when autonomous vehicle features are assumed [6]. Furthermore, as indicated in [7], an MFD is well-defined within a homogeneous region (i.e., within the particular region, all links have similar traffic characteristics and insignificant variance across their densities). Guided by the MFD analysis, we know that for every road segment, there exists a critical capacity and while the vehicle density in the segment is below this critical capacity, vehicle flows and speeds are high and more or less predictable. On the other hand, if the critical capacity is reached (congested regime) then capacity is dropped and vehicle flows and speeds become unpredictable. Thus, a key objective of the proposed architecture is to prevent the vehicle density in any road segment to exceed its critical capacity. For the purposes of this paper, we assume that the critical density of each road segment is known (e.g., through the MFD analysis) however, even if these are not known they can be computed through extensive simulations or other tools like perturbation analysis [8].

This work introduces a novel route-reservation architecture and a routing algorithm that utilizes the obtained reservations in order to determine the best possible path subject to avoiding road segments that are expected to be at their capacity. In the proposed architecture, a Road Side Unit (RSU) decomposes the road network spatially and temporarily. Given past requests, the RSU has an estimate of the number of vehicles that are expected in each road segment, during any interval from the current time and into the future. Based on these reservations, the RSU knows which road segments are expected to be below their critical capacity and thus available to more vehicles and which segments are unavailable. When a vehicle is about to begin its journey (or even earlier if "pre-bookings" will be allowed), it sends a request to the RSU with its origin and destination. The RSU then computes the best possible path for the vehicle such that any road segment that is near its critical capacity is avoided and taking into consideration that it may be best for a vehicle to wait at the origin until certain road segments become available. Once the RSU determines the best path, it assumes the vehicle will move with the speed at capacity of each segment in order to update the number of vehicles in each segment and each future time slot.

Note that the proposed architecture, has a number of benefits that are worth emphasizing. Obviously vehicles are routed through non-congested paths which is a benefit for the individual vehicle. In addition, by not allowing vehicles to go through segments that are near their capacity, it "protects" other vehicles that have already reserved those segments and it guarantees that they will not experience congestion either, thus the approach has also a more social benefit. Finally, by allowing the RSU to suggest delayed departures, it keeps vehicles and their drivers away from the road minimizing their travel time and the cost associated with lost productivity and environmental impact. On the other hand, the proposed approach admittedly faces certain implementation challenges. First is the communication and computation aspects involved in the implementation. Given the recent developments in the information and communication domain, the Internet of Things (IoT) technology and the proliferation of connected vehicles, these challenges will be addressed in the near future. We also point out that distributed versions of the architecture which are more scalable are feasible and is the topic of our current research. Another major challenge is driver compliance. This challenge can be easily addressed in the context of autonomous vehicles by enforcing that the vehicles will follow the paths provided by the RSU. Even in the case of human drivers, there are some possible solution by monitoring the actual path followed (e.g., using the vehicle's GPS) and by providing incentives to compliant drivers or penalties to non-compliant ones.

Once the RSU receives a request for finding a route for a vehicle, it needs to solve the routing problem and the objective is to determine the path that will allow the vehicle to reach its destination at the earliest possible time while avoiding unavailable segments and possibly delaying its departure. It turns out, that this is a difficult problem to solve and as demonstrated in the sequel certain instances of the problem are NP-complete. Thus, some heuristic algorithms are needed to solve this problem one of which is proposed in this work and is compared to other approaches through extensive simulation.

In summary, the contribution of this paper includes a reservation-based novel architecture for achieving congestion free routing, in the context of Intelligent Transportation Systems (ITS). Given the obtained reservations, the vehicle routing problem is formulated and is shown to be NP-complete while an efficient heuristic algorithm is proposed that provides good solutions. Furthermore, the benefits of the entire approach are demonstrated through extensive simulations on realistic networks.

The rest of the paper is organized as follows. Section 2 describes the related work, while Section 3 introduces the architecture of the proposed solution. Section 4 mathematically formulates the proposed route reservation problem and performs a complexity analysis of the problem. Section 5 presents a solution approach based on "Time Expansion" while Section 6 presents a heuristic greedy solution based on Dijkstra's algorithm called Route Reservation Algorithm (RRA). Extensive simulation results are included in Section 7 that demonstrate the benefits of the proposed solution in realistic scenarios. Finally Section 8 concludes this work.

2. Related Work

Currently, the *gating* and *perimeter control* methods constitute the state-of-the-art solutions for addressing the traffic congestion problem. Both approaches, implement boundary flow control mechanisms (e.g., street closures, pricing, traffic flow metering, traffic light signalling) to restrict the input flow around a particular homogeneous [2] region [10]. *Gating control* controls the volume of traffic within a homogeneous region by restricting external flows from entering in a region if the region's critical density has been reached [11] [12]. More specifically, gating uses a reduced MFD which was constructed using real-time measurements and thus avoids the extensive amount of data required for the identification of each region's MFD. Similarly, the two-level perimeter-and-boundary control is applied in multi-region networks to regulate traffic exchange between regions and the outside world [13] [9]. At the first level, an urban area is clustered into inter-connected homogeneous regions to ensure modelling accuracy within the macroscopic relations. At the second-level, similar to gating, vehicles are allowed to enter within a region only if its critical density has not been reached [14]. By discriminating between different areas of the network based on the homogeneity of the region more accurate decisions can be made. Furthermore, control decisions at the macroscopic level are simpler to implement since they do not require extensive traffic information (e.g., the per-link densities, speeds and flows) making these approaches computationally efficient [9], [14]. Perimeter and gating control however, do not take any control action for the endogenous flows thus traffic congestion may occur due to the flows of vehicles generated *within* the region [5]. The proposed route-reservation architecture does not distinguish between endogenous and exogenous flows and applies a more targeted control on all vehicles preventing the formation of long queues and excessive delays. Furthermore, the aforementioned approaches do not have a reliable mechanism for predicting the future state of the network, which is something achieved through the proposed reservation architecture.

Recent literature also considers routing techniques using online or offline network estimates [25]. Both static and stochastic models have been used to guide routing decisions based on online predictions of the travel times and speed conditions for each vehicle, [20, 21]. Scheduling decisions usually consider these network states to obtain shortest-travel-time paths, but neglect the negative effects that may occur when the selected road segments become congested [19]. Another important routing control approach is the Decreasing Order of Time (DOT) algorithm [23], which efficiently finds the time-dependent shortest path (using travel-time) within a user-chosen time window. Despite the fact that these approaches are of particular interest to ITS applications, most of them do not consider the unpredictability of driver behavior that is observed especially in the congested regime [5]. Correspondingly, operating in the congested regime can result in up to 15% less capacity for the specific road segment which further increases travel time, a phenomenon called *capacity drop*. Influencing the driver behavior using enroute advisory systems can affect the MFD shape and reduce capacity drop (by shrinking the hysterisis loop), but does not eliminate the problem [15].

An aggregated and approximate dynamic traffic assignment model, for establishing regional routing, is introduced in [17]. This work, performs a dynamic traffic assignment that incorporates the MFD dynamics to establish stochastic user equilibrium conditions, illustrating that such an approach outperforms other traffic assignment solutions. By the same token, [18] introduced a route choice strategy that aims to reduce congestion within urban areas assuming aggregated regional and partially known sub-regional information.

It should be noted here that the proposed methodology differs from the current state-of-the-art approaches on time-depended routing, as the latter only change the link costs according to the route density without restricting the inflow. Other approaches, such as [24], incorporate link based piecewise constant speed patterns that are calculated based on historic data using the Flow Speed Model (FSM) (for example, in a working day, during rush hour 7am to 9am the speed is 3 m/s, and at other times of the day the speed is 10 m/s). Evidently, such kind of approaches require to store a huge amount of information to identify the per link speed patterns and may require infrastructure

[2]Homogeneity describes the uniform spatial distribution of vehicles. Furthermore, recent literature indicates that an urban area can be described according to multiple MFDs of homogeneous regions [9].

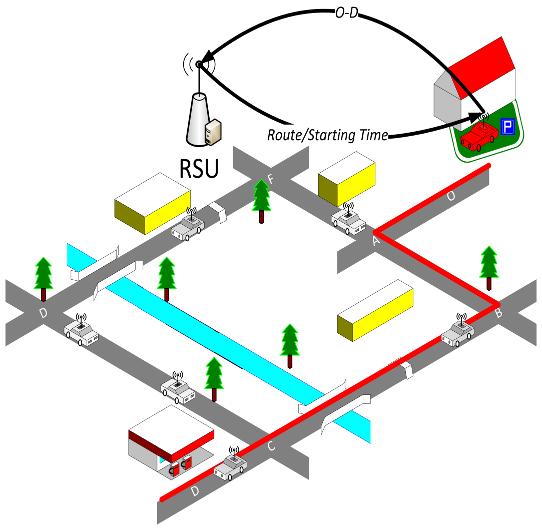

Figure 1: Proposed Architecture.

investment to support the ability to acquire and process such information (huge number of sensors required). Even if such information is collected, the identification of per link speeds patterns in real-time would be a tedious procedure. Instead, the proposed route-reservation approach uses the aggregate network information provided by each region's MFD to efficiently approximate the travel time of vehicles through the network [16]; in this way, the proposed solution allows for a simple and ease to implement solution, as it relies on macroscopic parameters. Moreover, as network density is maintained below the critical capacity, reservations provide accurate and robust state estimation without requiring real-time sensory information.

The key to efficient network utilization is to prevent network's region from getting into congested regimes altogether. Hence, the proposed route-reservation architecture achieves a congestion-free operation by limiting the maximum admissible capacity of each road segment. The route reservations are utilized by computing time-depended shortest path routes between an origin-destination pair varying the road segments travel-time costs (admissibility states). When the travel-time costs vary with time the problem may become NP-complete [26]. This is also true for some cases in which the travel-time cost is not time varying [27]. The NP-compleness of the route-reservatio algorithm is first discussed in [28] while this work provides a detailed complexity analysis.

The proposed solution has been conceived based on time-slot reservation models used to solve the ground holding problem for Air Traffic Management and Control Systems (ATM/ATC) since, airport utilization increases while runway capacity remains constant [29]. In favor of avoiding congestion, the ATM/ATC systems divide the runway capacity both in space and time and a time-slot reservation mechanism is used to improve the overall runway efficiency [30] [31]. This concept is introduced in [32] for road transportation networks while this work elaborates on the architecture and the performance of the route reservation algorithm under realistic settings and over actual network topologies.

