

# Congestion free vehicle scheduling using a route reservation strategy

C. Menelaou, P. Kolios, S. Timotheou and C.G. Panayiotou

**Abstract**—Traffic congestion in big cities has been proven to be a difficult problem with adverse effects in terms of driver delay and frustration, cost and impact to the environment. Motivated by the approaches used in air-traffic control, this work investigates a method for controlling traffic congestion using time-dependent route reservation. The advances in information, communication and computation technologies has made such a reservation strategy feasible. This paper illustrates that the new reservation strategy is scalable and can be applied even to large metropolitan areas. To do so, we decompose the road network spatially and temporarily and propose a vehicle scheduling and routing algorithm which completely eliminate congestion. Simulation results show that the proposed approach is very promising.

## I. INTRODUCTION

Traffic congestion is a daily phenomenon in big metropolitan areas. This is a serious problem that authorities are struggling to contain and results to multiple adverse affects. These effects include the delay and frustration experienced by drivers, the loss of productivity, increased fuel cost and increased fuel emissions. Traffic congestion occurs as road utilization increases and the number of vehicles that are competing to pass through a road segment is higher than the segment's maximum capacity. In most cases, traffic congestion occurs not because the demand surpasses the capacity of the network, but due to the fact that drivers prefer to follow more popular routes instead of following possibly longer but non-congested routes [1]. In these cases, users over-utilize some road segments and under-utilize some others. This is the result of drivers not having accurate information regarding *future* road utilization. Even if real-time accurate road utilization information is made available to drivers, the problem still remains difficult. Given such information, all rational drivers will prefer roads less congested, thus shifting traffic from congested roads to less congested ones. Due to these rerouting, the non-congested roads will become congested themselves while the congested ones will become underutilized. Thus, as such information becomes available to drivers, road utilization may oscillate without actually solving the problem. Therefore, there is an urgent need for better management of the existing road infrastructure in order to achieve a better traffic load balancing.

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

C. Menelaou, P. Kolios, S. Timotheou, and C.G. Panayiotou 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}@ucy.ac.cy

To date, a multiplicity of off-line approaches have been proposed and implemented for this purpose, including congestion charges, road permits, and even restricted issuance of car license plates. The work presented in this paper tries to achieve a congestion free operation of the road network while taking into account real-time information of the road density.

In order to maintain the road state in the non-congested region, the proposed strategy decomposes the traffic network spatially and temporarily and uses a reservation protocol to schedule vehicles through the earliest-destination-arrival-time path which is also congestion-free. Congestion-free routes can be achieved if the number of vehicles that simultaneously traverse a road segment is maintained under its critical capacity. A road segment that reaches its critical capacity becomes non-available, thus it cannot be used by any more vehicles, until density falls below its critical capacity. The time horizon is divided into intervals and for every interval we keep track of the number of vehicles that are expected to traverse the particular road. With today's technology, this process has become highly feasible. For example, when a vehicle starts its journey (or even earlier), it enters the origin-destination pair and a navigation system can determine the path each vehicle should follow. Given the nominal speed at every road segment, an estimate of the traveling time can be computed and used for route reservation. Doing so also allows for keeping track of the number of vehicles in the road segment during every interval.

In this paper we take this idea one step further. When a vehicle is about to start its journey, it sends its origin-destination pair to a central entity that can determine the best possible path, but avoiding to traverse roads that are expected to be at their maximum capacity during a specific time interval, thus avoiding congestion at every road segment. The central controller may instruct the vehicle to either wait at its origin and start its journey a little bit later (when there will be available capacity on road segments along the shortest path) or find alternative paths where all segments are below their critical capacity. Once a path is identified, it is reserved for the specific vehicle. In this work, it is assumed that all vehicles follow the reserved paths. Enforcing policies or rewarding/punishing strategies for cooperating/non-cooperating drivers, respectively, is a topic of future work.

At this point it is worth emphasizing another benefit of the proposed route reservation approach. By not allowing additional vehicles to enter a road segment once it reaches its maximum capacity, all vehicles that are scheduled to use the segment during that interval are "protected" in the sense

that their time to traverse the segment will not change. This is in contrast with other time-dependent routing approaches which may allow a vehicle to enter a road but change the road cost dynamically (e.g., the time to traverse the road). In this case, if the vehicle finds it beneficial, in terms of arrival time to destination, to traverse a slightly congested road, it also adversely affects the delays of all other vehicles that are also scheduled to traverse the same road.

