A Survey of Models and Algorithms for Intelligent Transportation and Traffic Management

Alfred M. Bruckstein, Roee M. Francos, Dmitry Rabinovich
Department of Computer Science, Technion
freddy@cs.technion.ac.il

I. Goal of the Survey:

In the ever changing modern world, transportation plays a vital role in our daily lives. For decades considerable effort was devoted by the scientific community to address various aspects of transportation. Various advancements in communication, control, sensing, distributed algorithms and information systems, set up the grounds for the development of Intelligent Transportation Systems (ITS) whose goals are to improve travel safety, increase mobility and reduce environmental impacts. ITS research spans multiple disciplines: from operational research and computer science to civil and transportation engineering, micro-machinery and behavioral sciences. Successful advancement of ITS will be accompanied by breakthroughs in the design of individual ground, air and marine vehicles as well as substantial improvement in distributed and centralized management algorithms, protocols and capabilities of multiple vehicles. Growing urbanization has created an urgent need for safer, faster and more eco-friendly transportation. We expect this need to be met by the progress in the deployment of autonomous vehicles that will bring a dramatic change to the transportation systems as we know them. In this review we analyze the current state of modeling and algorithms for ITS and present the main topics for future research.

II. Overview of the Field

In our overpopulated world, urbanization has been a global trend for quite some time. Numerous people choose to leave rural regions and make urban areas as their primary center of living. Human concentrations in urban centers led to an enormous pressure on the existing infrastructure, as vehicle fleets expand at a faster rate than roads are being constructed, and traffic congestion becomes more and more severe. The transportation systems of the future will have to address these issues in an innovative and smart manner. Unsurprisingly, Intelligent Transportation Systems (ITS) are believed to be our last chance to prevent an imminent transportation collapse. ITS will hopefully increase safety, decrease the number of traffic accidents, cut down high maintenance costs and free lands occupied by traffic related infrastructure. Reducing traffic congestion alone would drastically improve quality of life, lower energy consumption and lessen negative environmental impacts.

An anticipated vehicle autonomy and the development of smart infrastructure for autonomous vehicles (AV) will help future transportation systems to handle service demands efficiently. Smart connectivity is an another pillar associated with ITS. It is expected that vehicles, smart infrastructure and human users will exchange valuable transportation data.

Intelligent Transportation Systems should include advanced and comprehensive transportation management and service systems which builds upon sharing of information and communication between entities that are acting in a well designed environment. According to the European Union directive 2010/40/EU [A7], ITS aim to "provide innovative services relating to different modes of transport and traffic management and enable various users to be better informed and make safer, more coordinated and 'smarter' use of transport networks". ITS incorporate and combine advancements in high technological fields ranging from control, information systems, computer vision, signal processing, communication to machine learning and mathematics in order to advance the state of traditional transportation systems and make them suitable for the needs of the smart city of the future.

ITS goals, as are outlined by the United States Department of Transportation (USDOT) in [A3], can be broadly classified to enhancing mobility, providing sustainable transportation and improving the convenience of services to transportation system users. Better mobility can be achieved by exploring methods and management strategies that increase system efficiency and by improving individual mobility. Sustainability emphasizes safety by development of safer vehicles and roadways, by developing better crash avoidance and driver assistance mechanisms, and by developing smart infrastructure based on cooperative safety systems. Additionally, sustainability is concerned with limiting environmental impacts by better management of traffic flow and by transitioning to green transportation alternatives such as electric vehicles.

[1]Alfred M. Bruckstein, Roee M. Francos and Dmitry Rabinovich are with the Faculty of Computer Science, Technion- Israel Institute of Technology, Haifa, Israel, 32000. freddy@cs.technion.ac.il, roee.francos@cs.technion.ac.il, dmitry.ra@cs.technion.ac.il

ITS are expected to have a hierarchical structure where each layer is responsible for performing different tasks. In [A6], Lin et al. consider ITS to be constructed of 4 basic layers: a physical layer, a communication layer, an operation layer and a service layer. Where the physical layer is at the bottom of the hierarchy and the service layer is at the top. The physical layer is composed of the transportation system entities such as vehicles, infrastructure and people. The communication layer is responsible for the exchange of information between all components of the ITS. The operation layer aggregates and processes all the data it gathers from the ITS, translates it into information and sends it back to the physical layer in the form of services in the service layer. The three fundamental components that constitute the operation layer are: Advanced Transportation Management Systems (ATMS), Advanced Traveler Information Systems (ATIS) and Advanced Vehicle Control Systems (AVCS). ATMS is concerned with ITS management and aims mainly at improving traffic flow and safety. ATIS is responsible for acquisition, processing and information provision to travelers. AVCS consists of novel control technologies that turn vehicles into smart and intelligent. The service layer is responsible for executing ITS services by combining the outputs from the operation layer with the goal of improving transportation services.

In the process of preparing this survey we investigated previous ITS surveys [A1, A2, A4, A5, A6, A8]. Additionally, we identified major conferences and journals concerned with ITS or closely related to it. From these conferences, journals and previous surveys that concerned various aspects of ITS we distilled core research areas of ITS. In the next sections we review the topics that were investigated in this survey.

III. Freeway Traffic Modeling

A number of approaches to model freeway traffic were considered by the community throughout the years. Macroscopic models, where traffic is viewed as a compressible fluid formed by vehicles, and microscopic models, where individual vehicles are considered to be interacting particles have been developed since the middle of 20th century. Recently, a cellular automata (CA) models have become increasingly popular, due to low-computational requirements and the ability to reproduce numerous real-life traffic scenarios.

A. Microscopic car-following models

Car-following theory is based on the idea that each driver controls a car according to an input from the surrounding vehicles. Frequently, these are the preceding and the following vehicles, however a more complex behavior accounting for vehicles on the neighboring lanes was considered. In general, the input that influence the individual human driver behavior may include the distance between vehicles, velocity of the vehicle, acceleration of the vehicle and relative velocity. Consequently, different models have been proposed that consider a part or all of the above parameters.

