Minimizing traffic congestion through continuous-time route reservations with travel time predictions

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

Abstract—Road traffic congestion occurs as demand exceeds infrastructure’s capacity. In this work we propose a continuous-time route reservation architecture to efficiently manage vehicle routing decisions thus to eliminate congested conditions. The related problem is formulated as a mixed integer linear programme (MILP) that routes vehicles while avoiding road segments that reach their critical density. To reduce the high computational complexity of the MILP approach we develop a heuristic algorithm, which is shown to perform well compared to the MILP formulation. The heuristic algorithm is used to conduct extensive micro-simulations across a real network topology of the San Francisco area. Finally, an extension of the proposed architecture is also developed where the modeling uncertainties are considered through a time-varying regression method making able in real-time accurate travel-time predictions minimizing the reservation errors.

I. INTRODUCTION

Traffic congestion has significant economic and social consequences in modern cities and their citizens, including fuel waste, productivity loss and driver frustration. Congestion occurs as traffic demand surpasses the infrastructure’s available capacity, a phenomenon mostly observed during rush-hours. Interestingly, recent studies indicate that even with real-time accurate information of traffic state, congestion can still occur due to, for example, drivers’ selfish behavior who continuously seek to minimize their travel times [1].

Within the context of connected and autonomous vehicles, previous works by the authors [2], [3] proposed a route reservation scheme that routes vehicles through non-congested road-segments. As shown in Fig. 1, a vehicle that wants to perform a trip sends a request to a Road Side Unit (RSU), providing information about its origin-destination pair \((O-D)\) and its expected departure time. In turn, the RSU, given all previous requests, accommodates the vehicle’s request by determining a path consisting of congestion-free road segments\(^1\), i.e. road segments that are expected to be below their critical capacity. Notably, the RSU computes routes that avoid congested segments while it also ensures that each vehicle will arrive to its destination at the earliest possible time. Vehicles may be instructed to wait at their origin for a period of time before commencing their trip, if this action will lead to shorter trip times. At the same time, the RSU also reserves each road segment at the time that the vehicle is expected to traverse them.

These reservations provide estimates of the future state of each road segment, and thus when segment’s reservations exceed its critical density during particular time intervals, the RSU flags the segment as non-admissible. We point out that with the recent advances in information and communication technologies, such architecture is feasible. The previous works on vehicle routing with route reservations [2], [3] have assumed constant vehicles speed (typically the free-flow speed) which is used for the calculation of road segment transit-times considering discrete-time reservation intervals. Nonetheless, considering constant speed is not always a valid assumption in practice. Even when having free-flow conditions for the entire road network, speed variations are observed due to various factors such as the stochastic nature of human driving which leads to speed fluctuations between interacting vehicles, and the crossing of intersections with the same or different road segment priorities. Furthermore, in [2], [3] a discrete-time framework is adopted which in some cases may result to overestimating the origin waiting times and the reservation intervals.

In this work we modify the reservation architecture to address the above two shortcomings. First, instead of considering constant transit-times we investigate the use of predicted transit-times. This is achieved by introducing feedback to the RSU. In particular, whenever a vehicle exits a road segment it sends a message to the RSU with the experienced transit-time. Using the collected road segment transit-time measurements, the RSU predicts the transit-times for future vehicles aiming to improve the

\(^1\)Congestion-free road segments are also referred as admissible road segments.
accuracy of reservations. Second, reservations and vehicle routing are made considering a continuous-time horizon. In this work, two algorithms are proposed for vehicle routing that provide a different trade-off between solution quality and computational complexity.

More specifically, this paper extents the work presented in [4] with the following key contributions:

- Development of a Time-Varying Multiple Linear Regression (TMVR) method for predicting road segment transit-times.
- Formulation of the shortest-path problem with the origin waiting time and traversal of admissible road segments under the continuous-time framework, as a Mixed Integer Linear Program (MILP).
- Development of a heuristic Iterative Dijkstra-based Algorithm in Continuous-time (IDAC) that yields to fast, accurate and nearly optimal paths.

The remainder of the paper is organized as follows. Section II provides an extensive literature review, while Section III mathematically describes the continuous-time route reservation problem taking into consideration predicted road segment transit-times. Section IV describes the TMVR method for transit-time prediction to improve the accuracy of route reservations. The subsequent two sections describe two approaches for the solution of the considered continuous-time route reservation problem. Section V formulates a MILP that yields the optimal path, while Section VI proposes the IDAC which is a low-complexity heuristic algorithm. Simulation results included in Section VII examine the performance of the proposed algorithms. Finally some implementation challenges are provide in Section VIII while Section IX concludes the work and discusses future research directions.

II. RELATED WORK

Several recent works proposed dynamic approaches for controlling congestion such as: gating and perimeter control. These techniques focus on dynamic congestion mitigation by controlling the input flows at the boundaries of protected region (i.e., street closures and traffic signaling control) [5], [6]. Within these methods, all vehicle movements and traffic dynamics are defined according to the Network Fundamental Diagram (NFD) (extracted from real-time measurements) which describes the relation between the network-wide space-mean speed and its corresponding density. The NFD provides clear indications on the region’s critical capacity where, in turn boundary control techniques can be used to restrict regions’ in-flow, allowing the possibility to macroscopic model and control the complex traffic dynamics making feasible the implementation of more computational efficient approaches [7], [8]. The major drawback of these techniques is that, unwanted queues may be observed at boundaries which in turn obstruct the upstream network destinations [9], [10]. Recent work presented in [11] utilizes an on-line adaptive optimization scheme that promises a better congestion distribution as it tries to anticipate the aforementioned queuing problem by considering how generated queues affect the vehicular movements. Despite that effort, the region’s boundaries remain uncontrolled. In this way, queues can be generated within a region (called as “artificial inter-regional queues”) which produce delays in the network. In contrast to the aforementioned approaches, this work considers a control method in which congestion may be distributed throughout the network but it may also be avoided by restraining vehicles at their origin in order to prevent network delays and queue generation.

The motivation behind our work is the fact that network utilization is closely related to the drivers routing decisions, especially during peak hours [1]. In most cases, traffic congestion occurs due to lack of effective traffic management strategies that could in real-time inform drivers about road conditions and advise to follow possible alternative routes rather than to follow shorter-distance congested routes [12]. In this way, recent developments in Intelligent Transportation Systems provide a plethora of complementary solutions to minimize travel times by guiding drivers via the shorter-travel-time paths [13], [14], [15]. Solutions based on Dynamic Traffic Assignment (DTA) (as detailed in [16]) constitute the state-of-the-art routing advisory schemes within which travel times are improved by restricting the imbalance of flows within a region. Similar with perimeter control approaches, DTA schemes correlate routing decisions with the NFD [17], enabling the spread of the load across a larger area of the network that could significantly improve the overall traffic flow [18].