The remaining of this paper is organized as follows. A literature review is included in Section II and the dynamic routing problem formulation is presented in Section III. Section IV illustrates a heuristic solution for the dynamic shortest path problem. Simulation results that demonstrate the benefit of the proposed approach are illustrated in Section V. Finally Section VI concludes this paper.

## II. RELATED WORK

The work presented in this paper is related with several disciplines. The reservation approach is motivated by the reservations used when solving the ground-holding problem in air traffic control [2]. Finding a path using reservations is related to the routing problem in time dependent networks, as well as, congestion control in intelligent transportation networks. Several authors have addressed these problems in the literature. Exhaustive literature review is out of the scope of this paper, thus only a small subset is cited next.

Air traffic in major airports has been increasing rapidly while the runway capacity has remained stagnant [3]. This is a significant problem since many flights are delayed and as a consequence there is a high economical and environmental cost. Air Traffic Management and Air Traffic Control systems (ATM/ATC) contain this problem by temporarily decomposing the runway capacity and utilizing a reservation approach. The reservations made declare how the airport's available capacity is allocated between the arrival and departure times in such a way as to mitigate delays between flights and also to eliminate the phenomenon of airplanes circulating around waiting for clearance to land [4]. Clearly, the air traffic control problem is much simpler (less airplanes than vehicles, and pilots have to adhere to much stricter rules than vehicles on the ground) for this reason, we are looking into an "aggregated" reservation protocol, i.e., where the state will not include the id of each vehicle, but only a single count of the number of expected vehicles.

In recent years, time-dependent networks, where the cost of a road depends on time, have been proposed to solve time dependent shortest path problems with two main variations sufficing. The first variation deals with the topological changes that happen when road segments may occasionally become unavailable (i.e., road with temporarily infinite cost/weight). The second variation considers dynamic traversal costs, where these costs are updated upon vehicle arrival to a junction [5] [6]. Many existing models and algorithms find the solution over time dependent networks using traffic information that is being available from the infrastructure. Such solutions are based on Dijkstra's or the A\* algorithms

and their objective is to minimized the total travel time [7] [8].

In the literature, many approaches assume that the congestion state of the road network is known and try to navigate vehicles through uncongested regions in such a way as to mitigate congestion [9] [10]. Other approaches schedule vehicles based on the expected future state of the network which can be predicted based on statistical methods or based on dynamic traffic assignment models [11]. The objective of these approaches is to guide vehicles as separate entities aiming at reducing each vehicle's travel time [12]. Therefore, these approaches do not capture the fact that vehicle behavior changes nonlinearly with density. The situation is further exacerbated within the road capacity drop phenomenon which is observed when the state of the road enters the congested region [13]. In this case, the vehicle headway increases and inevitably the capacity drops. This phenomenon changes the maximum capacity of a road segment by up to  $-15\%$  [13]. Therefore, computing efficient routes becomes much more difficult in this region [14] [15] and has a considerable impact on vehicle flows. By emphasizing on the travel time of each individual vehicle, these approaches do not take into consideration the effect that a vehicle's path has on the other vehicles that share the same road segment during the same time. Thus these approaches do not protect the vehicles that have already been scheduled as much as the proposed approach which utilizes reservations.

## III. PROBLEM FORMULATION

In this work, the road network is considered as a graph  $G = (V, E)$  with vertices  $V$  and edges  $E$  being the road junctions and road segments, respectively. Time horizon is quantized into time slots denoted by  $t$ . Each road segment  $(i, j) \in E$ ,  $\{i, j\} \in V$  is characterized by its accumulated reserved traffic  $r_{ij}(t)$ ,  $(i, j) \in E$  (i.e., the number of vehicles in the segment expected at time  $t$ ) and its reservation state  $x_{ij}(t)$ .