1) General Motors (GM) model: The model was developed at the General Motors research laboratory in Detroit [B2]. The assumption behind the GM model is that a vehicle tend to adjust the speed to that of the leading vehicle. Therefore, an appropriate formulation of the model is

$$a_n(t+\tau) = k\Delta v_n(t), \quad n = 1, 2, \ldots, N, \tag{1}$$

where $a_n(t+\tau)$ is an acceleration of the nth vehicle implemented at time t after accounting for the driver response delay $\tau \geq 0$. $\Delta v_n(t) = v_{n+1}(t) - v_n(t)$ is the relative to the leading ($n+1$th) vehicle velocity at time t. N - is the total number of vehicles, and k - is the driver response intensity to unit stimulus.

Later Gazis et al. [B7] found that (1) could not explain the observed drivers behavior in high density traffic situations. The proposed amendment accounted for relative spacing and became

$$a_n(t+\tau) = k\frac{\Delta v_n(t)}{\Delta x_n(t)}, \tag{2}$$

where $\Delta x_n(t) = x_{n+1}(t) - x_n(t)$, and $x_i(t)$ - is the position of vehicle i at time t.

Finally, Edie [B4] generalized the model to account for own velocity. Therefore, in the most general form the model is now expressed as

$$a_n(t+\tau) = kv_n^m(t)\frac{\Delta v_n(t)}{\Delta x_n^l(t)}, \tag{3}$$

where m and l are non-negative parameters of the model. The key to the GM model application is then to specify these parameters. In the following years, numerous works explored the different combinations of m and l to find the “best” possible pair, however the findings were contradictory.

2) Collision Avoidance model: The Collision Avoidance model is also called the safety distance model, which is different from the above GM model. The model is based on (4), which describes the safe following distance required to avoid collision with the vehicle ahead. According to the model, if the distance is shorter than the safe distance the collision become unavoidable, assuming the unpredictable behavior of the leading vehicle.

$$\Delta x(t) = \alpha v_{n+1}^2(t) + \beta_1 v_n^2(t+\tau) + \beta_2 v_n(t+\tau) + \beta_3, \tag{4}$$

where $\alpha, \beta_1, \beta_2, \beta_3$ - are calibration constants of the model and other parameters are as in the GM model above.

The Gipps model [B8] is based on the Collision Avoidance model and is widely used in the microscopic traffic simulation software. The popularity of the model is attributed to the realistic behavior reported in the situations involving a platoon or a pair of vehicles. In the same times the basic assumption of the model is problematic. In fact, human drivers account for the behavior of several preceding vehicles to predict the actions of the leading driver.

B. Particle Hopping Models

The problem of traffic flow could be viewed as the large scale dynamics of interacting particle systems. Particle hopping models are characterized by the following features: a lane is represented as a one-dimensional grid of points. A "vehicle" can occupy a number of points on the grid, and it moves forward from its current location on the grid according to some rule. Most rules strive to implement a "crash free principle": no two "vehicles" should share points on the grid at any time. The model could have open or closed boundary rules. In the former a "vehicle" leaving from one side immediately appears on the other. This geometry is known as a one-dimensional torus geometry. In the latter, "vehicles" leave from one side, and new "vehicles" are introduced from the other side with initial speed preset to some constant speed.

1) Asymmetric Simple Exclusion Process: The simplest traffic model is the CA 184 [B21, B22] This interacting particles model has been investigated as an asymmetric simple exclusion process [B5, B17, B18]. Defined on a one-dimensional lattice of length L, particles perform an exchange with close neighbors in a biased in a specific direction manner. If such exchanges are restricted in one direction the process becomes totally asymmetric (TASEP).

The "motion" rule of the model is very simple: one of the "vehicles" is picked at random and moved forward by one site if that site is empty. Note, that the update rule sets the maximum possible speed to $v_{max} = 1$.

$$x_j(t+1) = x_j(t) + \min[1, x_{j+1}(t) - x - j(t) - 1], \tag{5}$$

where $x_j(t)$ is the position of the picked particle j at time t.

It is this random picking of "vehicles" that introduces stochasticity into the model and leads to a striking difference between TASEP models and models with deterministic dynamics produced by the parallel state update.

The extended CA model was later proposed by Fukui and Ishibashi [B6]. The velocity in Fukui-Ishibashi model takes an integer values between 0 and v_{max} and depends on the headway $\Delta x_j(t)$

$$x_j(t+1) = x_j(t) + \min[v_{max}, x_{j+1}(t) - x - j(t) - 1] \tag{6}$$

If $\Delta x_j(t)$ is larger than v_{max}, then the "vehicle" will move with maximal possible velocity. Otherwise, the "vehicle" updates its velocity to $\Delta x_{j+1}(t) - 1$.

2) Nagel-Schreckenberg model of Single-Lane Traffic: An elegant and simple model was proposed by Nagel and Schreckenberg in [B14]. In the most basic version of the model each site in a one-dimensional array of L sites can be empty or occupied by exactly one vehicle. Each vehicle has an integer velocity with values between 0 and v_{max}. The original paper set $v_{max} = 5$. With an arbitrary initial configuration an update rule for all vehicles is as follows:

1) **Acceleration:** if the velocity v is lower than v_{max} and if the distance to the next vehicle is larger then $v+1$, the speed is updated $v \leftarrow v+1$.
2) **Slowing down:** if a vehicle sees the next vehicle at distance $d \leq v$, then it updates speed $v \leftarrow d-1$.
3) **Randomization:** with probability p, the velocity of each vehicle is decreased by one: $v \leftarrow v-1$.
4) **Parallel update:** each vehicle is advanced v sites.

The original work of Nagel and Schreckenberg and the following works of Nagel and Herrmann [B12]s and Nagel and Paczuski [B13] have established that this random slowing along with asymmetry between acceleration and deceleration are crucial for realistic modelling of the freeway traffic. This model provides a mechanism for "traffic jam creation out of nowhere". Moreover, it reproduces a qualitative properties of the *fundamental diagram* (see Figure 2), which is the relation between the flux, density and speed of the vehicles on the road. The last rule, namely, the parallel update leads to a vehicle clustering rather than smooth density fluctuations. Another essential feature of the model is a wide gap between time scales of small perturbations in the system and the lifetime of the jams.