Notably, when there is large inconsistency between the observed travel times and travel time predictions then the performance achieved by DTA diminishes [19]. Clearly, the aforementioned issue can impact other state-dependent schemes and thus a large volume of work has focused on achieving accuracy in travel time predictions; using analytical or statistical methods [20]. Both methods, can be employed to predict travel times in a deterministic or stochastic manner [21], [13], always using data obtained through various traffic surveillance sensors (e.g., loop detectors, mobile detectors, radars and cameras [22]). With respect to the analytical methods, Kalman filtering algorithms constitute the primary state estimation method where fresh observations that become available are used to continuously update the state variables. Unfortunately, Kalman filtering cannot be utilized in our approach as there is no simple and accurate model that can be applied in route reservations.

Statistical methods, are mainly data driven techniques based of current travel time observations [20], [23], [24]. It should be noted here that all the aforementioned methods only use the travel time measurements of the corresponding road segments without considering factors affecting travel times such as driver interactions and delays observed at intersections [20]. Along these lines, the work presented in [25] indicates that route travel times are affected by the segment’s traversal time and the delays observed at the
intersections (expressed as travel time penalties). In this work, we propose a simple and accurate travel time prediction method based on linear multi-variable regression that takes into consideration the stochastic interaction between drivers, and is able to cope with larger scale networks.

The idea of trip reservations was mentioned in [26] within which trip reservations are proposed to relieve the holiday congestion problem on a rural motorway in a similar way as the train seat reservations. More specifically [26] studied how the trip reservations can be affected from adjustable departure times by quantitatively evaluating reservations with a stated-preference survey. Along same lines, work in [27] conceptually proposed a highway booking system that operates along side other driver information systems. In [27] for each time period each road segment has an available capacity where the number of available seats in each vehicle are considered within the problem’s capacity constraints. Both aforementioned works [26], [27] indicate that the trip reservations are promising as they can significantly improve the transportation systems efficiency.

Trip booking methods also are investigated in [28] where an infrastructure manager uses a slot allocation algorithm to manage the demand (departure time allocation) for pre-specified routes. Similarly, a congestion pricing alternative is proposed in [29] where a reservation system is proposed to control the capacity of vehicles that enter a protected region; by managing the vehicles’ flows passing through a cordon.

The idea of trip reservations in highways is also investigated in recent literature as shown in [30],[31] confirming that the efficiency of a reservation system can overpass the existing traffic management methods. Extensions of the last two works are presented in [32] and [33] where reservations are used only by users that are willing to pay in order to access a high priority lane providing better quality of service and travel time guarantees. In addition, in the later work an auction-based reservation is proposed to reduce inefficiency due to user heterogeneity. The contribution of this work is that it explicitly segments time and space and explicitly formulates and solve an optimization problem that allows vehicles to arrive at their destination avoiding road segments that are expected to be at their capacity.

### II. Problem Formulation

A homogeneous region is expressed as a directed graph \( G = (V, E) \) with vertices \( V \), \( N_V = |V| \), representing the road junctions and edges \( E \), \( N_E = |E| \), representing the road segments. Each road segment \((i, j)\) ∈ \( E \), \( \{i, j\} \) ∈ \( V \) is described by parameters \( \lambda_{ij} \), denoting the number of lanes and \( l_{ij} \) representing the segment’s length.

All traffic dynamics of each region are defined according to the region’s NFD [34] with parameters \( \rho^C \), \( \rho^I \), and \( v^f \), representing the critical density corresponding to the maximum flow, the jam density and the free-flow speed, respectively. Similarly, the traffic dynamics of each road segment \((i, j)\) ∈ \( E \), \( \{i, j\} \) ∈ \( V \) are described according to parameters \( \rho^I_{ij} \), \( \rho^C_{ij} \) and \( \rho_{ij}(t) \) indicating the jam density, the critical density, and the instantaneous density at time \( t \), respectively. Note that, the critical density denotes the maximum density that a road segment can accommodate in order to operate at \( v^f \), i.e., \( \rho_{ij}(t) \leq \rho^C_{ij} \) and is approximated with the quantity \( (\rho^C/\rho^I)\rho^C_{ij} \). Further, we define the variable \( t_{ij}(t) \) denoting the predicted travel time of road segment \((i, j)\), as predicted at time \( t \).

According to the proposed route reservation architecture, the RSU is responsible to keep track of all road segment reservations based on all previous vehicle requests. To do so, the variable \( r_{ij}(t) \) defines the accumulated number of reservations at time \( t \) and the variable \( R(t) \) denotes the network’s accumulated number of reservations (i.e., \( R(t) = \sum_{\forall (i,j) \in E} r_{ij}(t) \)), at \( t \). Thus, the state variable \( r_{ij}(t^l, t^u) \) denotes the accumulated number of reservations within \((i, j)\) during the time interval \( (t^l, t^u) \) where \( t^l \) and \( t^u \) denote the lower and upper time bounds i.e., \( t^l \leq t^u \). Accordingly, a road segment \((i, j)\) is considered as admissible within time interval \((t^l, t^u)\) if during this interval the number of reservations is not larger than the number of vehicles corresponding to the segment’s critical density, i.e.

\[
r_{ij}(t^u, t^l) \leq \rho^C_{ij}
\]

In this way, the RSU constructs the admissible set of each segment \((i, j)\) ∈ \( E \) at time \( t_0 \), \( S_{ij}(t_0) = \{(t_{ijk}^l, t_{ijk}^u) \mid (i, j, k) \in K_{ij}(t_0)\} \), which defines the time intervals \( (t_{ijk}^l, t_{ijk}^u) \), \( k \in K_{ij}(t_0) = \{1, \ldots, K_{ij}(t_0)\} \), at which the particular segment is admissible for new vehicles. The variable \( K_{ij}(t_0) \) represents the number of admissible time intervals of segment \((i, j)\) at time \( t_0 \). Note also that, \( t_{ijk}^l \) and \( t_{ijk}^u \) denote the lower and upper bounds of the \( k \)-th admissible time interval of the link \((i, j)\) ∈ \( E \), respectively, while it is true that \( t_{ijk}^l < t_{ijk}^u < t_{ijk}^{l+1} \). These time intervals are determined by the RSU based on past reservations, the path of the requesting vehicle and the predicted road segment transit times.