3. Route Reservation Architecture

The objective of the proposed route reservation architecture is to efficiently control all vehicle movements over the entire region. To achieve this, the network is decomposed into road sections and associated with each road section is a series of time slots starting from the current time into the future. For each time slot, the RSU keeps an estimate of the number of vehicles that are expected to be traversing the road section during the interval. As a vehicle plans to start its journey, it sends a request to the RSU in order to obtain a path from its current location (i.e., its origin) to the required destination. Given the current reservation state, the RSU determines the best possible path for the vehicle such that it will arrive at its destination at the earliest possible time while avoiding road segments that are expected to be at capacity at the time when the vehicle is expected to traverse the segment. Thus, the RSU responds to the vehicle request, giving the starting time of the journey and the route that the vehicle should follow (as indicated with the red line in Fig. 1).

At the same time, the RSU updates the reservation state of each road segment at the time-slot when the vehicle is expected to traverse the road segment assuming the vehicle will be traveling with a constant speed. Assuming the region's MFD is available, then one can use either the free flow speed or the speed at capacity to also account for some possible delays (in this work we are using the speed at capacity). If the MFD is not available, then the speed to be used can be obtained from historical data. At this point it is worth pointing out that it is unrealistic to expect that all vehicles will actually travel at the same constant speed, thus in practice, it is expected that there will be significant deviations. Despite these deviations, our simulation results indicate that the whole approach still works well and is robust with respect to such inaccuracies.

In order to compute its response, the RSU formulates and solves the routing problem as indicated in the subsequent sections.

4. Problem Formulation

A region of the road network is considered as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V}$, $N_V = |\mathcal{V}|$, representing the road junctions and edges $\mathcal{E}$, $N_E = |\mathcal{E}|$, representing the road segments. We assume that all vehicle movements are within a homogeneous region with MFD parameters ρ^C, ρ^J, v_c and v_f, representing the critical density corresponding to the maximum flow, jam density, speed at capacity and free-flow speed, respectively. Each road segment $(i, j) \in \mathcal{E}$, $\{i, j\} \in \mathcal{V}$ is described by parameters λ_{ij} being the road segment length, N_{ij} representing the number of lanes, $\rho_{ij}(t)$ the road segment instantaneous density, and ρ^J_{ij} representing jam density. Parameter $\rho^C_{ij} = (\rho^C/\rho^J)\rho^J_{ij}$ denotes the critical density of road segment (i, j) which represents the density at which maximum flow rate can be achieved (maximum admissible density) for the specific road segment. Furthermore, parameter d_j denotes the vehicle's earliest arrival time at junction j.

Congestion free road segments are those segments for which $\rho_{ij}(t) \leq \rho^C_{ij}$ so that the vehicles can be assumed to travel with speed-at-capacity v_c. The speed at capacity assumption is used instead of the free-flow speed so that travel time estimates account for the possible delays due to driver imperfection and the delays observed across non-priority road junctions. We assume that, the time is quantized into time-slots of duration T so that the number of time-slots required to traverse road segment (i, j) is $\tau_{ij} = \lfloor \lambda_{ij}/u_c/T \rceil$, where $\lfloor z \rceil$, is the nearest integer to z.

Let variable $n_{ij}(t)$ denote the accumulated number of vehicle reservations of road segment (i, j) for time-slot t. A road segment (i, j) is denoted as admissible if a vehicle entering road junction i at time t can traverse segment (i, j) without making the accumulated reserved density larger than the critical density during any time-slot for which the vehicle will travel on the segment. Thus, the quantity $n_{ij}(t)/(\lambda_{ij}N_{ij})$ denotes the instantaneous density of road segment (i, j) at time t. The *admissibility state* of road segment (i, j) at time-slot t denoted by $x_{ij}(t)$, attains $x_{ij}(t) = 1$ if road segment (i, j) is admissible and $x_{ij}(t) = 0$, otherwise. Mathematically, $x_{ij}(t)$ can be defined as follows:

$$x_{ij}(t) = \begin{cases} 1, \text{ if } n_{ij}(t+k)/(\lambda_{ij}N_{ij}) \leq \rho^C_{ij}, \ \forall \ k = 0, \dots, \tau_{ij} \\ 0, \text{ otherwise} \end{cases} \tag{1}$$

Vehicles are allowed to traverse road segments during time-slots where $x_{ij}(t) = 1$ since the RSU can make any reservations along those time-slots to enable congestion-free routing. To avoid non-admissible road segments, a vehicle may wait at its origin until a non-congested path becomes available (i.e., until all segments of the selected path become admissible), or it can choose alternative admissible road segments. Clearly, a combination of the two aforementioned options may be employed i.e., wait for a short period of time at O and then follow an alternative route.

Considering the above notation, the cost of traversing a road segment $c_{ij}(t)$ can mathematically be expressed as follows:

$$c_{ij}(t) = \begin{cases} \tau_{ij}, \text{ if } x_{ij}(t) = 1, i \neq O \\ \tau_{ij} + w, \text{ if } x_{ij}(t) = 0, i = O \\ \infty, \text{ if } x_{ij}(t) = 0, i \neq O \end{cases} \tag{2}$$

where, w denotes the number of time-slots that a vehicle may wait at the origin such that the path found to traverse from O to destination D is admissible. Given the above definitions, below we define the path finding problem that the RSU is required to solve for every vehicle.

Earliest Arrival Time at Destination (EATD) problem. Given an origin-destination ($O - D$) pair, the time-stamp t_0 at which the routing request is made, and the reservation states $x_{ij}(t)$, $(i, j) \in \mathcal{E}$, $\forall t \geq t_0$, then the EATD problem requests the earliest-arrival-time-at-destination (from O to D). Let p_h denote the h-th path from origin (O) to destination (D) denoted as $p_h = (v_0^h, v_1^h),\ (v_1^h, v_2^h),\ (v_2^h, v_3^h), \cdots (v_{L_h-1}^h, v_{L_h}^h)$, where $v_j^h \in \mathcal{V}$ is the j-th visited vertex in the h-th path, with $v_0^h = O$, $v_{L_h}^h = D$, and L_h is the length of the path p_h in terms of the number of hops. Additionally, let w and $d_{v_j}^h$ denote the waiting time at the origin and the earliest arrival time at junction v_j (assuming the vehicle was delayed by w at the origin), respectively. Then, the earliest arrival time to each vertex of the path can be expressed as:

$$\begin{aligned} d_{v_0^h}^h &= t_0 \\ d_{v_1^h}^h &= d_{v_0^h}^h + c_{v_0^h, v_1^h}(d_{v_0^h}^h) \\ &\vdots \\ d_{v_{L_h}^h}^h &= d_{v_{L_h-1}^h}^h + c_{v_{L_h-1}^h, v_{L_h}^h}(d_{v_{L_h}^h}^h) \end{aligned} \tag{3}$$

Hence, the EATD problem can be expressed in compact form as:

$$(\Pi)\ d_D^* = \min_{w,\ p_h} d_D^h \tag{4}$$
$$\text{s.t. Constraints (2) – (3)are satisfied.}$$

4.1. EATD Complexity Analysis

At a first glance, the formulated *EATD* problem looks similar to the well investigated time-dependent route planning problem [33]. Nevertheless, the *EATD* problem differs from the time-dependent route planning problem since *EATD* introduces road segments with infinite cost (non-admissible road segments) and also allows for waiting intervals that may be observed at the originating junction.

In this section, we investigate the complexity of the *EATD* problem and we prove that it is an NP-complete problem. Let (*EATD*) problem (4) also be denoted as (Π). The complexity analysis of (Π) requires to examine the complexity of two variations of the particular problem, that we denote as the (Π_{AW}) and (Π_{NW}) problems. (Π_{AW}) has a similar objective function as (Π) but it allows vehicles to wait at all road junctions until they become available. Clearly the solution to this problem is not implementable since physically there is not space for vehicles to park and wait until a road section becomes available, however, the solution to this problem can serve as a lower bound to the solution of (Π) while (as we will show) it can be solved in polynomial time. The other related problem is the (Π_{NW}) that does not allow for vehicle waiting neither at the origin nor at any other junction.

Starting from the (Π_{AW}) problem, the cost $c_{ij}(t)$ of traversing road segment (i, j) can mathematically be expressed as:

$$c_{ij}(t) = \begin{cases} \tau_{ij},\ \text{if } x_{ij}(t) = 1 \\ \tau_{ij} + w_{ij},\ \text{if } x_{ij}(t) = 0 \end{cases} \tag{5}$$

where, w_{ij} denotes the number of time-slots that a vehicle may wait at i such that the path, found to traverse from O to destination D, is admissible. Thus, (Π_{AW}) can mathematically be expressed as:

$$(\Pi_{AW})\ \ d_{D_{AW}}^* = \min_{w_{ij} \geq 0,\ p_h} d_D^h \tag{6}$$
$$\text{s.t. Constraints (3), (5)are satisfied.}$$

The cost $c_{ij}(t)$ of traversing a road segment for problem (Π_{NW}) can mathematically be stated as follows:

$$c_{ij}(t) = \begin{cases} \tau_{ij},\ \text{if } x_{ij}(t) = 1 \\ \infty,\ \text{if } x_{ij}(t) = 0 \end{cases} \tag{7}$$

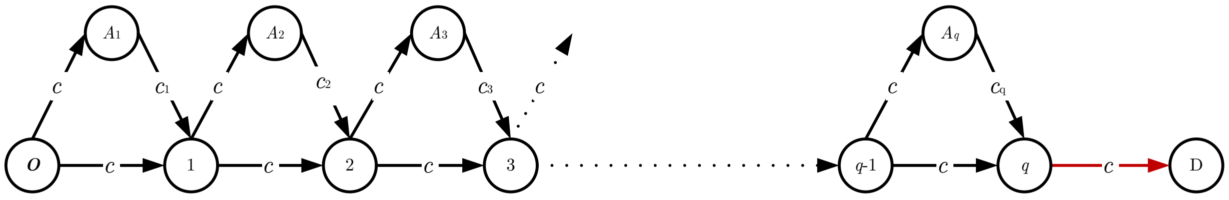

Figure 2: Special case of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (with edge (q, D) attain to non admissible state).