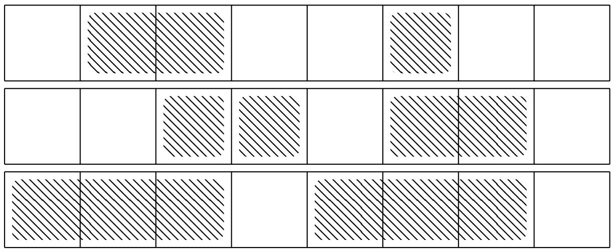

Fig. 1: A typical model of multi-lane freeway with heterogeneous vehicles (bus/car/truck/bike). In general v_{max} might differ according to a vehicle type.

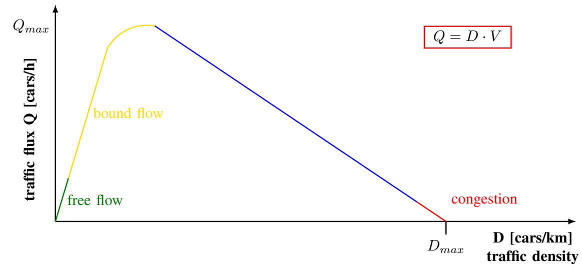

Fig. 2: Flow-density Fundamental Diagram of traffic flow. Triangular curve consists of two branches: on the left - free flow branch and on the right - congested flow branch. Q_{max} - the capacity of the roadway.

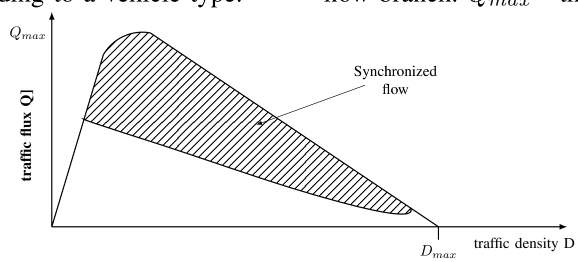

Fig. 3: The fundamental hypothesis of the three-phase traffic theory. Spatially homogeneous and time-independent states of synchronized flow in the flow-density plane are hatched.

3) Multi-Lane CA models: All the single-lane models are highly inadequate to describe the real world phenomena. Physical roads often come in variety of lane structure, road geometry and speed limits. Moreover, we are interested to handle different types of vehicles: small and fast private cars, long and slow trucks, bikes of any kind that can share the lane [B19], etc.

Usually, two- and more lane models update the state of the vehicles in two stages. At first, all vehicles are considered for the lane-change maneuver according to some rules. After the parallel implementation of the decision taken, all the vehicles considered in parallel for forward movement, as in the single-lane models. Lane changing rules are generally designed to mimic a human-like behavior. Therefore, those rules strive to answer two questions: a reason ("Do I want to change the lane?") and a safety ("Is it safe to change the lane?"). If both questions are answered positively, the vehicle changes the lane. While the safety concerns are similar and involve headway and speed of the vehicles around, the reason for lane-change vary deeply between the models.

A simplistic two-lane TASEP model was suggested by Nagatani [B11]. The maximum speed on both lanes was set to be identical, and, therefore, the model was lacking the ability to capture the heterogeneity of traffic and the pre-existing notion of "slow" and "fast" lanes used for safe-overtaking on the otherwise uncongested freeway.

A series of works expanding on the Nagel-Schreckenberg CA model was proposed. A common vehicle lane changing rules fire if the following are true: (a) the headway in the current lane is less than my current speed (b) the headway in the *other* lane is above my current speed (c) the nearest neighbor site on the *other* lane is empty. And the logic behind them is summoned from the observed human behavior on the road, i.e. change the lane if the vehicle in front of you is slow, there is a free slot near you on the other lane, your maneuver is beneficial and your actions do not disturb the traffic significantly. Some models assume the rules are symmetric regarding the change from left to right and vice versa [B3, B16]. Others embrace certain asymmetry between "fast" and "slow" lanes [B15, B20]. In some countries "fast" lane is reserved for overtaking, and the asymmetry comes to satisfy this law requirement.

4) KKW Three-phase CA traffic model: We define the term *steady state* to be a model solution, where vehicles move at the same distances one from another and with the same time-independent speed. Then all the above models were constructed such that in the limit they have a fundamental diagram of steady states, i.e. steady states form a curve in the flow-density plane.

Kerner proposed a different, no fundamental diagram, approach based on an empirical traffic flow analysis - a KKW model. A concept of "synchronized flow" and the related three-phase traffic theory were introduced in [B9, B10]. In the three-phase traffic theory features of spatial-temporal congested patterns are explained based on the phase transitions between the three traffic phases. Which, in addition to the "free flow" phase, are the "wide moving jam" and the "synchronized flow" phases.

Consider a moving traffic jam, where vehicles accelerate from a standstill on the downstream front of a jam If the downstream front propagates itself then such moving jam is called a "wide moving jam". In contrast, in "synchronized flow" traffic phase, the downstream front of the jam is fixed at a highway bottleneck. Usually, such bottlenecks are created on the on- and off-ramps, at intersections, near road geometrical features, crash-sites, etc.

The striking difference of the KKW model from the NS model is the acceleration/deceleration behavior of the vehicle. If the leading vehicle is outside the "synchronization distance", then it is safe to accelerate with some constant acceleration a. Otherwise, the behavior depends on the relative-to-the-leading vehicle speed. If the leading vehicle moves faster, then it is possible to accelerate until speeds are equalized, otherwise a deceleration should be applied to stay at the safe distance behind. Thus, there is neither a speed-dependent distance, nor the distance-dependent optimal speed. This feature could be summarized as the fundamental hypothesis of the three-phase traffic theory: steady states of synchronized flow cover a two-dimensional region in the flow-density plane. This means that in the synchronized flow state, where vehicles move at the same distance one from another and with the same time-independent speed, a given vehicle speed occurs at a different vehicle densities, and a given vehicle density could be attributed to a multitude different speeds (see Figure 3).