In order to ensure the free-flow operation, the volume of traffic on each road segment should be restricted bellow the segment's critical capacity,  $K_{ij}$ . Also, under free-flow operation, it is assumed that vehicles travel at the free flow speed and congestion conditions are avoided. For each time unit, a reservation is allowed provided that the accumulated reserved traffic is less than the road's critical capacity for the entire interval that is required to traverse the road. Therefore, if a vehicle arrives at the beginning of a road segment at time  $t$ , the reservation state of the segment  $x_{ij}(t)$  is defined as follows:

$$x_{ij}(t) = \begin{cases} 1, & \text{if } r_{ij}(\tau) < K_{ij}, \forall \tau = t, \dots, t + \bar{c}_{ij} \\ 0, & \text{otherwise} \end{cases}$$

where  $x_{ij}(t) = 1$  denotes the non-congested state and  $x_{ij}(t) = 0$  the congested state. Also,  $\bar{c}_{ij}$  is the time needed for the vehicle to traverse the road segment assuming the free-flow speed. When the state of a particular road segment

is congested, a vehicle should be forced either to wait at the origin of the trip (until all the road segments in shortest path become non-congested) or to be rerouted through a different path. This decision is made based on the alternative solution that achieves the earliest arrival at the destination junction. Thus, a vehicle is instructed to wait at the starting junction  $s$  or it can be rerouted through a path (not the shortest distance path) that can be used to arrive earliest to the destination; passing only through uncongested road segments, or a combination of the two, i.e., wait for a short period at  $s$  and then take the alternative path. Considering the above notation, the cost of traversing a road segment  $c_{ij}(t)$  can be expressed as follows:

$$c_{ij}(t) = \begin{cases} \bar{c}_{ij}, & \text{if } x_{ij}(t) = 1 \\ \infty, & \text{if } x_{ij}(t) = 0 \text{ and } i \neq s \\ \bar{c}_{ij} + W_s(t), & \text{if } x_{ij}(t) = 0 \text{ and } i = s \end{cases}$$

where,  $W_s(t)$  denotes the smallest number of time units that a vehicle should wait at  $s$  such that a way to traverse from  $s$  to destination  $e$  through only non-congested road segments can be found. As new reservation requests are issued by soon-to-be-departing vehicles, decisions should be made on which route to take, and the number of intervals that a vehicle should wait. When decisions are made, vehicles should follow the allocated route within the scheduled time constrains.

*Earliest Arrival Time problem (EAT):* Given the origin-destination pair ( $s - e$ ), the request time ( $t_0$ ), and route cumulative traffic reservation state, the EAT problem computes the earliest-arrival-time route from  $s$  to  $e$  starting at  $t_0$ . Let  $p_k$  denote the  $k$ -th path from source  $s$  to destination  $e$ .  $p_k = (v_0^k, v_1^k), (v_1^k, v_2^k), (v_2^k, v_3^k), \dots, (v_{n_k-1}^k, v_{n_k}^k)$ , where  $v_j^k \in V$  is the  $j$ -th node visited in the  $k$ -th path, with  $v_0^k = s$  and  $v_{n_k}^k = e$  and  $n_k$  is the length of the path. Also, let  $d_j^k(w)$  denote the earliest arrival time at junction  $v_j^k$  assuming (as before) that the vehicle waits  $w$  time units at the origin before it starts its journey. Then

$$\begin{aligned} d_0^k(w) &= t_0 + w \\ d_1^k(w) &= d_0^k(w) + c_{v_0^k, v_1^k}(d_0^k(w)) \\ &\dots \\ d_{n_k}^k(w) &= d_{n_k-1}^k(w) + c_{v_{n_k-1}^k, v_{n_k}^k}(d_{n_k-1}^k(w)) \end{aligned}$$

thus, the EAT problem is expressed as follows and is an NP-Complete problem and denoted as follows:

$$(EAT) \quad D^* = \min_{w, p_k} d_{n_k}^k(w) \quad (1)$$

#### IV. PROPOSED ITERATIVE DIJKSTRA'S ALGORITHM WITH INITIAL DELAYS

Iterative Dijkstra Algorithm (IDA) with initial delays is a heuristic solution of the EAT problem that aims to route vehicles through non-congested road segments. IDA is inspired by the well known Dijkstra's algorithm which is commonly used on the static (unconstrained) networks.