Hence, (Π_{NW}) can be mathematically stated as follows:

$$(\Pi_{NW}) \;\; d^*_{D_{NW}} = \min_{p_h} d^h_D \qquad \text{s.t. Constraints (3), (7)are satisfied.} \tag{8}$$

The two aforementioned variants (Π_{AW}) and (Π_{NW}) are used to prove that the (Π) problem can be categorized as an NP-complete problem. The NP-completeness of (Π) is derived using the *restriction method* [34]. The restriction method requires that problem (Π) is reduced as a special case to a known NP-complete problem.

The examined proof reduces the (Π) problem to the *Number Partitioning Problem* (Π') (described in [34]) which is defined as follows.

Number Partitioning Problem:. Let the set $\mathcal{A}$ consist of n integer numbers $\mathcal{A} = \{a_1, a_2, ..., a_n\}$, $a_j \in \mathbb{Z}^+$ and let an integer number $b \in \mathbb{Z}^+$. (Π') requires to identify the subset $\mathcal{A}'$ where $\mathcal{A}' \subseteq \mathcal{A}$, such that the sum of the numbers in $\mathcal{A}'$ is equal to a given number b. Equivalently, this problem can be expressed using variables $y_j = \{0, 1\}$ that indicate whether a_j is in $\mathcal{A}'$ $(y_j = 1)$ or not $(y_j = 0)$, as follows:

$$\sum_{j=1}^{n} a_j y_j = b, \text{ where } y_j = \{0, 1\} \tag{9}$$

Note that problem (Π') is an NP-complete problem [34].

Lemma 1. (Π_{NW}) *is an NP-complete problem in the case where at least one road segment attains a non-admissible state.*

Proof. To prove Lemma 1 we need to show that (Π_{NW}), can be reduced to a special case of (Π'). For this purpose a special case of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed as shown in Fig. 2. As illustrated in Fig. 2, the traversal cost of each road segment is defined by c and c_j values which are predefined integer constants, i.e., $c \neq c_j$ and $c_j \neq c_k$. Considering the structure of the graph, the cost to traverse the edge from node j to node $j+1$, (i.e., $\hat{c}_{j,j+1}$) can mathematically be stated as:

$$\hat{c}_{j,j+1} = \begin{cases} c + c_{j+1}, & \text{if path passes from } A_{j+1} \\ c, & \text{otherwise} \end{cases} \tag{10}$$

Let (q, D) (indicated with red color) be the single edge on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that attains a non-admissible state as follows:

$$x_{qD}(t) = \begin{cases} 1, & \text{for } t = cq + b \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

while all other edges always attain an admissible state. According to constraints (7), (10) and (11) the traversal cost $c_{qD}(t)$ of (q, D) can be expressed as follows:

$$c_{qD}(t) = \begin{cases} \tau_{qD}, & \text{for } t = cq + b \\ \infty, & \text{otherwise} \end{cases} \tag{12}$$

where, $\tau_{qD} = c$ as indicated in Fig. 2.

According to the above setup, the only possible solution consists of a path from O to D, (i.e., p) where, the arrival time at node q is exactly equal to $cq + b$. Therefore, the arrival time at junction q must be:

$$d_q = cq + b \tag{13}$$

Let, p contain the subpath p' from vertices O to q. There are in total 2^q possible combinations that can constitute subpath p' and the total travel time of each combinations (i.e., c_{Oq}) can be defined as follows:

$$c_{Oq} = \sum_{j=1}^{q} \hat{c}_{j-1,j} y_j = \sum_{j=1}^{q} c(1 - y_j) + \sum_{j=1}^{q} (c + c_j) y_j = cq + \sum_{j=1}^{q} c_j y_j \text{ where,} \tag{14}$$

$$y_j = \begin{cases} 1, & \text{if path passes from } A_j \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Considering Eq. (10), cq time-slots can be provided from all of the 2^q paths while the remaining b time-slots must be identified by the summation of $\sum_{j=1}^{q} c_j y_j$. Therefore, the solution returned according to the selected y_j values, provides a solution to the number partitioning problem since a subset of values (that sum up exactly to b) is required to be selected from the range of c_j, and this completes the proof. □

The second variant assumes that waiting intervals are allowed at all road junctions. This assumption is not feasible along real transportation networks due to lack of adequate buffering space where vehicles will wait. Nonetheless, this case can be considered as a lower bound solution and is part of the subsequent proof of theorem 1 used to prove that $EATD$ can be reduced to (Π') as a special case.

Lemma 2. *The problem (Π_{AW}), i.e., finding the earliest arrival time while waiting at every junction is allowed, can be solved in polynomial time.*

Proof. In the case when a vehicle can wait at all intersections, the problem becomes significantly easier and can be solved to optimality using a simple modification of the Dijkstra's shortest path algorithm [35] which is a polynomial algorithm. Specifically, given an arbitrary graph, at every step of the algorithm, given the time of the earliest vehicle arrival at any node p (through the previous steps of the algorithm), if the next link (p, q) is unavailable until u time units later, its cost $c_{p,q}$ is simply undated to $c_{p,q} + u$, while if the vehicle's earliest arrival at p is during a time period when the link is available, then its cost is simply $c_{p,q}$ which corresponds to the time needed to traverse the link. A detailed correctness proof can be shown using the Dijkstra's proof of correctness based on the contradiction method [36]. □

The third case completes the complexity analysis of the formulated $EATD$ (Π) problem as a combination of the two previous cases of problems (Π_{AW}) and (Π_{NW}).

Theorem 1. *The problem (*Π*), i.e., vehicles are only allowed to wait at the origin O, is an NP-complete problem when more than one road-segments become non-admissible during certain time-slots.*

The proof of theorem 1 is divided into two special cases. This distinction is required in order to find a special case in which (Π) can be stated as an NP-complete problem. The first case illustrates the situation where (Π) can always be solved in polynomial time and the second case covers the scenario where the problem (Π) can be reduced to the Number Partitioning Problem.

Special Cases 1

Considering Theorem 1 and assuming that only one road segment (which should be a part of the path) has to attain a non-admissible reservation state, then (Π) can be solved in polynomial time.

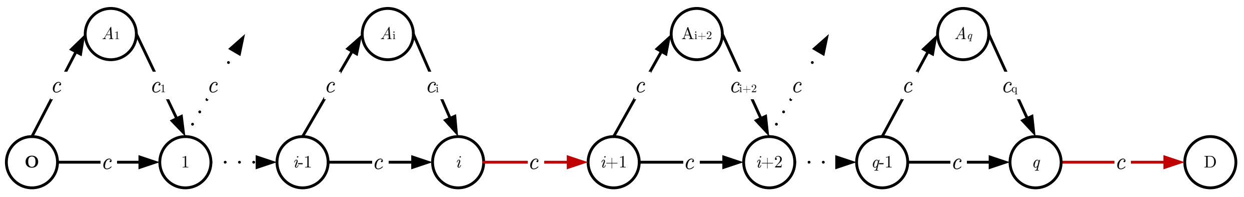

Figure 3: Special case of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (with edges $(i, i+1)$ and (q, D) attain to non admissible state)

Proof. Consider the same example of lemma 1 (shown in Fig. 2) illustrating edge (q, D) that attains a non-admissible reservation state. The solution requires a vehicle to depart from node q exactly at time $cq + b$ as indicated by equality constraint (13). When only one road segment attains a non-admissible state, the problem can be adequately expressed through Lemma 2, since (13) can be transformed to an inequality constraint. As shown in Lemma 2, a solution can easily be found with a feasible path from vertex O to q where $d_q \leqslant cq + b$ according to constraint. If the solution results in arriving at q on an earlier time then the vehicle can wait for the remaining time-slots to the originating junction to satisfy constraint (13). □

Special Case 2

Considering Theorem 1 and assuming that more that one road segments attain a non-admissible state during certain time-slots (which should be a part of the path) then (Π) results to an NP-complete problem.

Proof. Consider the special case in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as shown in Fig. 3 where the cost to traverse the link from node i to node $i+1$, is based according to eq. (10) (i.e, $\hat{c}_{i,i+1}$). Fig. 3 indicates that, in total, two road segments attain a non-admissible state (i.e., edges $(i, i+1)$ and (q, D)) as follows:

$$x_{i,i+1}(t) = \begin{cases} 1, \text{for } t = ci + b_1 \\ 0, \text{ otherwise} \end{cases} \tag{16}$$

$$x_{qD}(t) = \begin{cases} 1, \text{for } t = cq + b_1 + b_2 \\ 0, \text{ otherwise} \end{cases} \tag{17}$$

while all other links always attain an admissible state. According to constraints (2), (16), (17), the traversal cost of both links can be expressed as follows:

$$c_{ii+1}(t) = \begin{cases} \tau_{ii+1}, \text{ for } t = ci + b_1 \\ \infty, \text{otherwise} \end{cases} \tag{18}$$

$$c_{qD}(t) = \begin{cases} \tau_{qD}, \text{ for } t = cq + b_1 + b_2 \\ \infty, \text{otherwise} \end{cases} \tag{19}$$

where, $\tau_{i,i+1} = c$ and $\tau_{qd} = c$ as indicated in Fig. 3.