5) Urban traffic models: In the previous sections we have reviewed one-dimensional traffic models. Those strive to imitate the traffic on the freeway. However, city traffic flow is notoriously two-dimensional, thus rendering the previous ideas highly inappropriate. Moreover, the urban traffic, its patterns and associated issues are of great interest on their own. Especially, in the areas missing decent public transportation solutions and suffering from low traffic throughput and increasing pollution.

A notable approach to tackle traffic flow in two dimensions was proposed in [B1]. A number of simple cellular automaton models on the square grid were examined. Each cell on the grid can contain either an agent moving up, an agent moving right or the cell could be empty.

The model dynamics is controlled by traffic light, such that agents moving right move at even time steps, while agents moving up move at odd time steps. The movement is possible only if the destination neighbor cell is empty. The model is defined on the $N \times N$ square lattice with periodic boundary conditions, i.e. agents moving pass the border re-appear on the opposite side. Furthermore, the number of agents moving up and agents moving right is conserved in a column or row respectively.

Authors identified two qualitatively different asymptotic states which are separated by a sharp dynamical transition. Below the transition system self organizes into the separate rows of right- and up-moving agents along the diagonals. The arrangement allows agents to move at the maximal speed. While above the transition, agents are stuck and $\bar{v} = 0$.

In the second variant the cell could be occupied simultaneously by agents moving in different directions, though the movement is possible only if the neighbor cell in the movement direction was empty at the start of the time tick. In the last variant one of agents trying to move into the same cell will be prevented from moving. Agent choice is random with equal probabilities.

Sharp asymptotic state transition was identified in the third model, while in the second model case the transition is continuous. Therefore, an essential issue that leads to jam creation in the model is the need of up- and right- agent to cross the path of each other.

IV. Air Traffic Management

Intelligent air traffic management (ATM) systems are a prospective focus of research. Intelligent air vehicles will enable an infrastructure free environment which will result in lower costs of infrastructure development and maintenance. Moreover, adding a third dimension will remove bounds set by the extremely scarce resource, such as land in a city, that is allocated to roads. Air traffic management systems could be employed to reduce an air traffic congestion in busy airports. Furthermore, large scale delivery autonomous drone services are already operational in some urban areas. Therefore, systems and methods to manage growing navies of UAVs need to be developed.

In the near future, urban air mobility is expected to define the way people move from place to place and will substantially change the transportation systems as we know them today. We envision vehicles with vertical takeoff and landing used to transport people by air in urban areas. Such services will drastically reduce traffic congestion caused by the limits of physical road infrastructure and increase traffic flow significantly. However, these systems will need to ensure that safety separation requirements are met for air traffic at different altitudes, as flying vehicles will move along unstructured and dynamic routes, intersecting each other in an unpredictable way.

Safe management of considerable amount of flying vehicles in a shared airspace should be put to an intensive research. In the following section we will address several topics concerning ATM problems, methods of their analysis and algorithms that aim to solve them.

A. Airport Congestion Control

1) Traditional Airport Congestion Control: Traditional airport congestion problems are constrained by factors such as feasible landing time, time-based separation requirements, runway capacity and airline preferences Some of the frequently used methods to solve the aircraft arrival scheduling problem are position shifting [C10] and dynamic programming [C6]. Other methods include branch and bound, and branch and price techniques.

a) Rate Control for Taxiing Aircraft: Rate control of Taxiing Aircraft is one of the most important optimization problems in air transportation systems research. Ground taxiing accounts for non negligible part of delays at airports, flight time, fuel consumption and pollutant emissions. The congestion could be reduced by better towed aircraft management control polices. Control of queuing systems and dynamic programming are among the more prominent approaches that were investigated [C1]. Algorithms of queuing systems control suffers from practical limitations when applied to solve airport congestion problems. The need to interface with current air traffic control procedures and various sources of uncertainty, e.g. departure throughput variability and taxi-out times randomness, are the main limiting factors.

Pushback rate control is a policy to control aircraft departure from the airport gates based on a queue of departure times that allows towing an aircraft from its gate at a specific time. Usually this strategy is accompanied with a threshold heuristic approach that limits the number of simultaneously towed airplanes. In case the number of taxiing planes reach this threshold, all additional ships are kept at their gates until the number of taxiing aircraft decreases below the threshold. Since the desired pushback rate depends on the demand and the available resources at the airport it should be updated on a regular basis [C25].

A departure-based dynamic control approach that addresses the stochastic nature of an airport throughput is presented in [C24]. Dynamic programming is used to determine the optimal pushback rate in the departure queuing model. The solution is proved to minimize taxi-out times, while still maintaining runway utilization. The dynamic programming formulation considers a system state that depends on the number of towed aircraft and the length of the departure queue. The runway queue is modeled as a semi-Markov process based on runway throughput predictions, and the system state is determined by solving the resulting set of equations. The optimal policy is found by solving the Bellman equation for an average cost problem with infinite horizon.

b) Runway Configuration Selection: In [C1] the runway configuration selection problem is reviewed and two classes of models are described: prescriptive models and descriptive models. Prescriptive models aim to recommend an optimal runway configuration, subject to operational constraints. Prescriptive models include efforts to optimally schedule runway configurations, taking into account different models of weather forecasts and the loss of capacity during configuration switches. Descriptive models analyze historical data in order to predict the runway configuration selected by the decision makers.

c) Scheduling of Aircraft Landings: In [C3] a class of scalable dynamic programming algorithms are presented for scheduling of aircraft landings. This class of algorithms operates under the constrained position shifting requirement. In other words, an aircraft's position in the optimized sequence should not deviate significantly from its position in the first-come-first-served sequence. In [C2], the problem of finding the optimal landing sequence and landing times of aircraft subject to constrained position shifting requirements is investigated. A dynamic programming approach is presented and tested on realistic data in order to maximize runway throughput.