For a given origin node  $s$  in the network, Dijkstra's algorithm finds the minimum cost path between  $s$  and every other node. Dijkstra's algorithm is categorized as a label-setting algorithm since on each iteration a label (i.e.,  $d_j^k(w) \forall (i, j)$ ) becomes the actual shortest-travel-time path from origin to this particular node  $j$  where termination occurs when the final destination is permanently labeled [16]. Therefore, using the label-setting property and the relaxation technique<sup>1</sup>, Dijkstra's algorithm calculates the earliest-arrival-time on each road junction (i.e.  $d_i = \min(d_i, d_j + c_{ij}(t))$ ). Additionally, a Dijkstra-like algorithm stores all the permanently labeled junctions in an array called Previous (P) [16]. Consequently, IDA adopts the above properties and returns a feasible solution to the EAT problem.

#### Iterative Dijkstra's Algorithm (IDA):

The IDA procedure executes in two main loops. At the inner loop, IDA returns the earliest-arrival-time shortest path, from source to destination, by allowing vehicles to wait at the entrance of each road segment until the segment become available (uncongested). At the outer loop, the algorithm checks if the solution returned by the inner loop involves any waiting at an intermediate road junction. If no waiting is required, the optimal path has been found. On the other hand, if some waiting is required, the maximum waiting is assumed at the source node and Dijkstra's algorithm (inner loop) is run again. The out loop of the IDA may iterate for multiple times until a non-congested path is derived and the waiting intervals are restricted only at the originating junction. In the sequel, the running procedure of IDA is analytically shown.

The inner loop is responsible for finding the earliest arrival time at the destination by allowing waiting intervals at all intermediate junctions (Alg. 1). This loop returns the best possible solution but it is not applicable to real traffic networks since a vehicle can not stop and wait along arbitrary road segments. Nonetheless, this iteration provides a lower bound solution to the EAT problem.

Alg. 1 initializes all variables (similar to the original Dijkstra algorithm) and all junctions are initiated as non-labeled. The variables  $Wait_{temp}$  and  $Wait_{total}$  represents the amount of time units that a vehicle needs to wait at the origin, while the origin's arrival time is set to  $t_0$  and the arrival time of the other junctions is set as infinite. As the algorithm iterates, the junction that has the earliest arrival time (obtained by  $Extractmin(Q)$ ) is labeled and then performs relaxation on its neighbors. To do so, an estimate of the variable  $w_{ij}(t)$  is required where  $w_{ij}(t)$  denotes the smallest number of time units that a vehicle has to wait at  $i$  until road  $(i, j)$  changes state from non-congested to congested. Considering the reservations status of the concerned segment  $(i, j)$  and the arrival time at  $i$  the  $w_{ij}(t)$  can be estimated by calculating the number of consecutive time units that are required in order to pass through  $(i, j)$  when is on non-congested state (Calculate

<sup>1</sup>The term "relaxation" is used in such away to find an upper bound solution by amending the shortest path as explained in [16].

$w_{ij}(t)$ ). The relaxation method is applied in such a way to calculate dynamically the edge delay function  $c_{ij}(t)$  using the  $\bar{c}_{ij}$ ,  $w_{ij}(t)$  and the arrival time of labeled junctions ( $d_i$ ). Therefore if the examined labeled junction ( $i$ ) minimizes the arrival time at its neighbor ( $j$ ) then junction  $i$  is characterized as the previous junction of  $j$  (i.e.,  $P[j] = i$ ). So, in every iteration IDA calculates and updates the earliest arrival time  $d_i$  to each non-labeled neighbor road junction. The above procedure repeats until all road junctions are characterized as labeled.

**Data:**  $G(V, E)$ ,  $r_{ij}(t)$ ,  $s$ ,  $e$ ,  $t_0$ ,  $Wait_{temp}$ ,  $Wait_{total}$   
**Result:** Returns the minimum-arrival-time-route for initial time  $t_0$

```

for  $i \in V$  do
  |  $P[i] \leftarrow NULL$ ;
end
 $Q \leftarrow V$ ;  $P[s] \leftarrow 0$ ;  $t_0 = t_0 + Wait_{total}$ ;  $d_s \leftarrow t_0$ ;  $\underline{d}_s \leftarrow 0$ ;
while  $Q \neq \emptyset$  do
  |  $i \leftarrow ExtractMin(Q)$ ;  $i \leftarrow labeled$ ;
  | for  $e \in E$  do
  | | if  $x_{ij}(d_j) == 1$  then
  | | |  $c_{ij}(d_j) = \bar{c}_{ij}$ ;
  | | else
  | | | Calculate  $w_{ij}(t)$ 
  | | |  $c_{ij}(d_j) = \bar{c}_{ij} + w_{ij}(d_j)$ ;
  | | end
  | | if  $d_j > d_i + c_{ij}(d_i)$  then
  | | |  $d_j = d_i + c_{ij}(d_i)$ ;  $\underline{d}_j \leftarrow \bar{c}_{ij}$ ;  $P[j] = i$ ;
  | | end
  | end
end
 $Wait_{temp} = d_e - \underline{d}_e$ ;
Return( $Wait_{temp}$ );