According to Fig. 3, (Π) has a feasible solution only if an admissible $O - D$ path exists while the departure times at node i and q should be exactly at time-slots $ci + b_1$ and $cq + b_1 + b_2$, respectively. Following the analysis in the proof of Lemma 1, let p contain sub-paths p' and p'' where, p' is the sub-path from node O to i and p'' is the sub-path from node $i+1$ to q. Similar to Lemma 1, the total travel time costs of both sub-paths $c_{p'}(t)$ and $c_{p''}(t)$ can be expressed as follows:

$$c_{Oi} = \sum_{j=1}^{i} c(1 - y_j) + \sum_{j=1}^{i} (c + c_j) y_j = ci + \sum_{j=1}^{i} c_j y_j$$
$$c_{i+1,q} = \sum_{j=i+1}^{q} c(1 - y_j) + \sum_{j=i+1}^{q} (c + c_j) y_j = c(q - i) + \sum_{j=i+1}^{q} c_j y_j \tag{20}$$

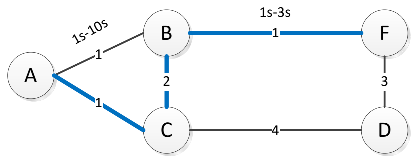

Figure 4: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

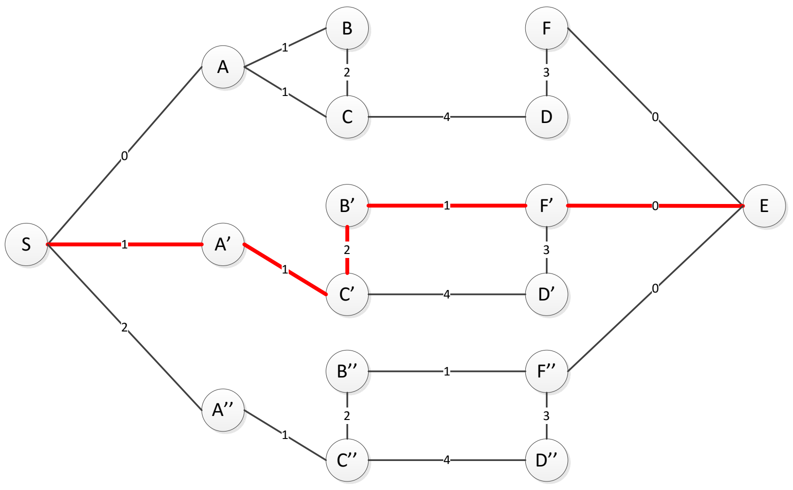

Figure 5: Time expanded $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

As indicated in Lemma 2, the first time constraint can be easily satisfied since there are 2^i possible paths from node O to i with $d_i \leqslant ci + b_1$ since waiting can take place at the originating junction in such a way as to achieve $d_i = ci + b_1$. Nonetheless, the second time constraint (i.e., $d_q = cq + b_1 + b_2$) is addressed by Lemma 1. Thus, a solution of sub-path p'' can be reduce to a problem addressed by Lemma 1.

Same as before, considering Eq. (10), the amount of $cq + b_1$ time-slots can be provided from all of the 2^q paths, while the remaining b_2 time-slots must be identified by the summation of Eq. (20) ($e.g., \sum_{j=i+1}^{q} c_j y_j$). Therefore, to select y_j values the number partitioning problem needs to be solved; completing the NP-complete proof. $\square$

5. Time Expansion Approach

In this section we utilize the "time expansion" approach to demonstrate some of the complexities associated with solving the (Π) problem optimally. Fig. 4 shows a simple graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where road segments (A, B) and (B, F) attain two non-admissible time-slots (from time intervals 1s-10s and 1s-3s, respectively). Notably, time-dependent networks can easily be transformed to static networks using time-expansion as discussed in [25] and allow the problem to be solved in the space dimension only. In this way, the problem is solved in two stages. In the first stage, the graph is expanded to future time-steps considering incremental waiting intervals at the originating junction. Thereafter, a static shortest path algorithm (e.g. Dijkstra [35]) is used to provide a solution. Fig. 5 illustrates the time-expanded graph for the network provided by Fig. 4. Fig. 4 illustrates the optimal solution with a blue line and total cost of $4s$ while Fig. 5 shows the shortest path solution over the time-expanded graph indicated with a red line and total cost of $5s$. As Fig. 5 indicates the shortest path algorithm (e.g., Dijkstra) miss the optimal solution since the non-admissible time slots are not considered to the time-expanded graph. Note that the earliest arrival time at each junction does not ensure the optimal choice based on the label setting property discussed in [36]. The possibility of selecting a junction a little bit later may reduce the destination arrival time since a currently non-admissible segment may become admissible in future time-slot. Therefore, all possible arrival times must be examined at each intermediate junction in order to ensure that an optimal solution is reached.

6. Route Reservation Algorithm (RRA)

A heuristic solution to the $EATD$ problem is derived through the Route Reservation Algorithm (RRA) which also allows an initial wait at the origin. The RRA algorithm employs the Dijkstra's algorithm which is commonly used on static (non-constrained) networks. The proof of correctness of Dijkstra's algorithm indicates two basic properties.

The first one is that Dijkstra's algorithm is a label-setting algorithm since on each iteration a label (i.e., d_j) becomes the actual shortest path from the origin to node j and the algorithm terminates when all nodes are permanently labeled. Labeled nodes are those which an optimal path is found and all the permanently labeled nodes are stored in a predecessor array [36]. The second property is a result of the first property known as the relaxation technique[3], where in each iteration the cost of all non-labeled nodes is $d_i = \min(d_i,\ d_j + c_{ij}(t))$. Therefore, using the label-setting property and the relaxation technique, Dijkstra's algorithm calculates the earliest-arrival-time from origin to each other road junction i. RRA adopts the above properties and returns a feasible solution to the $EATD$ problem accounting also for possible waiting that can take place at the origin.

The RRA algorithm executes in two stages (the inner and outer loop). The inner loop returns the earliest-destination-arrival-time path, from O to D, by allowing vehicles to wait at any intermediate junction until the road segment's state changes from non-admissible to admissible (i.e., it solves the (Π_{AW}) problem). As shown by Lemma 2, if waiting intervals are allowed to all intermediate road junctions (nodes) a polynomial time optimal solution can be found. This relaxed solution, which is not practically implementable, is a lower bound solution to the $EATD$ problem.

Algorithm 1 Inner loop of the RRA (IL-RRA)

```
 1: Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $x_{ij}(t)$, $O - D$, $t_0$
 2: Initialization
 3: $P[i] \leftarrow NULL\ \forall\, i \in \mathcal{V}$                ▷ Sets the predecessor matrix
 4: $Q \leftarrow i\ \forall\, i \in \mathcal{V}$                     ▷ Sets all non-labeled junctions
 5: $d_i = \infty\ \forall\, i \in \mathcal{V}$                       ▷ Sets the arrival time at $i$
 6: $P[O] \leftarrow 0$                                               ▷ Origin Predecessor
 7: $d_O \leftarrow t_0$                                              ▷ Arrival time at Origin
 8: $\epsilon = 10^{-6}$                                              ▷ Waiting Coefficient
 9: $w_{min} = \infty$
10: End of Initialization
11: while $Q \neq \emptyset$ do
12:     $\forall\, i \in Q$ Extract $i$ with $min(d_i)$
13:     Set $i$ as labeled                                            ▷ Since $d_i = d_i^*$
14:     for $\forall\,(i, j) \in \mathcal{E}$ do
15:         if $x_{ij}(d_i^*) == 1$ then
16:             $w_{ij}(d_i^*) = 0$
17:         else
18:             Calculate $w_{ij}(d_i^*)$                             ▷ Required waiting-slots
19:             $c_{ij}(d_i^*) = \tau_{ij} + w_{ij}(t) + \epsilon$    ▷ Update $c_{ij}(d_i^*)$
20:         if $d_j > d_i^* + c_{ij}(d_i)$ then
21:             $d_j = d_i^* + c_{ij}(d_i^*)$; $P[j] = i$             ▷ Update $d_j$
22:             if $w_{ij}(t) < w_{min}$ then
23:                 $w_{min} = w_{ij}(t)$                             ▷ Update $w_{min}$
24: return ($w_{min}$ and path)
```

Subsequently, the outer loop, checks if the solution computed by the inner loop involves waiting intervals at any intermediate junction. If the resulting shortest path from the relaxed problem (inner loop) does not require any waiting at any intermediate node, then the algorithm ends. The obtained solution is considered as the shortest path that the vehicle should follow after waiting the accumulated waiting interval at the origin. On the other hand, if the solution of the relaxed problem involves waiting at one or more intermediate nodes, the outer stage transfers the minimum waiting interval among all nodes to the origin and updates the vehicle's start time (i.e., $t_0 = t_0 + w_{min}$), where $w_{min} = \min(w_{ij}(t))$, $w_{i,j} > 0$ is the minimum waiting at an intermediate node in the obtained relaxed solution. Given the updated waiting time at the origin t_0, the relaxed problem is solved again. This procedure repeats until a path is

[3]The term "relaxation" is used in a way such that an upper bound solution is found by amending the shortest path as explained in [36].

found that does not include any links that are at their capacity (given the estimated reservations) nor any waiting at any intermediate node.

The execution procedure of the inner loop is illustrated in Algorithm 1. Algorithm 1 is similar to the Dijkstra's algorithm but road segment costs are calculated dynamically, since edges cannot be traversed if they are non-admissible. In that case, vehicles are forced to wait at a starting junction (i) of the road segment (i, j), until their admissibility state changes, thus their cost is updated to also include that waiting time.

The initialization of Algorithm 1 is identical to Dijkstra's algorithm with the predecessor matrix initiated as empty (line 3), all junctions initiated as non-labeled (line 4), all variables initiated to have an infinite cost (line 5) and the arrival time at the destination set to t_0 (line 7). Thereafter, the inner loop is executed for all non-labeled junctions and the one with the earliest arrival time is set as labeled (line 11 and 13). Evidently, the first junctions that the algorithm sets in the route is the originating node since all others have infinite cost while in subsequent iterations a new labeled junction is set to be the one that has the earliest possible arrival time (d_i^*) according to the label-setting property. With every new set junction, a dynamic calculation of the traversal cost from the new labeled junction to its neighbors is performed (lines 15 to 28). This dynamic calculation is performed in those cases where segment (i, j) is non-admissible at d_i^* (line 15). The minimum number of time-slots that may be required $w_{ij}(t)$ can be calculated based on both the reservation status of the concerned segment (i, j) and the arrival time at junction i (line 18). In every other case, when the segment attains admissible states, no waiting is necessary (line 16). Therefore, in every iteration the edge cost traversal function $c_{ij}(t)$ is calculated using only the constant travel time cost (free-flow conditions) and the waiting time duration (i.e., $c_{ij}(d_i^*) = \tau_{ij} + w_{ij}(t) + \varepsilon$) (line 19). After all costs have been calculated, a relaxations is performed (lines 20 to 23). If the traversal cost is lower than the arrival time d_j then the arrival time at junction j is relaxed to d_j (i.e., $d_j = c_{ij}(t)$) and junction i is characterized by the predecessor of j. By doing so, RRA updates the earliest arrival time d_i to each non-labeled neighboring junction and stores the minimum waiting interval was from all among all junctions (w_{min}) (lines 22 and 23). The above procedure repeats until all road junctions are characterized as labeled. Finally, the inner loop returns to the outer loop the w_{min} and the identified path.