2) Airport Congestion Control for Urban Air Mobility: Urban Air Mobility (UAM) is a novel concept that is based on the use of electrical vertical takeoff and landing vehicles (eVTOLs) for aerial transportation of people in urban areas. UAM has the potential to completely transform the way people move around in cities and in suburban areas. UAM can help mitigate some of the most negative effects of urban transportation such as time consuming traffic congestion and air and noise pollution.

Urban Air Mobility is envisioned to be mainly an on demand service meaning that scheduling and allocation of airspace has to be managed in an online manner. At first UAM flights shall operate only in small numbers but in the long term the scale of such operations is expected to grow rapidly. Therefore UAM will need to be integrated with other vehicles sharing the same airspace such as commercial flights and delivery service drones. UAM flights will be carried out from designated stations that are called vertiports. The locations for these stations should be selected to have an efficient access by ground based transportation in order to achieve the goal of reducing traffic congestion.

Since the air space is mainly free and unpopulated, vertiports are destined to be the main transportation bottlenecks for an aerial transportation system. Furthermore, eVTOLs operate on batteries and therefore are limited by the battery lifetime, hence designed trajectories must ensure that UAMs have sufficient (and reserve) energy in order to safely complete the mission. Therefore computing the most energy efficient trajectory under given battery limitations is a fundamental challenge.

In particular, the arrival is the most safety-critical flight phase with greater air traffic density, less battery power compared to other stages of the flight and a possibly limited number of landing pads. Recently, an energy efficient trajectory optimization constrained by a given required time of arrival (RTA) has become an area of intensive research. Assigning RTAs for eVTOLs increases the predictability of air traffic and allows conflict avoidance. Designing 4 dimensional trajectories which UAMs follow both spatially and temporally while adhering to safety separation requirement is one of the approaches in the field.

The optimal RTA for eVTOLs is computed in [C14] using a mixed-integer linear programming. The problem formulation includes minimum time separation, eVTOL battery energy and vehicle dynamics constraints and is applied to a single landing pad with multiple arrival routes in multiple arrival fixes. A fixed final time multiphase optimal control problem is formulated in [C19] for designing energy efficient arrival trajectories for an eVTOL. The optimal control problem is solved using numerical methods. The assigned RTA can be met by using a speed adjustment strategy or a path modification strategy, or a combination of both.

B. Unmanned Airspace Traffic Management

1) Conflict Detection and Resolution Modeling Methods: For a detailed review on conflict detection and resolution modeling methods for air traffic see [C15]. In [C27], an automated conflict resolution algorithm for air traffic management using cooperative multi-agent negotiation is investigated. [C28] considers distributed cooperative on-board planning for conflict resolution of unmanned aerial vehicles.

2) Dynamic Air Space Allocation:

a) Air Highway Placement: Air Highway - an airspace structure was introduced in [C8]. It allows a tractable movement analysis of the UAV flying platoons. The paper puts forward the air highway placement problem, which could be summarized as follows: Consider a map $c : \mathbb{R}^2 \to \mathbb{R}$ which defines the cost $c(p)$ of a UAV that flies over the position $p = (p_x, p_y) \in \mathbb{R}^2$. Given a position p, the value of $c(p)$ corresponds to how costly a point p is in placing a highway that passes through it. High values of $c(p)$ indicate that point p is an undesirable point for a route to pass through while small values indicate that choosing this point is beneficial to reduce congestion. Locations with high and low cost can be determined a-priori by transportation authorities based on considerations such as interruption to commercial aircraft, noise and pollution, cost of accidents and risks above different areas. Alternatively, the cost of each location could be a dynamic function dependent on the developing congestion, weather conditions or other unexpected events. Let p^0 denote an origin point and p^d denote a destination point. Consider a sequence of highways $\mathbb{S}_N = \{\mathbb{H}_1, \mathbb{H}_2, \cdots, \mathbb{H}_N\}$ that satisfies,

$$\begin{aligned} &\mathbb{H}_1(0) = p^0 \\ &\mathbb{H}_i(1) = \mathbb{H}_{i+1}(0),\ i = 0, 1, \cdots, N-1 \\ &\mathbb{H}_N(1) = p^d \end{aligned} \tag{7}$$

(7) implies that the start point of the first highway is the origin, the endpoint of highway i is the starting point of highway $i + 1$ and the endpoint of the last highway is the destination. The set of highways form the network through which air traffic can travel. Given only the origin point and the destination point there are an infinite number of routes that can be defined by a sequence of highways that a UAV can take. However, when considering the costs of flying above different locations by a cost map the highway placement problem can be formulated as,

$$\begin{aligned} &\min_{\mathbb{S}_N, N} \left\{ \left(\sum_{i=1}^{N} \int_0^1 c(\mathbb{H}_i(s)) ds \right) + R(N) \right\} \\ &s.t.\ Equation\ 7 \end{aligned} \tag{8}$$

The regularization term $R(N)$ is used to limit N from becoming too large, resulting in a complicated highway network system with many short paths. A common choice for $R(N)$ is $R(N) = N^2$. (8) implies that air highways are line segments of constant altitude over which platoons of UAVs travel. A solution for the set of highways is obtained by solving a relaxation problem of (8) that results in an Eikonal equation that can be solved numerically using the fast marching method [C23].

The solution is then post-processed in order to obtain an approximation for (8). Cost-minimizing paths that are obtained from the solution of the Eikonal equation are composed of closely spaced set of points. Each of these paths is then sparsified in order to convert the dense paths into key waypoints that describe the geometric layout of each path.

b) Unmanned Aerial Vehicle Platooning: The air highways constructed in [C8] are characterized by wide routes shared between vehicles travelling to different destinations. This is an expected outcome since vehicles will travel in areas that minimize the cost function. Shared routes inspire the usage of UAV platoons moving in the same direction in order to avoid vehicle conflicts and increase traffic flow on these routes. Platoons allow meeting safety and destination requirements at large scales by considering pairwise interactions between platoons instead of individual entities in the air space thus lowering computation complexity. Grouping the vehicles into platoons, together with restriction to move only on designated air routes, leads to a restricted set of maneuvers available to UAV. The set depends on the individual vehicle's role in the platoon.