```

**Algorithm 1:** Inner loop of the IDA (IL-IDA)

The outer loop of the IDA algorithm checks if the waiting delay that is required by the inner loop (i.e.  $w_{ij}(t) \neq 0$ ) can be transferred to the originating junction. If waiting intervals are necessary in the outer loop, the inner loop may re-iterate multiple times. Execution of the outer loop is illustrated on Alg. 2.

**Data:**  $G(V, E)$ ,  $r_{ij}(t)$ ,  $s$ ,  $e$ ,  $t_0$ ,  $Wait_{temp}$ ,  $Wait_{total}$   
**Result:** Returns the earliest-arrival-time-route

```

 $Wait_{temp} \leftarrow 1$ ;  $Wait_{total} \leftarrow 0$ ;
while  $Wait_{temp} \neq 0$  do
  |  $t_0 = t_0 + Wait_{total}$ ;  $Wait_{temp} \leftarrow 0$ ;
  |  $Wait_{temp} = IL-IDA(G(V, E), r_{ij}(t), s, e, t_0,$ 
  |  $Wait_{temp}, Wait_{total})$ ;
  |  $Wait_{total} \leftarrow Wait_{total} + Wait_{temp}$ ;
end

```

**Algorithm 2:** Outer loop of the IDA (OL-IDA)

The outer loop executes two possible cases. If no waiting intervals are needed (i.e.  $Wait_{temp} = 0$ ) the outer loop iterates only once, and consequently the inner loop iterates

once. If waiting intervals are needed then the inner loop may iterate more than one time. In that case, after the first time execution the waiting intervals that are required are summed to the  $Wait_{total} = Wait_{total} + Wait_{temp}$ . While the  $Wait_{temp} \neq 0$  the inner loop runs again with new initial time (i.e.  $t_0 = t_0 + Wait_{total}$ ).

*Observation:* Suppose there are two equivalent traveling-cost paths where the first path does not require any waiting while the second path requires some waiting to take place at the origin. If the algorithm chooses the first path it terminates without further processing. In the second case, the inner loop of the algorithm must re-execute at least another time. This extra run may lead to solutions with worse performance. To avoid this problem, a constant  $\varepsilon = 10^{-6}$  is added every time waiting is required at a particular road segment. Thus, the coefficient  $\varepsilon$  is added to  $w_{ij}(t)$  (i.e.,  $w_{ij}(t) = w_{ij}(t) + \varepsilon$ ) whenever waiting at junctions is required. In a nutshell, this additional weight ensures that algorithm chooses a path without waiting delay whenever the two equivalent cost paths exist.

Notably, IDA is a sub-optimal solution but it is executed efficiently in real time. Finally the complexity of IDA using the binary heap structure is  $O(TE \log V)$  [16], where  $T$  denotes the number of reiterations that IDA needs to converge to feasible solution.

## V. SIMULATION SETUP AND RESULTS

The SUMO simulator [17] was selected for our experiments. SUMO is a microscopic simulator that uses the TraCI interface [18] to control and manage the behavior of vehicles according to the Krauss car following model [19]. For our simulations, a Manhattan-style network topology of 24 two-way, single-lane road segments and 9 junctions was setup.

Because the IDA algorithm requires the critical density for its operation, several simulations were conducted to calibrate a macroscopic model and hence provide an estimate of the critical density of each road. For this reason, several microscopic Monte-Carlo simulations were conducted using SUMO for varying flow-rates. Figs. (1a), (1b), (1c) summarize the obtained results for the density-flow, flow-speed, and density-speed relationships, respectively, as well as the derived calibrated Van-Aerde model (VAM). VAM calibration requires estimates of four parameters [20]: free-speed ( $u_f$ ), speed-at-capacity  $u_c$ , flow capacity  $q_c$ , and jam density ( $k_j$ ); in our calibrated model the values for these parameters are  $u_f = 54$  km/h,  $u_c = 35$  km/h,  $q_c = 865$  veh/h and  $k_j = 80$  veh/km/lane, resulting in an estimated critical density  $k_c = 24$  veh/km/lane. As can be observed, the calibrated VAM is a close match to the observed SUMO micro-simulation behavior for both the free-flow and congested regions, while the calibrated VAM parameters values are representative of realistic urban environments.