Fig. 2 depicts a toy example of how the admissible time intervals evolve in time for link \((i, j)\). It is predicted that the link can be traversed in 2 time units (tu), while its critical density is equal to 2 vehicles for its entire length.
Also, the considered time horizon is assumed to be 6 tu. In the figure, the number in the shaded area represents the remaining capacity of the link for each particular time interval. Observing, the figure at $t_0 = 0$ tu there are no vehicle requests for a path, hence the link has a single admissible interval such that $t^u_{ij} = 0$ tu and $t^u_{ij} = 6$ tu. Next, the first vehicle is expected to arrive at $t_0 = 1.1$ tu and hence RSU make a reservation from $t = 1.1$ tu until $t = 3.1$ tu. After the first request, link is admissible for the same time duration (whole interval) as the link’s critical density is 2 vehicles. The second vehicle is expected to arrive at $t = 2.8$ tu and reserves the link from $t = 2.8$ tu until $t = 4.8$ tu. Both requests share the link during the interval 2.8 tu to 3.1 tu so that the link becomes non-admissible for that period. This yields to two admissible time intervals for the remaining time (i.e. from $t^l_{ij1} = 0$ tu to $t^l_{ij1} = 2.8$ tu and from $t^l_{ij2} = 3.1$ tu to $t^l_{ij2} = 6$ tu). In the same way, a request by a third vehicle allows a reservation within the interval $t = 4$ tu to $t = 6$ tu, yielding to three separate time intervals.

Congestion-free routing is ensured whenever vehicles traverse the network without violating the capacity constraints (i.e., vehicles traverse only admissible road segments). To formulate the resulting mathematical programme, first define the variables $d_t$ and $d_l$ representing the routing request time and the vehicle arrival time at node $i$, respectively. The road segments admissibility intervals are predicted based on the assigned vehicle path, the past reservations and the predicted travel times at each road segment (i.e., $\hat{\tau}_{ij}(t)$), according to the observed speed. The RSU utilizes the admissible intervals of each link to compute vehicle routes that adhere to the admissibility condition equation (1).

Hence, the cost of traversing the road segment $(i, j)$, $c_{ij}(d_t, t_0)$, is given by:

$$c_{ij}(d_t, t_0) = \begin{cases} \hat{\tau}_{ij}, & \text{if } d_t + t \in S_{ij}(t_0), \forall 0 \leq t \leq \hat{\tau}_{ij}, \ i \neq O \\ \hat{\tau}_{ij} + w, & \text{if } t_0 + w + t \in S_{ij}(t_0), \forall 0 \leq t \leq \hat{\tau}_{ij}, \ i = O \\ \infty, & \text{otherwise} \end{cases}$$

(2)

where $d_t$ denotes the arrival time at node $i$, $t_0$ the request time, $O$ the origin node, and $w \in \mathbb{R}$ the origin waiting time, i.e., the time that the particular vehicle should wait at its origin prior to start its journey. Accordingly, two alternative options arise in case that the shortest-distance path of vehicles includes a non-admissible segment. The first prompts vehicles to wait at their origin until all road segments in the shortest path become admissible. The second chooses an alternative path where all segments are admissible. Combining both options may yield a better solution (i.e., wait for a period at the origin and then take an alternative path).

This work formulates and solves an optimization problem for determining the path that will allow a vehicle to arrive at its destination node $D$ at the earliest possible arrival time by avoiding traversing non-admissible road segments [2]. This problem is referred to as the Earliest Arrival Time at Destination (EATD) problem which is mathematically stated in the following section.

### Earliest Arrival Time at Destination (EATD) problem

Given an origin-destination $(O - D)$ pair let $p_h$ denote the $h$-th path from source $O$ to destination $D$ denoted as $p_h = (v^h_0, v^h_1), (v^h_1, v^h_2), (v^h_2, v^h_3), \ldots (v^h_{L_h-1}, v^h_{L_h})$, where $v^h_i \in V$ is the $j$-th visited node in the $h$-th path, with $v^h_0 = O$ and $v^h_{L_h} = D$. Additionally, let $d^h_t$ denote the arrival time at junction $v_j$ then, the arrival time to each node of the path can be expressed as:

$$d^h_{v^h_0} = t_0,$$

$$d^h_{v^h_t} = d^h_{v^h_{t-1}} + c_{v^h_{t-1}, v^h_t}(d^h_{v^h_{t-1}}, t_0)$$

$$\vdots$$

$$d^h_{v^h_{L_h}} = d^h_{v^h_{L_h-1}} + c_{v^h_{L_h-1}, v^h_{L_h}}(d^h_{v^h_{L_h-1}}, t_0)$$

(5)

with, $w \geq 0$ see Equation (2).

Given a vehicle routing request time $t_0$ and the admissible sets $S_{ij}(t_0)$, $\forall (i, j) \in \mathcal{E}$, the EATD problem requests the earliest-destination-arrival-time (from $O$ to $D$) subject to avoiding links that are at their critical density. Hence, the EATD problem can be expressed in compact form as:

$$d^*_D = \min_{w, p_h} d^h_D$$

s.t. Constraints (2) – (5).

In the next section we develop a method for predicting the travel-time of road segments, before examining the solution of the EATD problem.

### IV. Multiple Linear Regression Predictor Method

Linear regression is a statistical technique that considers historical observations to forecast near-future states. In this work, we employ a simple prediction method based on the Multiple Linear Regression technique to predict the transit-times of road segment using the most recent transit-time observations in each road segment. Evidently, as the density of a particular road segment increases, the speed of vehicles traversing the link decreases [35] which further results in increased travel times. Nonetheless, other factors also affect travel times such as the network traffic state, road junction priorities, the road geometry and the weather conditions. In this work, in addition to the density of the traversed road segment we explicitly consider the densities of neighboring segments (i.e., the road segments forming the down-link intersection). Other factors affecting travel times are considered implicitly by adopting a time-varying prediction approach that employs the most “recent” collected observations to capture the short-term traffic dynamics achieving more accurate predictions.

The Time-Varying Multiple Linear Regression (TVMLR) prediction method uses the $H$ most recent measurements of the response variable, i.e., the transit-time of the road segment of interest, and the predictor...
variables, i.e., the observed densities of the road segment of interest and its neighbors, to construct the best linear relationship between the two variable sets using the vector of regression coefficients.