The authors propose a hybrid system that models UAV flying in a platoon. Every UAV can be in one of the three modes: free, leader or follower. Each mode has a specified set of the allowed maneuvers. If a vehicle malfunctions and is unable to operate within the set of allowed maneuvers, it transitions to a faulty state, descends through a fail-safe mechanism and leaves the air highway so that it does not pose a threat to vehicles moving on the highway. Likewise a functioning vehicle could transition to a faulty state, descend and leave the air highway to prevent collision with an other faulty vehicle.

The goal is to obtain control strategies that guarantee the success and safety of all the mode transitions. Platoon component enables each vehicle to perform complex actions, such as to merge onto a highway to form a platoon, join a new platoon, leave a platoon to create a new one, or react to malfunction or intruder vehicles. Additional basic controllers that perform simpler actions, such as to follow the highway at constant altitude at a specified speed or maintain a constant relative position and velocity with respect to the leader of a platoon are also proposed. In general, the control strategy of each vehicle has a safety component, which specifies a set of states that it must avoid, and a goal satisfaction component, which specifies a set of states that the vehicle aims to reach. Together, the safety and goal satisfaction controllers guarantee the safety and success of a vehicle in the airspace making any desired mode transition. These guarantees are satisfied using reachability analysis, thus allowing the multi-UAV system to perform joint maneuvers essential to maintaining structure in the airspace.

3) Path Planning for Unmanned Aircraft Systems: Path planning for unmanned aircraft systems shares many ideas and algorithms with path planning intended for ground vehicles, however trajectories have to planned more carefully since fuel and battery constraints are more tight for air transportation. The energy consumption of an unmanned aerial vehicle is proportional to its weight and therefore batteries should be adapted to the type of journey the vehicle undergoes. While traditional aircraft trajectories where often planned in advance, more agility in the planning stage will be needed when a growing numbers of aircraft systems both manned and unmanned, with differing communication and sensing capabilities and knowledge of the environment will share the sky. Since designated roads and lanes are not present in the air, unmanned aircraft systems should take into account making changes to their plans due to disruptions such as avoiding obstacles or adapting their path to avoid bad weather conditions. An additional constraint that needs to be taken into account when planning a path for an aircraft system is that aircraft have a minimum turning radius.

In [C29], Yu et al. describe several path planning approaches for unmanned aircraft trajectory conflict resolution. We follow the path planning classification as outlined in [C29]. Path planning and motion planning algorithms are compared using two metrics, completeness and computational efficiency. We say that an algorithm is complete if it terminates in finite time, guarantees that if a solution exists it will find it. Computational efficiency addresses run time issues and how the computation time of the solution increases as the state space increases. A classification of the reviewed UAV path planning approaches is presented in Fig. 4.

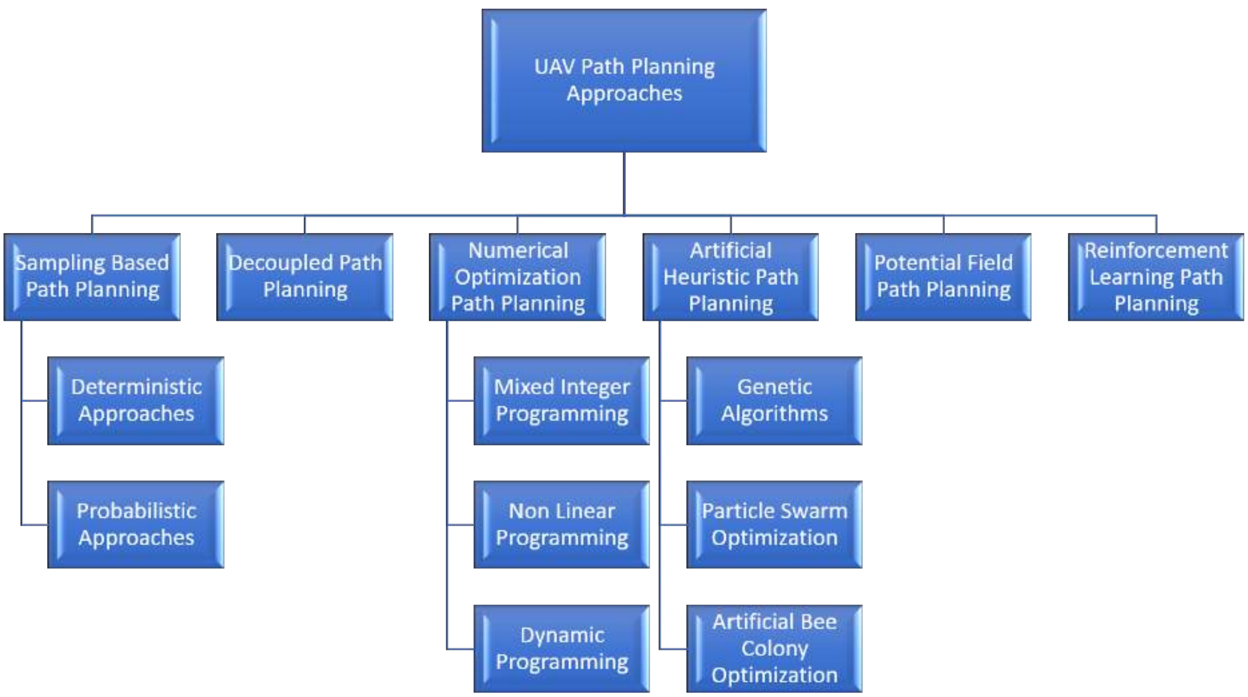

Fig. 4: Classification of UAV path planning approaches

a) Sampling Based Path Planning Approaches: Sampling based path planning methods sample the continuous state space in a finite set of points in order to reduce the continuous motion planning problem to a planning problem on a finite discrete graph called a roadmap. A path is then constructed between the source and the destination vertices using a graph search algorithm. The underlying graphs are mostly directed graphs. Traversable segments are represented by vertices and edges, where an edge's weight corresponds to the cost of travelling through it.