The IDA algorithm is compared against the traditional behavior (TB) experienced by vehicles when no reservations are made and with one of the state-of-the-art algorithms, namely DOT (Decreasing Order of Time) [6]. DOT is an efficient technique for the problem of finding the fastest

travel-time path in time-dependent transportation networks, when waiting at the origin is allowed for a certain time-period, without considering the waiting time on the travel time cost. Nonetheless, in this work the waiting time at the origin for DOT is considered in the total travel-time for fair comparison with the proposed IDA. In our simulations the maximum allowed waiting time at the origin for the DOT algorithm was set to 15min.

For the evaluation of the examined algorithms (i.e., IDA, DOT and TB), Monte-Carlo simulations were conducted for different flow-rates, executing a total of ten experiments for each case. In each experiment, vehicles arrive in the network according to the Poisson process with rate equal to the examined flow-rate and are assigned a random source-destination pair. Notice that only the vehicles that completed their journey within the simulated time (1 hour) were considered in the presented results. No overtaking was allowed to ensure that all vehicles followed FIFO queuing (i.e.,  $t+c_{ij}(t) \leq (t+1)+c_{ij}(t+1) \forall i \rightarrow j$ ). Finally, two different car-following model parameter sets were considered to test the performance of the examined algorithms, as shown in table I.

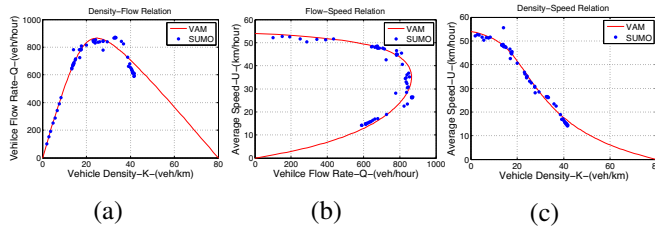


Fig. 1: SUMO Calibration

TABLE I: Car-following Parameters

	Scenario 1	Scenario 2
acceleration	$2.5m/s^2$	$2.5m/s^2$
deceleration	$4.5m/s^2$	$4.5m/s^2$
free-flow speed	$15m/s$	$15m/s$
vehicle length	$5m$	$5m$
min gap	$2.5m$	$2.5m$
driver reaction time	$0.5s$	$1s$
speed deviation factor	0%	20%
driver imperfection	0%	20%

#### Scenario 1:

Scenario 1 represents the ideal case scenario where all algorithms compared were evaluated considering perfect driver reactions and decisions with no delays. Fig. 2 illustrates the average number of vehicles that reach their destination as a function of the different flow rates. Specifically, the dashed lines represent the average number of vehicles (over the 10 realizations) that have reach their destinations within simulation time and the scattered plots are the realizations obtained by each simulation run. Likewise, Fig. 3 shows the average vehicle travel time as a function of the flow rate while, the dashed lines in Fig. 3 represents the average value of the mean travel time for different simulated scenarios and the scattered plots represents the mean travel time of each Monte Carlo simulation.

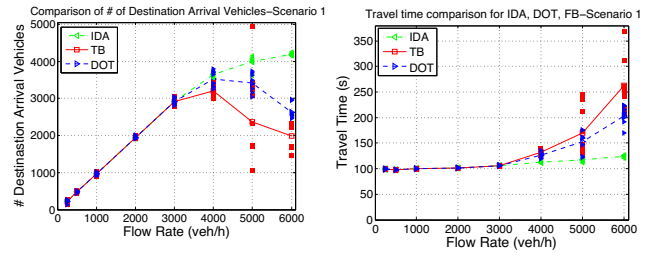


Fig. 2: Number of vehicles Fig. 3: Average travel time towards to the route end ( $t \rightarrow s$ )

Fig. 2 and Fig. 3 indicate that at low flow rates, IDA and DOT behave similar to the traditional behavior since no congestion is experienced. However, at higher flow rates IDA performs better since a larger number of vehicles reach their destination and average travel time is significantly smaller. In either case, both strategies perform better than TB. Comparing IDA and DOT it becomes obvious that it is not enough to schedule vehicles through fastest-travel-time but is considerably greater benefit in considering road capacity and introducing a reservation protocol like the one proposed in this work.