To simplify notation, the TVMLR prediction method is described for a single road segment, say \((p, j)\). Let \((i, j)\), \(i \in P_j\) denote the segments whose down-link intersection is \(j\), and \(|P_j|\) the number of such links. Let also \(y_k\) and \(\hat{y}_k\) denote the observed and predicted transit-time of the \(k\)-th vehicle traversing \((p, j)\), respectively; while \(y\) is the response variable of the TVMLR prediction method. Assuming that the \(k\)-th measurement is collected at time \(t_k\), the number of vehicles in link \((i, j)\) at time \(t_k\) is denoted by the variable \(n_k = \rho_{ij}(t_k) \cdot l_{ij}, i \in P_j, i = 1, \ldots, |P_j|\), value that can also derived from the reports obtained from each vehicle. Inasmuch as, each vehicle reports to the RSU whenever it exits a certain road segment, and considering that, vehicles’ routes are known from the RSU, an accurate calculation of the number of vehicles in each segment can be maintained by simply increasing/decreasing the density value whenever a vehicle enters/exits a segment, i.e.,

\[
n_k \leftarrow \begin{cases} 
\rho_{ij}(t_k) \cdot l_{ij} + 1, & \text{if vehicle enters } (i, j), \\
\rho_{ij}(t_k) \cdot l_{ij} - 1, & \text{if vehicle exits } (i, j), \\
\rho_{ij}(t_k) \cdot l_{ij}, & \text{otherwise,}
\end{cases}
\]

assuming that \(\rho_{ij}(0) = 0\).

In this way one obtains the measurement vector \(y_H^K = [y_0^K, \ldots, y_{K-H+1}^K]^T\), and density matrix \(N_H^K = [n_0^K, \ldots, n_{K-H+1}^K]^T\), where \(K\) is the latest vehicle that traversed link \((p, j)\), and \(n_k = [n_0, \ldots, n_{|P_j|}]^T\). Using this information, the TVMLR prediction method finds the regression vector \(\hat{\beta}\) that minimizes the least square residual error by solving the optimization problem

\[
\min_{\beta} \sum_{k=K-H+1}^{K} \epsilon_k^2 = (y_H^K - N_H^K \beta)^T (y_H^K - N_H^K \beta) \tag{7}
\]

Whenever a new route observation request arrives, the method identifies for each road segment \((i, j)\) in \(\mathcal{E}\) the best regression vector in the least square sense, \(\hat{\beta}_{ij}\). Using the resulting regression vectors one can obtain the predicted travel time of road segment \((p, j)\) for the \(K+1\)-th vehicle traversing the segment, given by

\[
\hat{\tau}_{pj} = \left[1 \ n_{1,K} \ \cdots \ n_{{|P_j|,K}}\right] \hat{\beta}_{pj}.
\]

Note that the initial values of the predicted road segment travel-times are set equal to those corresponding to free-flow speed, i.e., \(\hat{\tau}_{pj} = l_{pj}/v^f\).

Next, we present a MILP formulation that utilizes the predicted travel times to solve the EATD problem. This is an NP-complete problem as shown in [2], and thus the MILP formulation can only solve small instances of the problem. In the subsequent section we present a low-complexity heuristic algorithm for obtaining a solution in real-time.

V. MIXED INTEGER LINEAR FORMULATION OF EATD PROBLEM

To optimally solve the EATD problem, we develop a Mixed Linear Programming Formulation (MILP) that selects road segments respecting the admissibility condition in order to connect the origin \(O\) with the destination \(D\) thus to minimize the arrival time at \(D\). To do so, the optimal path from \(O\) to \(D\) is described through the binary variables \(x_{ij}\), \((i, j) \in \mathcal{E}\), denoting whether a road segment \((i, j)\) is part of the optimal path \(p^*\) (\(x_{ij} = 1\)) or not (\(x_{ij} = 0\)). In addition, the auxiliary binary variables \(\psi_{ijk}\), \((i, j) \in \mathcal{E}, k \in K_{ij}(t_0)\) are introduced to indicate whether a road segment \((i, j)\) satisfies the admissibility condition at the \(k_{th}\) time interval (\(\psi_{ijk} = 1\)) or not (\(\psi_{ijk} = 0\)). The mathematical formulation for the EATD problem can be expressed as follows:

\[
\left(\text{P}_1\right) \min_{x_{ij}, \psi_{ijk}, d_t, \psi_{i,j,k}} d_D \tag{10a}
\]

s.t. \(\sum_{(i,j) \in \mathcal{E}} x_{ij} - \sum_{(j,i) \in \mathcal{E}} x_{jt} = \left\{ \begin{array}{ll} 1, & \text{if } i = O, \\
-1, & \text{if } i = D, \\
0, & \text{if } i \in \mathcal{V} \setminus \{O, D\} \end{array}\right. \)

\[
\sum_{k \in K_{ij}(t_0)} \psi_{ijk} = x_{ij}, \ (i, j) \in \mathcal{E}, \tag{10b}
\]

\[
d_t - d_j + \hat{\tau}_{ij}(t_0) \leq M_1 (1 - x_{ij}), \ (i, j) \in \mathcal{E}, \tag{10c}
\]

\[
d_t - d_j + \hat{\tau}_{ij}(t_0) \geq M_2 (1 - x_{ij}), \ (i, j) \in \mathcal{E}, \tag{10d}
\]

\[
d_t \geq t_{ijk} \psi_{ijk}, \ (i, j) \in \mathcal{E}, \ k \in K_{ij}(t_0), \tag{10e}
\]

\[
d_j \leq t_{ijk} \psi_{ijk} + M_3 (1 - \psi_{ijk}), \ (i, j) \in \mathcal{E}, \ k \in K_{ij}(t_0), \tag{10f}
\]

\[
d_t \geq t_0, \ i \in \mathcal{V} \setminus \{O\}, \ d_t \geq t_0, \ i = O, \tag{10g}
\]

\[
x_{ij} \in \{0, 1\}, \ (i, j) \in \mathcal{E}, \tag{10h}
\]

\[
\psi_{ijk} \in \{0, 1\}, \ (i, j) \in \mathcal{E}, \ k \in K_{ij}(t_0). \tag{10i}
\]

where \(M_1, M_2\) and \(M_3\) are appropriately selected constants. In the above formulation, equality (10b) describes the flow constraints that ensure connectivity of the optimal path from source to destination, while (10c) forbids the traversal of link \((i, j)\) at any time if \(x_{ij} = 0\). Constraints (10d) and (10e) ensure the logical condition “if \(x_{ij} = 1\) then \(d_t = d_i + \hat{\tau}_{ij}(t_0)\)” which describes the cost increase when traversing road link \((i, j)\). To examine the validity of the condition, notice that for \(x_{ij} = 1\) constraints (10d) and (10e) enforce the logical equality \(d_t = d_i + \hat{\tau}_{ij}(t_0)\). For \(x_{ij} = 0\), constraints (10d) and (10e) should have no effect on the optimization problem; for this reason, constants \(M_1\) and \(M_2\) are selected to provide tight upper and lower bounds on \(d_t - d_j + \hat{\tau}_{ij}(t_0)\), respectively, such that the resulting inequalities are always true. Letting \(d^u\) denote an upper bound to the solution of the EATD problem (e.g., obtained
through a heuristic algorithm), such that $-d^u \leq d_i - d_j \leq d^u$, yields the bounds $M_1 = d^u + \max_{(i,j) \in E} \{ \tilde{t}_{ij}(t_0) \}$ and $M_2 = -d^u + \min_{(i,j) \in E} \{ \tilde{t}_{ij}(t_0) \}$. In a similar fashion, we can deduce that constraints (10f) and (10g) are equivalent to the logical condition “if $\psi_{ijk} = 1$ then $\tilde{t}_{ijk} \leq d_i$ and $d_j \leq \tilde{t}_{ijk}$,” indicating that when $\psi_{ijk} = 1$ then the admissibility condition needs to be satisfied for link $(i,j)$, and time interval $k$. Note that $M_3$ needs to provide an upper bound to $d_j$, hence $M_3 = d^u$. Constraints (10h) and (10j) simply denote the nature (e.g. continuous, binary) and range of each set of variables. Note that the origin waiting time (i.e., $w$), is implicitly imposed by letting $d_O \geq t_0$, so that $w = d_O - t_0$.