Sampling based path planning approaches are divided into deterministic and probabilistic. We will start with presenting some deterministic algorithms. One can consider an algorithm as resolution complete, if the returned solution provably converges to a global optimum as the number of samples grows.

Several deterministic sampling algorithms for routing the drones to a discrete set of locations are reviewed in [C18]. Drones visit some set of points, but start and end in the distribution center. The planning problems are modeled as generalized versions of Travelling Salesman Problem, the multiple Travelling Salesman Problem or the vehicle routing problem. The latter is a problem to find an optimal set of routes that a fleet of vehicles has to traverse in order to deliver its goods to a given set of customers. Generally, the routing problem is solved using a combinatorial optimization or integer programming methods.

Among the most popular probabilistic approaches are the Rapidly-exploring Random Tree algorithm by [C16] and the Probabilistic Road Map algorithm [C13] which both generate a search tree. When addressing probabilistic approaches one should change the definition of completeness to probabilistic completeness. Probabilistic completeness means that if a sufficient time to check an immense number of samples is given then the probability that a solution will be found if it exists converges to one.

b) Decoupled Path Planning Approaches: This type of algorithms first search for a discrete path through the configuration space using an algorithm such as A^* [C11] or the probabilistic roadmap. The best trajectory is constructed under given constraints that are encoded in the topology and edge weights of the graph that represents the configuration space. At the second stage the obtained discrete path is used as the basis to produce a trajectory that is both feasible for the dynamics of the vehicle and steers it away from obstacles. Completeness of an algorithm implies that it terminates with a solution if one exists. While decoupled path planning approaches are computationally efficient it is difficult to theoretically prove completeness and optimality. When addressing A^* that includes a heuristic function we need to describe the concepts of admissibility and consistency of heuristic functions. Admissibility means that the heuristic function never over estimates the real cost to get to the target vertex.

Let $d(x, y)$ denote the length of the edge between vertices x and y. If the heuristic h satisfies $h(x) \leq d(x, y) + h(y)$ for every edge (x, y) of the graph then h is called consistent. With a consistent heuristic, A^* is guaranteed to find an optimal path without processing any vertex more than once. The basic steps for an A^* algorithm are as follows. Consider a graph with multiple vertices. The goal is to reach the target vertex from the starting vertex as quickly as possible. What A^* search algorithm does is that at each step it picks the vertex according to a value f which is equal to the sum of g and h. At each step it picks the vertex having the lowest f, and selects it to be the next vertex in the path towards the target. We have that, $f(n) = g(n) + h(n)$, where n is the previous vertex on the path, $g(n)$ is the cost of the path from the start vertex to n and $h(n)$ is a heuristic that estimates the cost of the cheapest path from n to the target vertex. A^* progresses by building a tree of paths that originate from the starting vertex and expands these paths one edge at a time until the termination criterion is met. Several papers that use this approach for UAV path planning are listed in [C17].

c) Numerical Optimization Approaches: Path planning is often performed using numerical optimization algorithms that aim to find an optimal trajectory by formulating an objective function subject to constraints and maximizing or minimizing it using a mathematical programming methods. The programming methods that are mainly used for trajectory optimization are Mixed Integer Programming, Nonlinear Programming, Quadratic Programming and Dynamic Programming.

A mixed integer programming problem is an optimization problem in which some of the decision variables are restricted to be integers while some are continuous. When some constraints or an objective function are nonlinear we get nonlinear programming problem. Another interesting class are problems that aim to optimize a quadratic objective function of several variables based on linear constraints on these variables. These are quadratic programming problems.

The last, but not least programming approach is dynamic programming. It is both an optimization method and a computer programming method. Essentially, dynamic programming aims to solve a complicated problem by decomposing it into simpler sub problems, then solve the full problem in a recursive manner. Dynamic programming is built on the concept that if a problem can be decomposed into sub problems that can be solved optimally than the whole problem can be solved optimally using dynamic programming approach.

When applying dynamic programming to an optimization problem the decision process corresponds to taking a series of decisions over a time horizon that can be finite or infinite. This is done by defining a sequence of value functions $V_1, V_2, \cdots, V_n$ taking y as an argument that represents the state of the system at time i. $V_n(y)$ is the value obtained in state y at the last time n. The values V_i at earlier times $i = n-1, n-2, \cdots, 2, 1$ are obtained by going backwards using a recursive relation called the Bellman equation. For $i = 2, \cdots, n$ V_{i-1} at any state y is calculated from V_i by maximizing a function of the gain from the decision at time $i-1$ and the function V_i at the new state of the system if this decision is made. Since V_i has already been calculated for the needed states, this operation yields V_{i-1} for those states.

Using this backwards procedure, V_1 - the value at the initial state of the system is the value of the optimal solution and the values of the decision variables are obtained by tracking the performed calculations. The solution of the sub problems are stored in a memory structure and are used in future calculations that build upon the solutions of the subproblems. The popular Dijkstra, Floyd-Warshall, and Bellman-Ford Algorithms are all examples of dynamic programming algorithms and can be applied in the context of UAV path planning.

[C20] provides an overview of path-planning and obstacle avoidance algorithms for UAVs and compares various approaches to the problem. We shall now discuss several papers that use numerical optimization in the context of path planning for UAVs based on the applied optimization method.

Integer programming is a main tool that is used to solve path planning and scheduling problems that could be modeled in a graph structure. Integer variables can represent binary decisions that correspond to choosing paths between vertices of a graph.

A mixed integer linear programming (MILP) is a restricted approach that only uses linear and binary variables. Difficulties associated with nonlinear programming, such as choosing a suitable initial guess for the optimization, are eliminated altogether if an optimization problem is formulated as MILP. Furthermore, mixed integer linear programming incorporates logical constraints that restrict the airspace into allowable regions that do not contain obstacles. Along with continuous constraints it allows path planning for fixed-wing or quad-rotor dynamics aircraft.