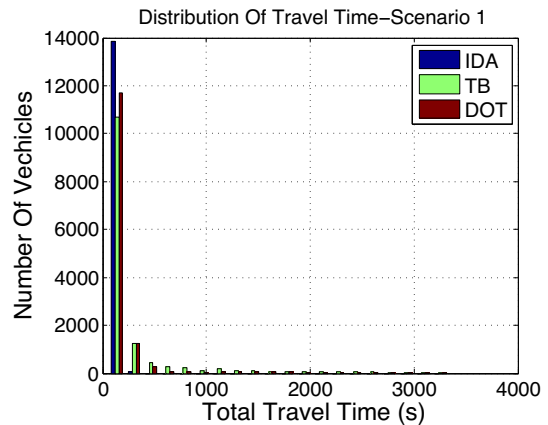


Fig. 4: Travel time distribution (6000veh/h)

Fig. 4 illustrates the travel time distribution for flow rate of 6000  $veh/hour$  where only vehicles that had reach their destination with identical id were selected for evaluation. The mean value of travel time for IDA is 107.981s, for DOT is 207.93s and for TB is 282.81s. The standard deviation of IDA is 34.71, for DOT is 207.93 and for TB is 282.81 seeing that as congestion of the road segments increases, the travel time of DOT and TB increases at a higher rate than that of IDA. Hence, Fig.4 demonstrates the great potentials of applying route reservation based on capacity constrains in contrast to the solution proposed to DOT that does not take into account the network behavior during congested conditions.

#### Scenario 2:

This scenario uses the parameters of Scenario 2 (Table I) and represent a more realistic driver behavior.

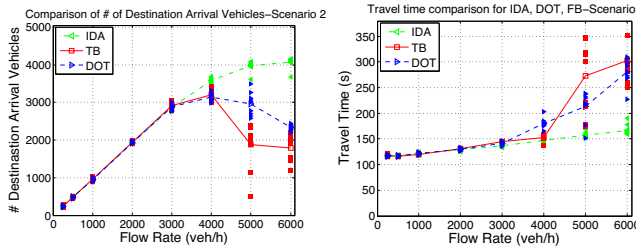


Fig. 5: Number of vehicles towards to the route end ( $t \rightarrow s$ )

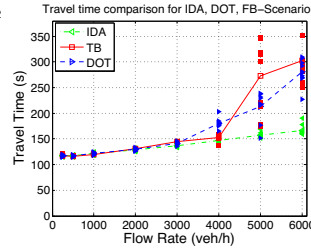


Fig 5 and Fig. 6 indicate that on lower flow rates the mean travel time of all approaches was slightly affected in the sense of travel time and number of destination arrival vehicles due to drivers imperfections that emerge from parameters change. On higher flow rates IDA maintained its behavior since the number of vehicles that successfully terminated was only slightly affected and a minor effect was observed due to parameters changes. As in the previous scenario, DOT performed better than TB but both approaches were highly affected by the parameter changes. Both figures shows the significant impact on DOT and TB when considering driver imperfections, and longer travel times were observed on high flow rates as opposed to the stable performance achieved by IDA.

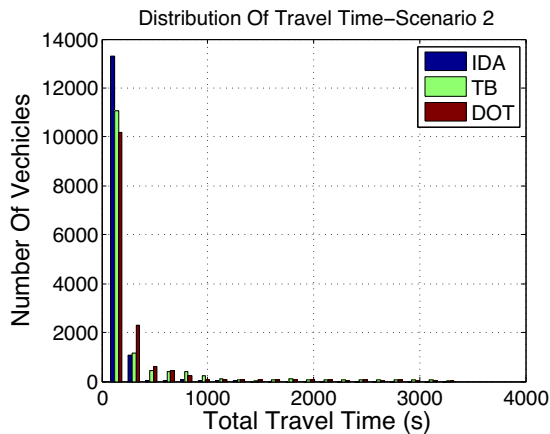


Fig. 7: Travel time distribution (6000veh/h)

Fig. 7 shows the travel time distribution for 6000 veh/h. The mean value of travel times for IDA is 145.56s, for DOT is 271.06s and for TB is 291.04s. The parameter changes have affected the standard deviation of TB to 498.88, DOT to 415.23 and IDA to 75. Finally, Fig. 7 shows that IDA can maintain its behavior while it demonstrates the robustness of its behavior regardless to driver characteristic variations.