Problem $P_1$ is a MILP problem that can be solved with standard optimization solvers, yielding the optimal solution to the EATD problem. Nonetheless, the mixed-integer nature of the formulation implies that in certain cases the MILP solver may need exponentially large time to complete. For this reason, we also develop a close-to-optimal low-complexity heuristic in the next section.

VI. ITERATIVE DIJKSTRA-BASED ALGORITHM IN CONTINUOUS-TIME

To provide a polynomial-time solution, in this section we describe the Iterative Algorithm in Continuous-Time (IDAC). IDAC is a heuristic algorithm that solves the EATD problem over a sequence of iterations by employing a variant of Dijkstra’s shortest path algorithm. A discrete-time version of the algorithm was originally developed in [2] where reservations are made for discrete time slots assuming constant speed equal with the free-flow speed.

Algorithm 1 Iterative Dijkstra-based Algorithm in Continuous-Time.

Input: $G(V,E), S_{ij}(t_0), O, D, t_0$

Initialization: $m = 0; w^p_0 = 0$

repeat
  $m \leftarrow m + 1$
  \[ [w^T, P] \leftarrow \text{relaxed-EATD}(G(V,E), S_{ij}(t_0), O, D, t_0, w^{p}_{m-1}) \]
  $w^{p}_m \leftarrow w^{p}_{m-1} + w^T$
  until \{ $w^T = 0$ \}

Output: $P$ and $w^{p}_m$.

In this work we remove this assumption and also illustrate how the algorithm can be implemented in continuous-time. At the $m$-th iteration of the algorithm a relaxed-version of the EATD problem (Relaxed-EATD) is solved where waiting is allowed at all nodes. Once a route is computed, the algorithm checks if any waiting is necessary at intermediate nodes and updates the route starting time (i.e., $w^p_m = w^{p}_{m-1} + w^T$, where $w^T$ is the total waiting at all nodes from the solution of the Relaxed-EATD) before reiterating. The algorithm finally terminates when a route is computed with no waiting at any of the intermediate nodes. Note that the Relaxed-EATD problem can be optimally solved so that its solution provides a lower bound to the original problem.

Algorithm 2 Solution of the Relaxed-EATD problem.

1: Input: $G(V,E), S_{ij}(t_0), O, D, t_0, w^p$
2: Initialization: $d_i = \infty, i \in V, d_O \leftarrow t_0 + w^p, Q \leftarrow V$
3: $P[i] \leftarrow \text{NULL}, i \in V$
4: while $Q \neq \emptyset$ do
5:   $l \leftarrow \text{argmin}\{d_i\}; Q \leftarrow Q - \{l\}$
6:   for $(i,j) \in E$ do
7:     $w_{lj} \leftarrow \min\{d_l + w \in S_{lj}(t_0), 0 \leq t \leq \hat{t}_{lj}(t_0)\}$
8:   end for
9:   if $d_i > d_l + c_{lj}$ then
10:      $d_i \leftarrow d_l + c_{lj}, P[j] = l$
11: end if
12: end while
13: $w^T \leftarrow d_D, v \leftarrow D$
14: repeat
15:   $w^T \leftarrow w^T - c_{P[v], v}; v \leftarrow P[v]$
16: until $v = O$
17: Output: $w^T, P$

The proposed Dijkstra variant is described in Alg. 2. Each computed route is expressed through the predecessor list $P$ together with the necessary waiting $w^p_m$ that needs to be incurred at the origin before commencing the trip. At each iteration, the arrival time $d_i$ of each node is set to infinity, except of the source node whose arrival time is equal to $t_0$ (the time that the request was made) plus any waiting time incurred in previous iterations of the algorithm. The algorithm maintains a set $Q$ that contains all nodes that have been permanently labeled, i.e. the nodes whose shortest distance from the source node has been found. Initially all junction nodes are non-labeled and thus the respective set contains all nodes, $Q \leftarrow V$. Also, the predecessor list $P$ is set to null (lines 2-3).

Subsequently, junction $I$ with the earliest arrival time is permanently labeled (i.e. the arrival time to that junction is set) (line 5) before checking if the arrival time to each neighbor of $l$ can be improved (lines 6-12). Checking is simply done by examining if the waiting time $w_{lj}$ to move from junction $l$ at time $d_l$ to junction $j$ is smaller than the existing arrival time at junction $j$. If the condition holds, $d_j$ is updated and $l$ is set as the predecessor of $j$ (lines 8-11). The aforementioned procedure repeats until all junctions nodes are labeled. By the end of the procedure, the predecessor list will hold the best route from $O$ to $D$ and can be used to calculate the total waiting $w^T$ that needs to be incurred along the route (lines 14-17). The complexity of the IDAC algorithm is $O(LE^2/\log(V))$ since the waiting intervals that may be required at any intermediate road junction can be calculated more efficiently through continuous-time intervals.
vehicle flow-rates vary between assuming Poisson arrivals and random over five Monte Carlo randomly generated scenarios, as shown in Fig. 3. All examined algorithms are simulated for two-way, single-lane road segments and 28 junctions, considering a Manhattan grid topology consisting of a lower number of Relaxed-EATD iterations are required (since no time intervals are wasted due to quantization).

VII. Simulation Results

Performance evaluation is performed in both ideal and micro-simulation environments. In ideal environment it is assumed that there is no uncertainty in the fundamental diagram and that the developed algorithms result in free-flow conditions for all road links. It is further assumed that the speed under free-flow conditions is constant and equal to the free-flow speed $v_f$. This means that the topology can be modeled as a graph with edge cost equal to the transit-time of the corresponding road segment with free-flow speed so that the path travel time is equal to the sum of the free-flow transit-times of all traversed links.