The outer loop determines if any waiting has been included in the path computed by the inner loop (i.e. $w_{min} \neq 0$). The execution of the outer loop is illustrated in Alg. 2 where, as a first step the total delay that may be observed at the origin (i.e., w_{total}) is initiated to zero (line 2) and afterwards the inner loop is executed (line 3). Thereafter, w_{total} is updated according to the returned w_{min} (line 5). Whenever waiting is identified, the procedure repeats until no waiting is necessary within the computed path (lines 5 to 9). Waiting is added to the origin (i.e., the entry point to the region) and the start time is updated (i.e., $t_0 = t_0 + w_{total}$) (line 7) before the inner loop re-executes with the new starting time (line 8). With each inner loop execution, the waiting intervals that are required are summed to $w_{total} = w_{total} + w_{min}$) (line 9) and repeats until no waiting is necessary.

Algorithm 2 Outer loop of the RRA (OL-RRA)

```
 1: Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $O - D$, $t_0$, $w_{min}$, $P$
 2: $w_{total} = 0$
 3: Execute IL-RRA($\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $x_{ij}(t)$, $O - D$, $t_0$)
 4: $w_{total} = w_{total} + w_{min}$
 5: while $w_{min} \neq 0$ do
 6:     $w_{min} = 0$
 7:     $t_0 = t_0 + w_{total}$
 8:     Execute IL-RRA($\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $x_{ij}(t)$, $O - D$, $t_0$)
 9:     $w_{total} = w_{total} + w_{min}$
10: return (Path and $w_{total}$)
```

Observations. There are cases where two or more feasible solutions for the $EATD$ problem may exists with equal cost. In those cases if one of the two does not require any waiting while the other does, then the algorithm chooses the path with no intermediate node waiting and discards the other one since the algorithm terminates by the first iteration. In the case where both alternative paths experience some waiting at intermediate nodes, then the inner loop should re-iterate at least one more time to identify if the waiting interval can be allocated only at the originating junction. To

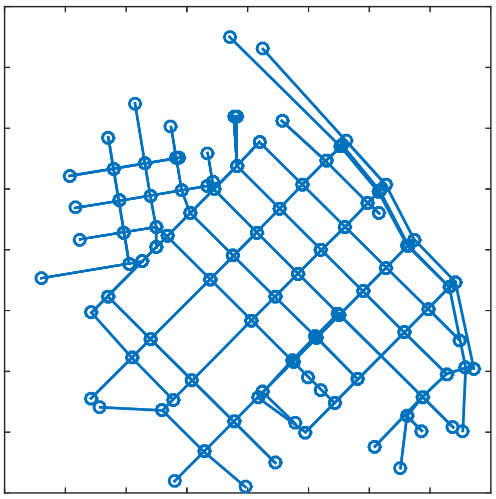
Figure 6: San Francisco road network under consideration.

overcome the selection problem between the alternative solutions, a constant $\varepsilon = 10^{-6}$ is added in case where waiting is required at each particular road segment. Thus, the coefficient ε is added to $c_{ij}(t)$ (i.e., $c_{ij}(d_i^*) = \tau_{ij} + w_{ij}(t) + \epsilon$) (line 18) whenever waiting at a junction is required. This additional cost ensures that when equivalent paths exist, the algorithm will choose the one with the least waiting.

As emphasized above, RRA is a heuristic solution that can be executed efficiently in real time to provide either the optimal or a near optimal path. For example the RRA algorithm will miss the optimal solution for the example illustrated in Fig.4. As already mentioned, the optimal solution is indicated with a blue line and has a total cost of $4s$. The inner loop of RRA will first return as a solution the path consisted from road segment (A, B), (B, F) with a waiting delay of $2s$ at junction B. The returned solution is equivalent to the optimal one, however, the outer loop of RRA requires to clarify if that waiting can be transferred to the origin. Hence, the RRA inner loop re-executes with the new starting time and returns the path consisted from road segment (A, C), (C, B), (B, F) as the final solution with total cost $6s$.

The complexity of RRA is $O(ME^2V)$, where $M < \infty$ denotes the number of iterations that the outer loop of RRA needs before converging to a solution. At this point it is worth pointing out that the RRA algorithm will always terminate in a finite number of iteration. Note that in any scenario, there is a finite number of vehicles which means that there is a finite number of reservations which also means that the intervals for which any link is non-admissible are also finite. Let T_{max} denote the maximum time when a non-admissible interval for any link ends. For any execution of the RRA T_{max} is fixed, while the vehicle initial waiting time t_0 is monotonically increasing at discrete steps which are associated with the end of some non-admissible interval. Clearly, a non-congested path will always be found when $t_0 > T_{max}$, thus $M < \infty$.

7. Simulation Setup And Results

7.1. Network setup and parameters

The road network under consideration is an $1.8\,km^2$ homogeneous region of downtown San Francisco as illustrated in Fig. 6. The spatial compactness and homogeneity of this area was initially investigated in [14], while a similar region is used in [9]. The selected region consists of 99 road junctions and 208 single-lane road segments with lengths varying from $100\,m$ to $400\,m$. To simulate mobility along this region, SUMO micro-simulator [37] is employed using Krauss' car-following model [38]. Standard car-following parameters were used, including: vehicle length of $5\,m$, maximum speed $15\,m/s$, acceleration $2.5\,m/s^2$, deceleration $4.5m/s^2$, driver imperfection 5%, driver reaction time $0.5\ s$, and minimum gap distance $2.5\,m$. The simulation time-step was set to $0.1\ s$.

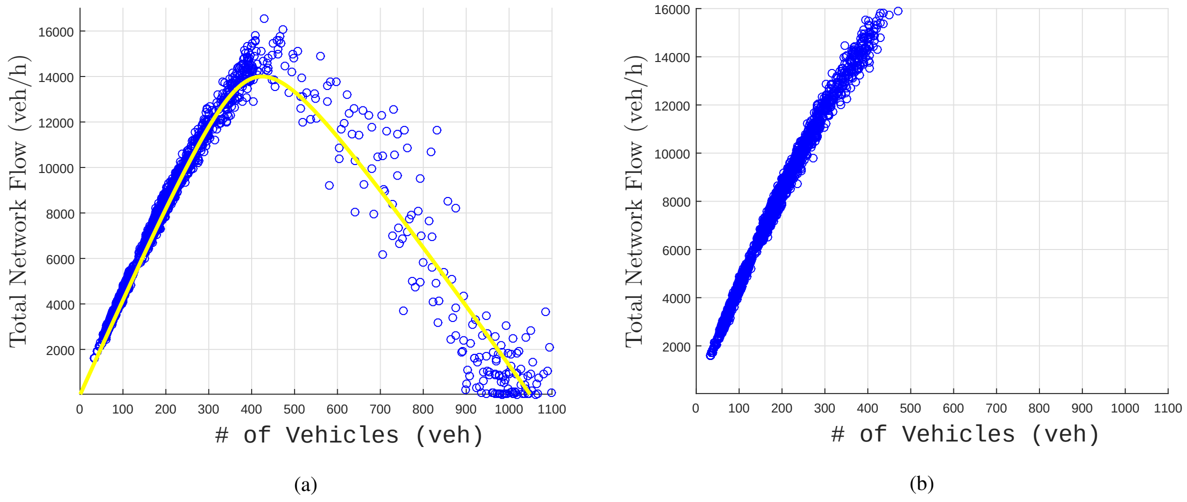

Figure 7: Region MFD of: (a) US; (b) RRA.

7.2. *MFD Analysis*

As a first step, the region's MFD is analysed in order to identify the parameters to be used by the RRA, including v_c and ρ_{ij}^C. To do so, a 6 *hours* scenario was simulated within which for the first four hours the input flow was set to 2000 veh/h and incrementally increased by 2000 veh/h for the next three hours. Thereafter, the input flow was set to 4000 veh/h and 2000 veh/h for the last two hours in order to discharge the network. For the results presented hereafter, ten Monte Carlo simulations were conducted within which the $O - D$ pairs and inter-arrival times were randomly generated.

Fig. 7 (a) depicts the Macroscopic Diagram of the uncontrolled scenario (US) (i.e., where vehicles select their path strictly based on shortest path) which illustrates the total flow as a function of the total density of the network. In the figure, each point corresponds to 1min measurements. The calibrated model shown by the solid yellow line is derived through the automated calibration method proposed by [39] for the single-regime *Van Aerde* model [40]. As detailed in [39], an initial set of free-flow-speed (v_f), speed-at-capacity (v_c), capacity and jam density (ρ^J) values are used together with an iterative procedure to update v_f, v_c and ρ^J to compute the best fit values of the varying parameters which minimize the sum of squared orthogonal errors. From the figure, the following model parameters are obtained: $v_f = 47\,km/h$, $v_c = 40.5\,km/h$ and $\rho^J = 1050\,veh$.

Hence, the RRA algorithm was set to use $\rho_{ij}^C = 40\,veh/km/lane$ (i.e., around 40% of the regions total density) and travel time calculations are estimated using $v_c = 40.5\,km/h$. We emphasize that even though for the purposes of computing the reservations for each vehicle, the constant v_c was used, the actual speed of each vehicle is determined by the simulator based on the assumed model. Fig. 7 (b) depicts the resulting MFD when the RRA algorithm is employed demonstrating the absence of the congested regime. This is achieved by restricting the number of vehicles allowed to simultaneously traverse the network.