## VI. CONCLUSIONS

This work introduced the potential benefit of applying a route reservation method that aims to schedule vehicles through non-congested road segments. The IDA has been developed to solve the EAT problem and several simulations have been conducted on a small network. Simulation results were compared against the traditional vehicle behavior and another state-of-the-art approach, demonstrating that the

proposed solution enhances the networks efficiency while it decreases the average travel time at congested conditions. Finally this work demonstrate that route reservation can achieve substantial improvements in road utilization and thus practical solutions could result in considerable benefits.

Future challenges includes the design of a distributed version of route reservation that are able to handle larger traffic networks and more intelligent reservation algorithms that approach optimality.

## REFERENCES

- [1] C. Chen, Z. Jia, and P. Varaiya, "Causes and cures of highway congestion," *IEEE Control Systems Magazine*, vol. 21, no. 6, pp. 26–32, 2001.
- [2] C. G. Panayiotou and C. G. Cassandras, "A sample path approach for solving the ground-holding policy problem in air traffic control," *Control Systems Technology, IEEE Transactions on*, vol. 9, no. 3, pp. 510–523, 2001.
- [3] R. De Neufville and A. Odoni, *Airport Systems. Planning, Design and Management*, 2003.
- [4] D. Condorelli, "Efficient and equitable airport slot allocation," *Rivista di politica economica*, vol. 1, pp. 81–104, 2007.
- [5] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. Werneck, "Route planning in transportation networks," Tech. Rep. MSR-TR-2014-4, January 2014. [Online]. Available: <http://research.microsoft.com/apps/pubs/default.aspx?id=207102>
- [6] I. Chabini, "Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time," *Transportation Research Records*, vol. 1645, pp. 170–175, 1998.
- [7] B. Ding, J. X. Yu, and L. Qin, "Finding time-dependent shortest paths over large graphs," in *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*. ACM, 2008, pp. 205–216.
- [8] G. Nannicini and L. Liberti, "Shortest paths on dynamic graphs," *International Transactions in Operational Research*, vol. 15, no. 5, pp. 551–563, 2008.
- [9] D. E. Kaufman and R. L. Smith, "Fastest paths in time-dependent networks for intelligent vehicle-highway systems application\*," *Journal of Intelligent Transportation Systems*, vol. 1, no. 1, pp. 1–11, 1993.
- [10] L. Xiao and H. K. Lo, "Adaptive vehicle navigation with en route stochastic traffic information," 2014.
- [11] I. Kaysi, M. Ben-Akiva, and H. Koutsopoulos, *Integrated approach to vehicle routing and congestion prediction for real-time driver guidance*, 1993, no. 1408.
- [12] L. Du, S. Chen, and L. Han, "Coordinated online in-vehicle navigation guidance based on routing game theory," in *Transportation Research Board 94th Annual Meeting*, no. 15-3613, 2015.
- [13] S. Hoogendoorn and V. Knoop, "Traffic flow theory and modelling," 2012.
- [14] J. Wardrop, "Some theoretical aspects of road traffic research," in *Inst Civil Engineers Proc London/UK*, 2004.
- [15] B. S. Kerner, "Optimum principle for a vehicular traffic network: minimum probability of congestion," *Journal of Physics A: Mathematical and Theoretical*, vol. 44, no. 9, p. 092001, 2011.
- [16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein *et al.*, *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.
- [17] M. Behrisch, L. Bieker, J. Erdmann, and 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.
- [18] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: an interface for coupling road traffic and network simulators," in *Proceedings of the 11th communications and networking simulation symposium*. ACM, 2008, pp. 155–163.
- [19] S. Krauss, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E*, vol. 55, no. 5, p. 5597, 1997.
- [20] H. Rakha and B. Crowther, "Comparison of greenshields, pipes, and van aerde car-following and traffic stream models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1802, no. 1, pp. 248–262, 2002.