Micro-simulation environment aim to capture the stochastic nature of traffic mobility within a real road network which results in uncertainty in the fundamental diagram. Hence, the travel time is time-varying and thus predictions of road segment’s travel-times are required. Micro-simulations are performed using the Simulation Urban MObility (SUMO) micro-simulation software [36].

A. Ideal environment

In the ideal simulation environment the network under consideration is a Manhattan grid topology consisting of 36 two-way, single-lane road segments and 28 junctions, as shown in Fig. 3. All examined algorithms are simulated over five Monte Carlo randomly generated scenarios, assuming Poisson arrivals and random $O-D$ pairs. The total vehicle flow-rates vary between 1000 and 8000 veh/h, while a critical density of $\rho_{ij}^C = 30$ veh/km/lane and a free-flow speed of 15 m/s are considered.

In this context the performance of three algorithms is examined:

1.) **MILP**: The MILP formulation developed in Section V.
2.) **IDAC**: The heuristic approach developed in Section VI.
3.) **IDAD**: The discrete time version of the IDAC, proposed in [2] and referred to as Iterative Dijkstra-based Algorithm in Discrete-Time (IDAD).

Note that the MILP formulation is constructed and solved using the Gurobi mathematical programming solver [37], while the IDAC and IDAD are implemented in the C++ programming language.

Fig. 4 depicts the mean travel time that vehicles experience within the network for different demand flow rates. Travel time is defined as the time lapse between origin departure time and the destination arrival time. As anticipated, all algorithms experience approximate similar travel times in low demand flow rates (below 3000 veh/h) as there is no congestion. As the flow rate increases, the MILP experience higher average travel times than the IDAC and IDAD approaches. In terms of path length, MILP produces longer paths as the network travel time increases, while both the IDAC and IDAD paths appear to maintain constant travel times irrespective of the flow demand.

Figs. 5 and 6 show the origin waiting time and the total time (origin waiting time plus travel time) that vehicles experience for different demand flow rates, respectively. Despite the fact that the MILP approach schedules vehicles through longer paths, the total time is lower compared to both the IDAC and IDAD algorithms, because the former generates solutions with shortest origin waiting times. More specifically, at low flow-rates (1000 - 5000 veh/h) the performance of all algorithms is almost identical as there are no significant restrictions in terms of road segment admissibility. However, at higher demands the MILP approach outperforms the IDAC and IDAD by up to 20% and 30%, respectively, as the admissibility sets become more fragmented and require the examination of a large number of time-interval combinations to find the best path. Fig. 5 also illustrates that as the demand flow rate increases, the average waiting time increases exponentially and becomes more than one order of magnitude larger than the mean travel time. This figure also indicates that the origin waiting time for IDAD algorithm is up to 10% larger compared to the IDAC, since IDAD may reserve more time-slots than it actually needs due to inaccuracies in time discretization. Despite the large origin waiting times, we demonstrate that under micro-simulation environment all proposed algorithms lead to significantly better performance compared to the uncontrolled scenario (US) which yields higher travel times due to congestion.

An interesting observation is that the MILP approach may impose virtual waiting within network by introducing cycles to the vehicle paths. Although such paths do have smaller total times, this often leads to higher fuel consumption, which may be undesirable. On the contrary, the IDAC and IDAD approaches introduce all waiting at the origin, as they are based on Dijkstra’s algorithm which prohibits cycles in the produced paths.

Fig. 7 examines the average execution time of all algorithms for different flow rates. As expected, higher demand flow rates lead to longer execution times for all algorithms because the admissibility sets become more fragmented. Clearly, the IDAC and IDAD algorithms are significantly faster than the MILP approach, outperforming the latter by around three orders of magnitude. Another interesting observation regards the execution speed of the two
Dijkstra-based algorithms, as no algorithm is dominated by the other. In particular, IDAD appears faster than the IDAC for low flow rates and slower than the IDAC for high flow rates. The increase in the execution time of IDAD at higher flow rates is possibly due to decreased road segment admissibility, as a result of unnecessary reservations in discrete time, which leads to longer horizon problems to be solved. Since, the IDAC algorithm achieves the lowest execution times with near-optimal performance it has also been selected for performance evaluation in micro-simulations.

B. Micro-simulation Environment

To examine the performance of the developed algorithms under micro-simulation conditions we consider the SUMO. In SUMO the traffic mobility characteristics are determined by the Krauss car-following model [38] with the selected parameters: vehicle length 5 m, maximum speed 15 m/s, acceleration 2.5 m/s², deceleration 4.5 m/s², driver imperfection 5%, driver reaction time 0.5 s, minimum gap distance 2.5 m, and simulation time-step 0.5 s. Due to the stochastic nature of SUMO all obtained results are averaged over ten (10) Monte Carlo simulations. In the simulation scenarios that follow it is assumed that all instructions (route, origin waiting time) issued by the RSU to vehicles are followed without any deviation. It is important to note that in SUMO environment, even under free-flow conditions the vehicle speeds vary significantly due to various random events that occur due to acceleration and deceleration of vehicles, and queuing delays at intersections due to the passing of higher priority vehicles.

The network under consideration is an 1.8 km² unsignalized homogeneous region of down-town San Francisco as depicted in Fig. 8 (consisting of 99 road junctions and 208 single-lane road segments with lengths varying from 100 m to 400 m. Note that, the spatial compactness and homogeneity of this area was initially investigated in [39], [40]. In this context, four algorithms are examined:

1.) IDAC: The heuristic approach developed in Section VI combined with the TVMLR prediction method proposed in Section IV.

2.) IDACNP: The IDAC algorithm without the TVMLR prediction method. In this case, the IDAC algorithm considers the free-flow speed $v_f$ of each road segment for each vehicle.

3.) DOT: The Decreasing Order of Time (DOT) algorithm [41]. The DOT algorithm seeks to find the time-dependent travel-time path that minimizes the user’s arrival time at the destination within a user-specified departure time window.

4.) US: The uncontrolled scenario where each vehicle travels along its shortest-distance path without any waiting time at the origin.

Note that in the case of both IDAC and IDACNP, new
route reservations are computed using only information from previous reservations without any consideration of the actual network state.

**NFD Identification**

As the proposed algorithms rely on the NFD of the region of interest, we first generate the NFD of the considered road network to identify the region’s critical density and free-flow speed \((\rho_{ij}^C \text{ and } v_f)\) parameters. 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, for the last two hours the input flow was set to 4000 veh/h and 2000 veh/h in order to allow a network discharge.