To demonstrate the performance of RRA, the average volume of total network flow, the average volume of total network density and average volume of mean network's speeds, obtained from each Monte Carlo realization of the aforementioned network scenario, are depicted in Figs. 8, 9, and 10. For comparison, the performance of US is also superimposed in these figures. Specifically, Fig. 8 illustrate the average volume of the total network flow for both US and RRA, as a function of the simulation over the Monte Carlo simulations. Similarly, Fig. 9 illustrates the average total network density and Fig. 10 the average of the mean network speeds over the Monte Carlo simulations, for both US and RRA. Comparing these three figures, it is evident that using RRA the density decreases (near 330 veh for RRA compared to more than 500 veh for US as shown in Fig. 10) but the traveling speeds remain high and thus the flow is similar to that of US. Additionally, as Fig. 10 illustrates RRA always maintains traffic below critical capacity ρ^C (near 350veh) even when demand is high (i.e., simulation time 200-240min). At the same time, RRA maintains vehicles

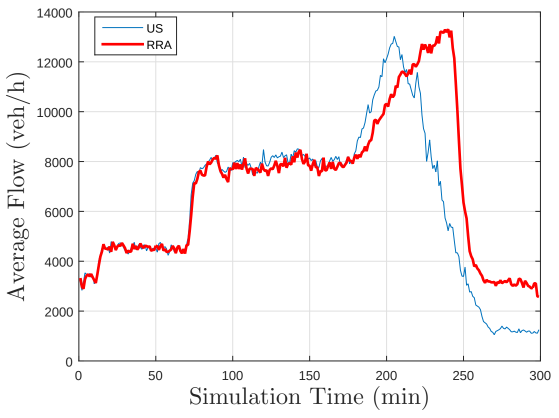

Figure 8: Average network flow over time for US and RRA.

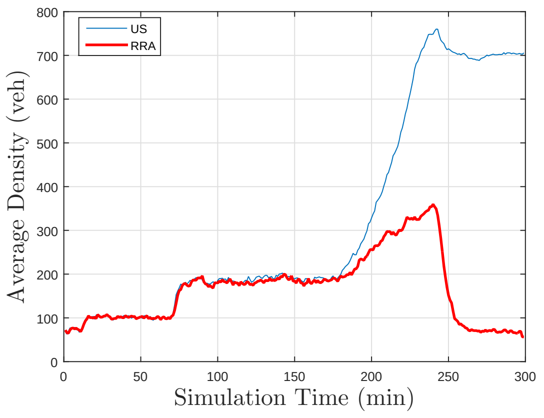

Figure 9: Average network density over time for US and RRA.

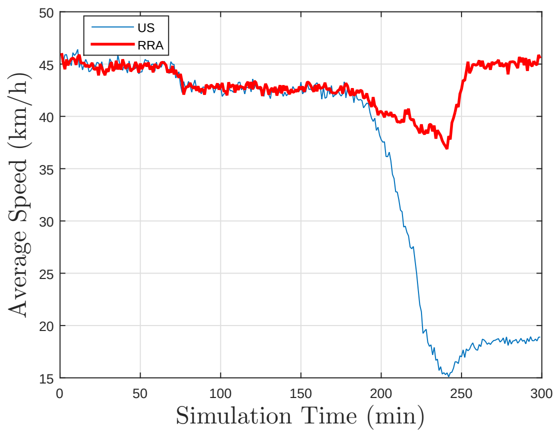

Figure 10: Average network speed over time for US and RRA.

speeds near the speed-at-capacity at all times, as shown in fig. 9.

To demonstrate the improvements obtained by RRA, Figs. 11 and 12 depict the percentage of per road segment density in relation with ρ^C and the per road segment speed as a function of the simulation time, respectively, for the case of US. Similarly, Fig. 13 and Fig. 14 illustrate identical results for the case of RRA. As Figs. 11, 12, 13 and 14 indicate, at low flow-demands the performance of both US and RRA is similar while at high flow-demands RRA outperforms US by avoiding congestion. Clearly, this is due to the fact that at low demands there are no significant restriction in the admissibility of particular road segments and so both approaches yield similar results; on the other hand, as demand increases, there is limited admissibility on road segments and RRA ensures that vehicles wait at their origins until an admissible path can be identified. As shown in Fig. 11, without a control mechanism, a subset of the road segments exceed their critical density and some of them get fully loaded especially in high densities (indicated with the magenta color in the figure). For these road segments, speed drops to near zero (as indicated with blue color in Fig. 12). On the contrary, with RRA road segment densities are maintained below the critical capacity (as shown in Fig. 13), allowing vehicles to maintain their speed near the free-flow speed. Hence, despite the increase in demand, RRA can greatly improve the overall network utilization.

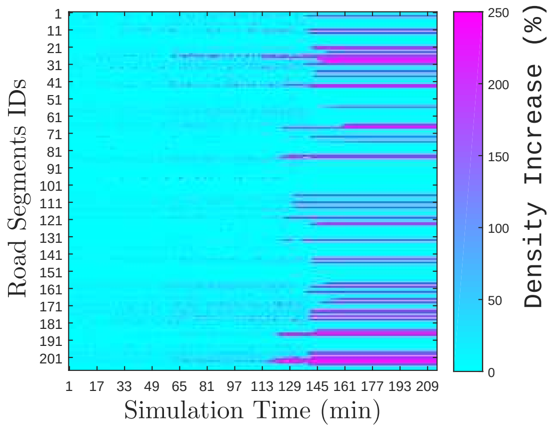

Figure 11: Evolution of traffic density for each road segment over time for US.

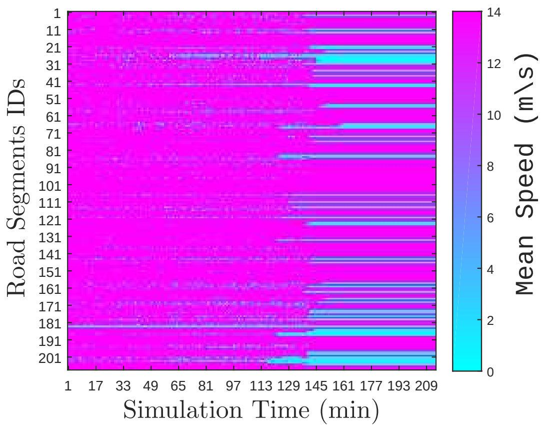

Figure 12: Evolution of speed for each road segment over time for US.

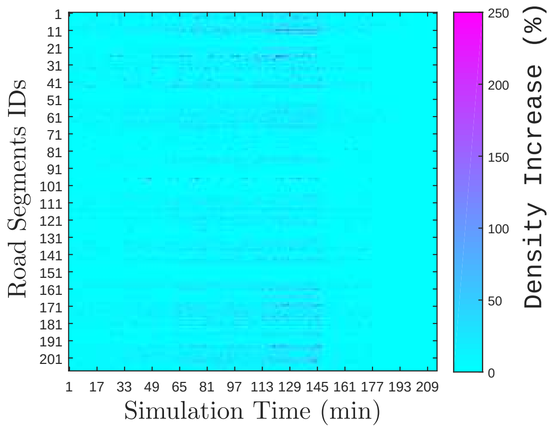

Figure 13: Evolution of traffic density for each road segment over time for RRA.

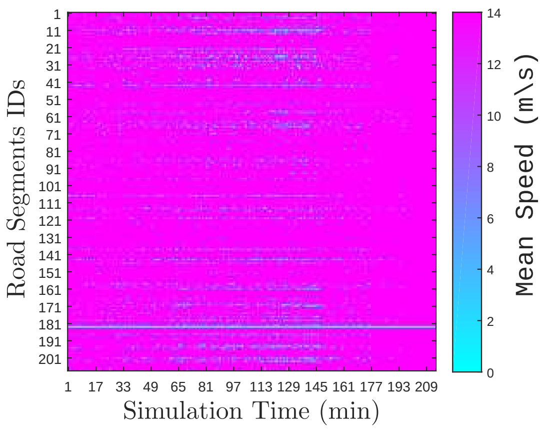

Figure 14: Evolution of speed for each road segment over time for RRA.

7.3. Comparative Performance Results

The proposed route-reservation architecture, that uses the RRA algorithm, is compared against US and with the state-of-the-art Decreasing Order of Time (DOT) algorithm [23]. DOT is an efficient algorithm that finds the time-dependent earliest path (using travel time) within a user-chosen time window. As such, in this work the waiting time at the origin for both RRA and DOT is not considered in the total travel-time for a fair comparison. For the same reason, the travel time estimates for the DOT algorithm were done according to the route reservation requests and using identical $O - D$ pairs. Finally, the maximum allowed waiting interval for DOT was set up to $1min$ (i.e., half the average trip length for the considered network).

It should be noted here that, in the proposed solution, new route reservations are computed solely based on information from previous reservations made and not the actual network state. Since a number of different factors can affect vehicle journeys (including waiting at intersections and other vehicle interactions) the actual traversal of the reserved road segments can occur at time periods not anticipated. These estimation errors are thoroughly examined in the sequel.

As before, 10 Monte Carlo simulations were executed with random $O - D$ pairs and with flow rates varying

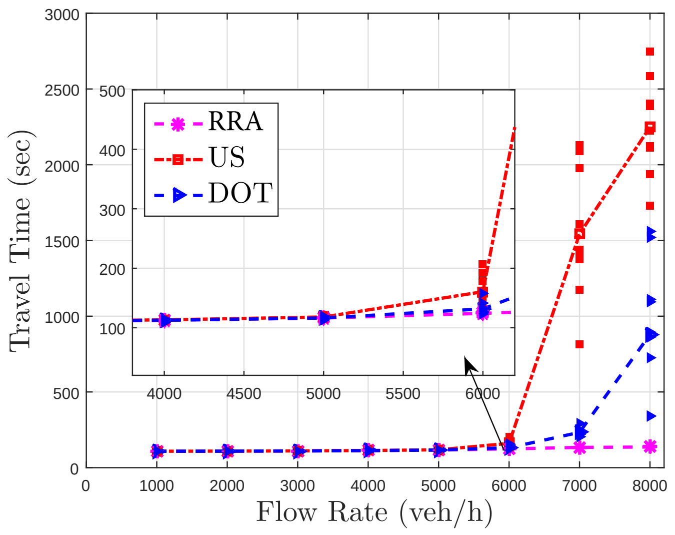

Figure 15: Average travel time.

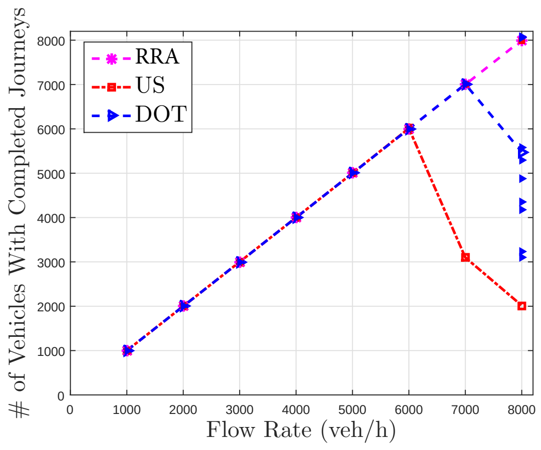

Figure 16: Number of vehicles with completed journeys.