Fig. 9 shows the total network flow as a function of the network’s density (i.e. the total number of vehicles within the region) for the US, which depicts the NFD diagram of the region. In the figure, each point corresponds to the sliding mean of 5 measurements that are calculated every 15 s. To calibrate the model, the Van-Aerde automated calibration method proposed in [42] is employed. The procedure produced the following calibrated parameters \(v_f = 42.5 \text{ km/h}, \ v_c = 27.5 \text{ km/h}\) and \(\rho_f = 675\) vehicles, which correspond to the red line depicted in the figure. Having obtained the calibrated model one can analytically obtain the critical density which in our case is equal to \(\rho_{ij}^C = 33 \text{ veh/km/lane}\) (i.e., around 33% of the region’s jam density); accordingly, the vehicle speed that will be used for the IDACNP is \(v_f = 42.5 \text{ km/h}\).

Figs. 10 (a) and (b) compare the accuracy of reservations for the IDACNP and IDAC algorithms over the 6-hour scenario. In particular, the figures depict the actual and predicted instantaneous network density over time with red and green lines, \(R(t)\) and \(N(t)\), respectively, while the blue dotted line represents the instantaneous residual density, \(E(t) = |R(t) - N(t)|\). During the first two hours of the considered scenario, the predicted density, through reservations, closely follows the actual network density, as the demand flow rate that enters the network is low. As the demand increases over time, the network becomes more congested which results in a significant growth of the residual density when using the IDACNP; on the contrary, the IDAC manages to maintain an excellent prediction of the network density over the entire 6-hour scenario, due to the integration of the TVMLR prediction method.

Figs. 11 (a) and (b) depict the maximum instantaneous residual density of individual road segments for the (a) IDACNP and (b) IDAC. Comparing both approaches, the IDACNP is more accurate than the IDAC method as it can achieve slightly better than the IDAC approach, the IDAC manages to maintain an excellent prediction of the network density over the entire 6-hour scenario, due to the integration of the TVMLR prediction method.

Figs. 12 (a) and (b), respectively. The figures illustrate acceptable performance, as both methods maintain the network operation within the non-congested region. Comparing both approaches, the IDACNP makes reservations that underestimate the transit time of each road segment allowing a little bit higher rate to pass through the network.

**Performance Evaluation**

The four aforementioned methods are evaluated for various scenarios and varying demand flow rates (3000 – 8000 veh/h) over a 2 hours simulation period where the demand is constant for the first hour and equal to zero for the second hour. For a fair comparison, the origin waiting
Fig. 13: (a) Travel time and (b) Number of vehicles with completed journeys for different simulation scenarios with varying demand flow rate.

Fig. 14: Per vehicle travel time distribution for the highest demand flow rate (i.e., 8000 veh/h).

time for all approaches is not considered in the travel time and for DOT the maximum allowed origin waiting time is set to 1 minute (i.e., half the average trip length for the considered network).

Figs 13 (a) and (b) depict the average travel time of all vehicles and the total number of vehicles that have managed to finish their journey within the simulation time, respectively. In the figures dashed lines indicate the average values over the five simulations run, while the markers indicate each individual simulation run value. From the figures it is clear that for low demand flow rates (under 6000 veh/h) all approaches have similar behavior with the US because no congestion occurs in the network. Nonetheless, under congested conditions (demand flow rates larger or equal to 6000 veh/h) the behaviour of the different algorithms varies significantly. With regards to travel time, Fig. 13 (a) indicates that both the IDACNP and IDAC significantly outperform both DOT and US approaches. As congestion intensifies the travel time of both DOT and US grows exponentially, while the travel time of the IDACNP increases only for the highest demand scenario. Interestingly, the IDAC exhibits constant travel time in all scenarios considered which indicates that it is robust to different demand levels. With respect to the total number of vehicles that have managed to finish their journey within the simulation time, Fig. 13 (b) demonstrates that for all demands the IDACNP and IDAC manage to complete all the assigned trips within the simulation time, while in the case of US and DOT, vehicles experience major delays due to congestion and as a result a large percentage of vehicles do not complete their journeys.

This observation is also supported from the distribution of the vehicle travel times for the highest demand flow rate (8000 veh/h) shown in Fig. 14. As can be seen, the IDAC manages to complete more vehicles than all the other algorithms for all bins that correspond to travel times smaller than 125 s.

Figs 15 (a) and (d) compare the accuracy of reservations for the IDACNP and IDAC algorithms over the 2 hours scenario. The figure supports the observations made in Figs. 10, as the instantaneous residual network density over time grows significantly for the IDACNP, while for the IDAC the instantaneous residual network density is very close to zero. The superior reservation accuracy of the IDAC over IDACNP is also shown from Figs. 15 (b) and (e), as well as Figs 15 (c) and (f) which depict the instantaneous residual density of individual links and the maximum residual density of individual links over time, respectively. More specifically, the average residual density of the IDAC is approximately three times smaller compared to the IDACNP (8 vehicles compared 25 vehicles per road segment, respectively). Figs 15 (c) and (f) further indicate that more than one third of network’s segments exhibit high maximum residual density (more than 20 vehicles) when using the IDACNP, while for the IDAC only two road segment exhibit this behaviour.

Figs 16 (a) and (b) illustrate the origin waiting time assigned to vehicles before commencing their trip for the IDACNP and IDAC solutions in the form of a box-plot. For both cases, as demand grows, the origin waiting time increases since more vehicles request to enter the network, with the IDAC responding accordingly at higher input rates as opposed to the IDACNP solution. More specifically, the origin waiting time assigned by the IDAC is about 4 times higher than the IDACNP approach. Nonetheless, the average origin waiting is within acceptable levels (14 min for the highest demand approach), while origin waiting times observed at the origin do not impact travel times. This behavior is expected because the IDACNP allows more vehicles to enter the network compared to the critical capacity (see Fig. 15 (a)) so that the origin waiting time of the vehicles is smaller.

VIII. IMPLEMENTATION CHALLENGES

Prior the implementation of the proposed route-reservation methodology the following challenges need to be addressed:

1. Drivers adhering: The route reservation architecture is well suited for connected-autonomous vehicles. On the other hand if human drivers are considered, drivers may not adhere to the RSU’s instructions. To anticipated this issue in future work, we will consider scenarios that both normal and autonomous drivers coexist and we will investigate the maximum allowable percentage of non-adhering drivers thus to provide travel time guarantees and congestion free operation.

2. Privacy: Under the proposed architecture drivers are asked to provided private and sensitive information.

The bottom and top of each box indicate the first and third quartiles (25% and 75%) of a ranked data set, while the horizontal line inside the box indicates the median value (second quartile). The horizontal lines outside the box indicate the lowest/highest datum still within 1.5 inter-quartile range of the lower/upper quartile.