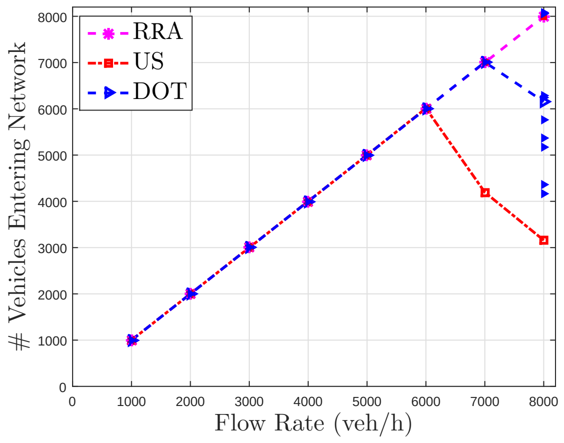

Figure 17: Number of loaded vehicles.

between $1000 - 8000\,veh/h$ over a period of 2 hours.

Figs. 15, 16, and 17 show the average vehicle travel time, the average number of vehicles that completed their journeys and the number of vehicles entering the network within the simulation time, as a function of the different flow rates. The scattered plots in Fig. 15 depict the mean travel time of each realization, while the dashed lines represent the mean travel time for all realizations.

Similarly, the dashed lines in Figs. 16 and 17 illustrate the average number of vehicles that have finished their journey within the simulation time and the average number of vehicles entering the network, respectively. The scattered plots represent the realizations obtained by each simulation run. Figs. 15, 16, and 17 illustrate the overall network behavior considering different flow rates. As indicated in Figs. 15, 16, and 17, in low flow rates ranging from ($1000\,veh/h\ -\ 6000\,veh/h$), there is minimal congestion and thus both algorithms have similar behavior to US. At higher flow rates, congestion emerges and RRA is shown to greatly outperform DOT since the travel time remains short for RRA and all vehicles arrive at their destination within the investigated simulation time.

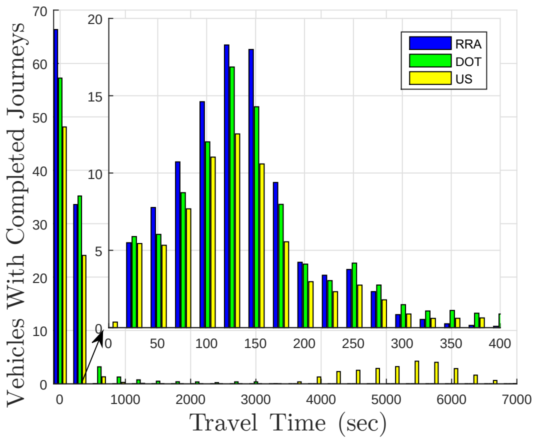

Figure 18: Travel time distribution of 7000 veh/h.

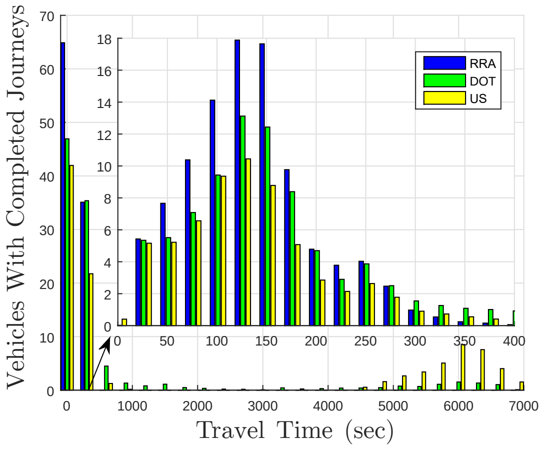

Figure 19: Travel time distribution of 8000 veh/h.

Figs. 18 and 19 illustrate the travel time distribution for all vehicles that manage to reach their destination during the simulation time for flow rates of 7000 veh/h and 8000 veh/h. As illustrated, RRA greatly improved travel time compared to DOT. As shown in Fig. 19, the mean travel time for RRA is 135.9s, for DOT is 695s and for US is 2163.5s. The standard deviation for RRA is 64.8s, for DOT is 1536.8s and for US is 2774.1s demonstrating that as congestion of the road segments increases, RRA is more stable and accurate than DOT. Further, RRA is more resilient to the increase in flow rate since travel times do not significantly deviate.

The RRA performance for different values of ρ_{ij}^{C} is also examined. Figs. 20 and 21 show the average vehicle travel time and the average number of vehicles that completed their journeys for the cases where $\rho_{ij}^{C} = 0.4\rho_{ij}^{J}$, $\rho_{ij}^{C-} = 0.3\rho_{ij}^{J}$, $\rho_{ij}^{C+} = 0.5\rho_{ij}^{J}$ where ρ_{ij}^{C-} and ρ_{ij}^{C+} deviate by -10% and 10%, respectively from the selected critical capacity value (*i.e.*, ρ_{ij}^{C}). Both figures indicate that a 10% increase over the ρ_{ij}^{C} result to a drop in algorithm performance since travel times increase and a lower number of vehicles manages to complete their journeys. Interestingly, using lower capacities the observed algorithm performance is similar to that of ρ_{ij}^{C} since no congestion occurs and travel times are similar since segment densities do not exceed their critical values.

Nevertheless, a lower ρ_{ij}^{C} value increases the waiting time at the origin. This is illustrated in Fig.22 which shows the waiting-time that vehicles need to wait before departing for their journeys. This behavior is expected since a decrease of the allowed capacity reduces the number of vehicles that simultaneously traverse the network. Additionally, as illustrated in Fig.22 a higher ρ_{ij}^{C} value decrease the waiting time at the origin affecting the algorithm performance as congestion occurs. Therefore, the late depart can affected the network behavior as congestion is avoided.

Notably, as demand increases, a higher number of vehicles request to traverse the network. Since the allowed density is restricted bellow the critical value, vehicles prefer to wait at their origin until an admissible path is feasible. Fig. 23 demonstrates that as flow rates increase, waiting time increases exponentially. However, this is expected since in high-demand scenarios, significant waiting needs to be incurred to maintain high network flows. Even so, the average waiting is within acceptable levels (5min) and therefore, a small departure delay could prove sufficient for the overall network operation.

Moreover, Fig. 24 illustrates the mean distance traveled by all vehicles as a function of different flow rates in relation to the shortest distance path (computed using Dijkstra's algorithm). In fact, RRA paths appear to maintain constant travel times (close to the shortest distance path) irrespective of the flow demand, as illustrated in Fig. 24. Looking at the findings of both Figs. 24 and 23 whenever there are non-admissible road segments, the RRA algorithm tends to postpone departures and enable vehicles to traverse through shortest distance paths instead of taking longer routes. This is also verified in Fig. 25, which illustrates the percentage of vehicles that travel through paths other than the shortest distance path. The figure assumes a flow rate of 8000 veh/h. As shown, the majority of vehicles (around

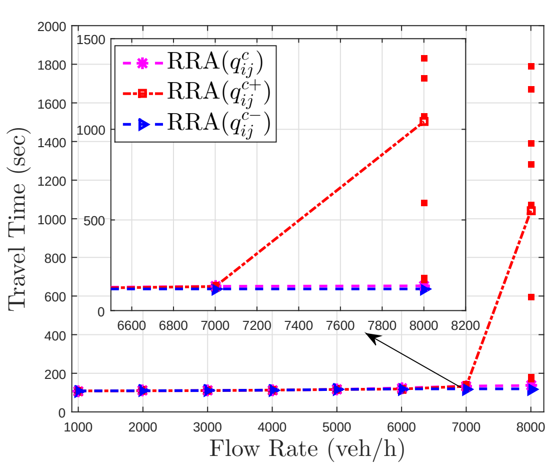

Figure 20: Average travel time for RRA with varying critical capacity values.

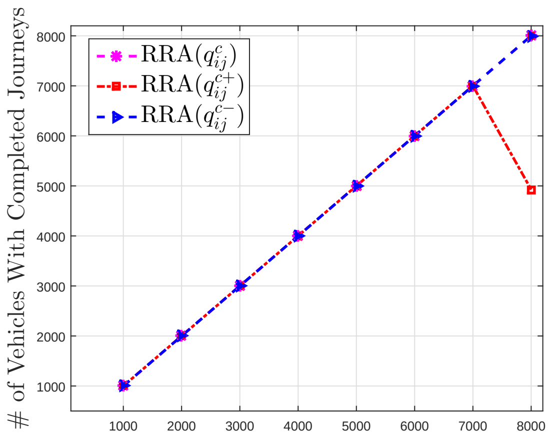

Figure 21: Number of vehicles with completed journeys using RRA with varying critical capacity values.

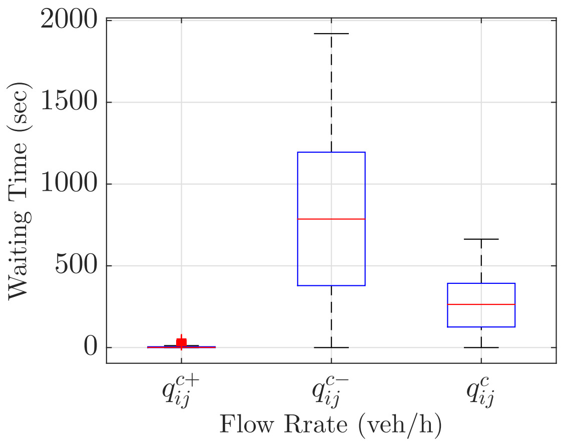

Figure 22: Waiting intervals for ρ^{c+}_{ij}, ρ^{c-}_{ij} and ρ^c_{ij} ($8000veh/h$).

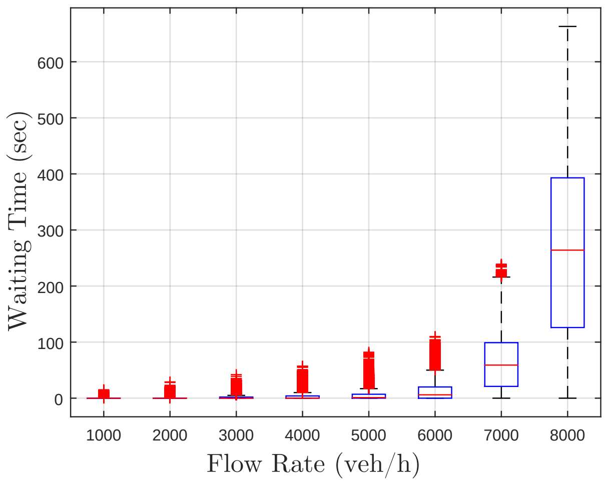

Figure 23: RRA origin waiting times.

75%) were guided through the shortest paths. This indicates that through RRA longer paths are avoided so that both travel time and cost are minimized.

8. Conclusions

This work proposes a new route-reservation architecture which aims to prevent congestion by restricting the traffic density in different road segments within a homogeneous region. The key advantage of this architecture is that it considers both the spatial and temporal density of regions and it exploits more accurate future traffic estimates. The Earliest Arrival Time at Destination problem is formulated and shown to be an NP-complete problem; thus, the Route Reservation Algorithm is proposed which produces low-complexity, close-to-optimal solutions.

Simulation results demonstrate the superiority of the proposed route-reservation algorithm compared to uncontrolled traffic behavior indicating considerable benefits in terms of road utilization and expedited travel times, especially during high demand. Future work includes the improvement of travel time estimations based on waiting

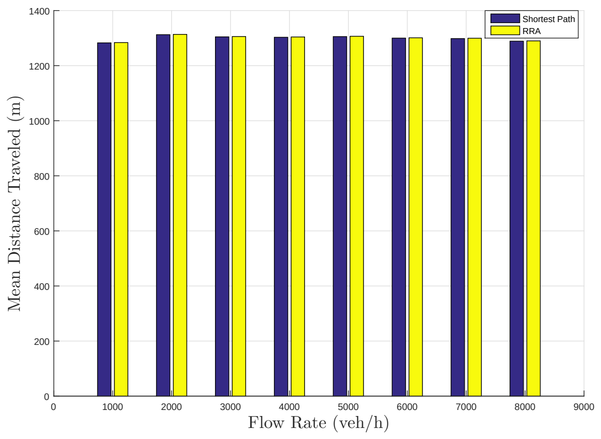

Figure 24: Mean distance traveled comparison for RRA and shortest path.

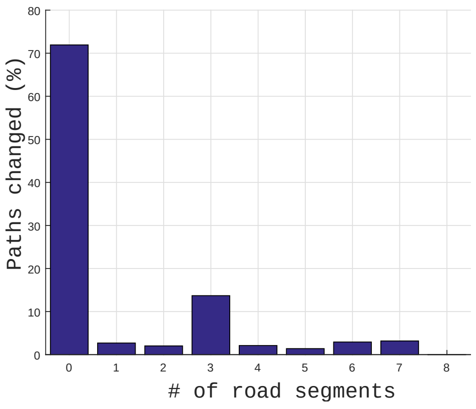

Figure 25: Road segments that changed.

incurred at junctions and the consideration of multiple MFD regions.

References

[1] C. Chen, Z. Jia, P. Varaiya, Causes and cures of highway congestion, IEEE Control Systems Magazine 21 (6) (2001) 26–32.

[2] S. Çolak, A. Lima, M. C. González, Understanding congested travel in urban areas, Nature communications 7 (10793).

[3] N. Geroliminis, J. Sun, Properties of a well-defined macroscopic fundamental diagram for urban traffic, Transportation Research Part B: Methodological 45 (3) (2011) 605–617.

[4] L. Immers, S. Logghe, Traffic flow theory, DEPARTMENT OF CIVIL ENGINEERINGSECTION TRAFFIC AND INFRASTRUCTURE, KASTEELPARK ARENBERG 40, B-3001 HEVERLEE, BELGIUM, course H 111 (05 2003).

[5] C. F. Daganzo, Urban gridlock: macroscopic modeling and mitigation approaches, Transportation Research Part B: Methodological 41 (1) (2007) 49–62.

[6] C. Roncoli, M. Papageorgiou, I. Papamichail, Traffic flow optimisation in presence of vehicle automation and communication systems–part i: A first-order multi-lane model for motorway traffic, Transportation Research Part C: Emerging Technologies 57 (2015) 241–259.

[7] N. Geroliminis, J. Sun, Properties of a well-defined macroscopic fundamental diagram for urban traffic, Transportation Research Part B: Methodological 45 (3) (2011) 605–617.

[8] Y.-C. L. Ho, X.-R. Cao, Perturbation Analysis of Discrete Event Dynamic Systems, The Springer International Series in Engineering and Computer Science, 1991.

[9] K. Aboudolas, N. Geroliminis, Perimeter and boundary flow control in multi-reservoir heterogeneous networks, Transportation Research Part B: Methodological 55 (2013) 265–281.

[10] A. Mazloumian, N. Geroliminis, D. Helbing, The spatial variability of vehicle densities as determinant of urban network capacity, Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 368 (1928) (2010) 4627–4647.

[11] M. Keyvan-Ekbatani, A. Kouvelas, I. Papamichail, M. Papageorgiou, Exploiting the fundamental diagram of urban networks for feedback-based gating, Transportation Research Part B: Methodological 46 (10) (2012) 1393–1403.

[12] J. Haddad, N. Geroliminis, On the stability of traffic perimeter control in two-region urban cities, Transportation Research Part B: Methodological 46 (9) (2012) 1159–1176.

[13] M. Keyvan-Ekbatani, M. Yildirimoglu, N. Geroliminis, M. Papageorgiou, Traffic signal perimeter control with multiple boundaries for large urban networks, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 1004–1009.

[14] Y. Ji, N. Geroliminis, On the spatial partitioning of urban transportation networks, Transportation Research Part B: Methodological 46 (10) (2012) 1639–1656.

[15] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, R. Werneck, Route planning in transportation networks, Tech. Rep. MSR-TR-2014-4 (January 2014).
URL `http://research.microsoft.com/apps/pubs/default.aspx?id=207102`

[16] D. E. Kaufman, R. L. Smith, Fastest paths in time-dependent networks for intelligent vehicle-highway systems application∗, Journal of Intelligent Transportation Systems 1 (1) (1993) 1–11.

[17] L. Xiao, H. K. Lo, Adaptive vehicle navigation with en route stochastic traffic information.

[18] L. Du, S. Chen, L. Han, Coordinated online in-vehicle navigation guidance based on routing game theory, in: Transportation Research Board 94th Annual Meeting, no. 15-3613, 2015.

[19] I. Chabini, Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time, Transportation Research Records 1645 (1998) 170–175.

[20] H. S. Mahmassani, M. Saberi, A. Zockaie, Urban network gridlock: Theory, characteristics, and dynamics, Transportation Research Part C: Emerging Technologies 36 (2013) 480–497.
[21] M. Yildirimoglu, N. Geroliminis, Approximating dynamic equilibrium conditions with macroscopic fundamental diagrams, Transportation Research Part B: Methodological 70 (2014) 186–200.
[22] M. Yildirimoglu, M. Ramezani, N. Geroliminis, Equilibrium analysis and route guidance in large-scale networks with mfd dynamics, Transportation Research Part C: Emerging Technologies 59 (2015) 404–420.
[23] E. Kanoulas, Y. Du, T. Xia, D. Zhang, Finding fastest paths on a road network with speed patterns, in: 22nd International Conference on Data Engineering (ICDE'06), IEEE, 2006, pp. 10–10.
[24] V. Knoop, S. Hoogendoorn, J. Van Lint, Routing strategies based on macroscopic fundamental diagram, Transportation Research Record: Journal of the Transportation Research Board (2315) (2012) 1–10.
[25] G. V. Batz, P. Sanders, Time-dependent route planning with generalized objective functions, in: Algorithms–ESA 2012, Springer, 2012, pp. 169–180.
[26] R. K. Ahuja, J. B. Orlin, S. Pallottino, M. G. Scutella, Dynamic shortest paths minimizing travel times and costs, Networks 41 (4) (2003) 197–205.
[27] C. Menelaou, P. Kolios, S. Timotheou, C. Panayiotou, Congestion free vehicle scheduling using a route reservation protocol, in: IEEE ITSC, 2015.
[28] R. De Neufville, A. Odoni, Airport Systems. Planning, Design and Management, Eurocontrol (12 2004).
[29] D. Condorelli, Efficient and equitable airport slot allocation, Rivista di politica economica 1 (2007) 81–104.
[30] C. G. Panayiotou, C. G. Cassandras, A sample path approach for solving the ground-holding policy problem in air traffic control, Control Systems Technology, IEEE Transactions on 9 (3) (2001) 510–523.
[31] C. Menelaou, P. Kolios, S. Timotheou, C. Panayiotou, On the complexity of congestion free routing in transportation networks, in: IEEE ITSC, 2015.
[32] A. Orda, R. Rom, Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length, Journal of the ACM (JACM) 37 (3) (1990) 607–625.
[33] R. G. Michael, S. J. David, Computers and intractability: a guide to the theory of np-completeness, WH Freeman & Co., San Francisco.
[34] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische mathematik 1 (1) (1959) 269–271.
[35] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, et al., Introduction to algorithms, Vol. 2, MIT press Cambridge, 2001.
[36] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz, Sumo-simulation of urban mobility-an overview, in: SIMUL 2011, The Third International Conference on Advances in System Simulation, 2011, pp. 55–60.
[37] S. Krauss, P. Wagner, C. Gawron, Metastable states in a microscopic model of traffic flow, Physical Review E 55 (5) (1997) 5597.
[38] M. Van Aerde, H. Rakha, Multivariate calibration of single regime speed-flow-density relationships, in: Proceedings of the 6th 1995 Vehicle Navigation and Information Systems Conference, Vol. 334, 1995, p. 341.
[39] M. V. Aerde, Single regime speed-flow-density relationship for congested and uncongested highways, in: Presented at the 74th TRB Annual Conference, Washington, D.C. Paper No. 950802, , 1995.