The bottom and top of each box indicate the first and third quartiles (25% and 75%) of a ranked data set, while the horizontal line inside the box indicates the median value (second quartile). The horizontal lines outside the box indicate the lowest/highest datum still within 1.5 inter-quartile range of the lower/upper quartile.
to the RSU. Such as the OD pair and the path to be followed, possible solutions include the pre-specified OD pairs such that users will not reveal their identity.

3. Fairness: In this work vehicles are served in first come first served order, enabling system susceptible to fairness problems as some vehicles maybe instructed through longer routes instead to follow the shortest ones or some of them maybe forced to wait longer than others. A possible solution to this problem is to collect multiple reservation requests and then solve the problem for the entire batch taking into consideration the previous routing decisions.

IX. CONCLUSIONS AND FUTURE RESEARCH

This work extends the route reservation architecture proposed in [2] to investigate the EATD problem in continuous time and develop prediction methods for more accurate estimation of the time needed to traverse different road segments for better route reservations. In this context, the TVMLR prediction method is developed, a time-varying multiple linear regression method that takes into account the densities of neighbouring road segments affecting the exit of a vehicle from a certain road segment. For the solution of the EATD problem in continuous time, a novel MILP formulation is constructed that optimally solves the considered problem using specialized optimization solvers. A heuristic approach is also developed based on a customized version of Dijkstra's shortest path algorithm that provides fast and close-to-optimal solutions. Extensive performance evaluation confirms the usefulness of the continuous-time approach to solve the EATD problem as it leads to better results compared to algorithms proposed previously. The results further indicate the importance of the TVMLR prediction method to ensure proper utilization of the infrastructure capacity that leads to faster trips.

Future research directions include the extension of the IDACNP and IDAC methods to operate in larger networks. This can be done, if the route reservation problem is solved in distributed manner. Under this setting, a multi-region network should be considered where each region will be controlled by a single RSU. Further, a load balancing framework can also be applied in order to balance the utilization between each region and to manage the vehicles distribution across the boundaries of neighboring regions.

Notably, the proposed route reservation architecture does not consider any incidents or any vehicle reservations
cancellations. Having said this, in future work formulations we plan to also consider accidents and allow to reschedule vehicles on demand thus reacting to accidents, early and late vehicles departures and also to taking into consideration reservation cancellations.

Additionally, future work should also includes, the examination of the proposed route reservation method when only a portion of the drivers adheres to the RSU schedules to fulfill the implementation challenges constraints.

REFERENCES


Charalambos Menelaou received a B.Sc. degree from the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus, in 2013, where he is currently pursuing the Ph.D. degree. He is also a researcher at KIOS Research and Innovation Center of Excellence at the University of Cyprus. His research interests focus on large scale urban transportation systems, traffic flow control and optimization, and on dynamic routing of autonomous vehicles. Mr. Menelaou is an executive committee member of the IEEE student branch of University of Cyprus and a member of IEEE Intelligent Transportation Society, since 2014.

Stelios Timotheou (S’04-M’10-SM’17) received a B.Sc. from the Electrical and Computer Engineering (ECE) School of the National Technical University of Athens, and an M.Sc. and Ph.D. from the Electrical and Electronic Engineering Department of Imperial College London. He is currently a Research Associate at the KIOS Research Center for Intelligent Systems and Networks of the University of Cyprus (UCY). In previous appointments, he was a Visiting Lecturer at the ECE Department of UCY, a Research Associate at the Computer Laboratory of the University of Cambridge and a Visiting Scholar at the Intelligent Transportation Systems Center & Testbed, University of Toronto. His research focuses on the modeling and system-wide solution of problems in complex and uncertain environments that require real-time and close to optimal decisions by developing optimization, machine learning and computational intelligence techniques.

Panayiotis Kolios Panayiotis Kolios is a member of the KIOS Research and Innovation Center of Excellence at the University of Cyprus. His interests focus on both basic and applied research on networked intelligent systems. Some examples such systems include intelligent transportation systems, autonomous unmanned aerial systems and the plethora of cyber-physical systems that arise within the Internet of Things. Particular emphasis is given to emergency response aspects in which faults and attacks could cause disruptions that need to be effectively handled. He is an active member of IEEE, contributing to a number of technical and professional activities within the Association. He received his B.Eng and Ph.D degrees in Telecommunications Engineering from King’s College London in 2008 and 2011, respectively.

Christos G. Panayiotou is an Associate Professor with the Electrical and Computer Engineering (ECE) Department at the University of Cyprus (UCY). He is also the Deputy Director of the KIOS Research Center for Intelligent Systems and Networks for which he is also a founding member. Christos has received a B.Sc. and a Ph.D. degree in Electrical and Computer Engineering from the University of Massachusetts at Amherst, in 1994 and 1999 respectively. He also received an MBA from the Isenberg School of Management, at the aforementioned university in 1999. Before joining UCY in 2002, he was a Research Associate at the Center for Information and System Engineering (CISE) and the Manufacturing Engineering Department at Boston University (1999 - 2002). His research interests include distributed and intelligent control systems, wireless, ad hoc and sensor networks, computer communication networks, fault diagnosis, optimization and control of discrete-event systems, resource allocation, transportation networks and intelligent buildings. Christos has published more than 190 papers in international refereed journals and conferences and is the recipient of the 2014 Best Paper Award for the journal Building and Environment (Elsevier). He is an Associate Editor for the Conference Editorial Board of the IEEE Control Systems Society, the IEEE Transactions on Control Systems Technology, the Journal of Discrete Event Dynamical Systems and the European Journal of Control. He held several positions in organizing committees and technical program committees of numerous international conferences.

Marios M. Polycarpou is a Professor of Electrical and Computer Engineering and the Director of the KIOS Research Center for Intelligent Systems and Networks at the University of Cyprus. He received the B.A degree in Computer Science and the B.Sc. in Electrical Engineering, both from Rice University, USA in 1987, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, in 1989 and 1992 respectively. His teaching and research interests are in intelligent systems and networks, adaptive and cooperative control systems, computational intelligence, fault diagnosis and distributed agents. Dr. Polycarpou has published more than 300 articles in refereed journals, edited books and refereed conference proceedings, and co-authored 7 books. He is also the holder of 6 patents. Prof. Polycarpou is a Fellow of IEEE and IFAC. He is the recipient of the 2016 IEEE Neural Networks Pioneer Award and the 2014 Best Paper Award for the journal Building and Environment (Elsevier). He has served as the President of the IEEE Computational Intelligence Society (2012-2013), and as the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems (2004-2010). He is currently the Vice President of the European Control Association (EUCIA). Prof. Polycarpou has participated in more than 60 research projects/grants, funded by several agencies and industry in Europe and the United States, including the prestigious European Research Council (ERC) Advanced Grant.