Trajectory optimization for autonomous fixed-wing aerial vehicles based on a receding horizon control approach was presented in [C4]. Receding horizon control, also known as model predictive control, designs a trajectory that is optimized over a period of time, called the planning horizon. The recommended trajectory is followed over a shorter execution horizon and the optimization is performed again starting from the state that is reached. The re-planning step uses feedback to account for disturbances and modelling errors in order to correct the trajectory. The controller produces almost time-minimal planar trajectories under diverse constraints that include no-fly zones, the vehicle's maximum speed and its turning radius. In order to reduce a considerable non-linear computational cost short paths over the receding horizon are planned. MILP is then applied to guarantee that the vehicle reaches its goal. The cost to the goal is estimated by constructing a visibility graph representation of the environment and searching this graph for the minimal time to reach the goal using a modified version of Dijkstra's algorithm.

The problem of planning a feasible trajectory for a quad-rotor UAV while avoiding obstacles is investigated in [C9]. A small number of obstacle-free convex regions are created to cover the space. The trajectories are constructed using mixed-integer optimization. A substantial reduction in computation time is achieved due to assigning variables to the above regions instead of abundant obstacle faces in cluttered environment.

The problem of optimal cooperative three dimensional conflict resolution that involves multiple UAVs is posed in [C21]. The authors tackle the problem as a continuous time optimal control problem of finding optimal trajectories while maintaining safe separation between every UAV pair. UAVs operate in a free flight mode and the aim to get them from a known starting location to a goal without colliding with each other or with obstacles. Protection zones associated with each UAV pair and each UAV-obstacle pair are modeled using disjunctions which are represented using continuous variables. The resulting model is a non-convex finite dimensional nonlinear program which is solved with an Interior Point algorithm that incorporates a line search method and an initialization strategy that enables fast and robust convergence of the optimal control problem.

A relaxed version with an UAV simple linear kinematics is presented and solved with MILP in [C5]. The MILP and non linear programming solutions are compared and an approach that uses the MILP solution as an initialization for non linear program solvers that allow more complex dynamic models of UAV movement is analyzed.

B-spline based path planning algorithm was proposed in [C12]. The algorithm takes as input a discrete set of way-points, incorporates dynamic constraints of the vehicle and outputs a path that passes through all the way-points. B-spline curves are used to compute smooth and dynamically feasible trajectories for UAVs in a polygonal channel bounded between piece-wise polylines. Mathematically, B-splines can be represented by a small number of parameters, removing the need for a full geometric description of the path, therefore allowing to compute the trajectories efficiently and in an on-line manner. The parameters for the B-splines that form the trajectories are found by solving a constrained optimization problem using a sequential quadratic programming solver.

d) Artificial Heuristic Approaches: Computing exact solutions for path planning problems in complex dynamic environments is often time consuming and requires extensive computational resources in order to generate a feasible trajectory in real time. Some times an optimal solution may not be required, and a heuristic-based quick approximate solution may be suited for dynamic environments. Choosing a heuristic essentially trades optimality, completeness and theoretical performance guarantees for a fast and convenient solution. Below we review several notable artificial approaches that are used to solve the UAV path planning problem such as Genetic Algorithms, Particle Swarm Optimization and Artificial Bee Colony Optimization.

A Genetic Algorithm is a heuristic search method inspired by the concepts of evolution and the survival of the fittest in nature. It provides computationally efficient results by relying on biologically inspired operators such as mutation, crossover and selection. Path planning genetic algorithm iteratively optimizes a candidate path based on some fitness criteria until a solution satisfying termination condition is found.

Particle Swarm Optimization is a population based algorithm. The population is called a swarm, and candidate solutions are called particles. The particles move around the search space changing their positions and velocities. The optimization is performed iteratively, and better solutions are found by incorporating both local information known to a single particle and collective knowledge of the swarm that guides movements of other particles. The algorithm performs the global search followed by a local search under the constraints of UAV dynamics, environment and real time requirements. The path planning problem is formulated as a hybrid optimization problem.

In [C22] a comparison of parallel genetic algorithm and particle swarm optimization are performed for real time UAV planning. This algorithm performs in every iteration both local and global searches, therefore increasing the chance to obtain optimal solutions.

Another notable technique is an Artificial Bee Colony Optimization. This is an optimization algorithm motivated by the intelligent collective foraging behavior of a honey bee swarm.

e) Potential Field Approach: Potential Field algorithms use an artificial potential function that assigns a weight to every position in the environment based on the location. This type of algorithms rely on the notion of attractive and repulsive forces by simulating the reaction of a vehicle to a potential field that pushes it towards the minimum potential. Destination point is assigned a minimum potential in the environment, and, as a consequence, the vehicle is attracted towards it. Usually, the potential function assigns repulsive forces to obstacles, thus preventing collisions between obstacles and UAVs.

A challenge for this type of algorithms is to avoid getting stuck in a local minima. This could happen, for example in cases where an obstacle is located between the UAV and the goal. On the other hand, Potential Field based algorithms generate trajectories without significant computational effort and therefore can be implemented in real time.

Decentralized control law that enables a group of vehicles to accomplish a control objective without colliding with one another and with unexpected obstacles is presented in [C7]. The control law relies on constructing a potential function that uses gyroscopic forces and does not require global information of the environment in order to achieve the task. An optimized artificial potential field (APF) algorithm for multi-UAV operation in a three dimensional dynamic space was proposed in [C26]. The authors presented a distance factor and a jump strategy that guarantees collision-less trajectories as well as addressing the problem of getting to unreachable targets. The method considers the UAV companions as dynamic obstacles to realize collaborative trajectory planning.

f) Reinforcement Learning: Reinforcement Learning (RL) is a machine learning approach in which a set of agents learn to take actions by interacting with a dynamic environment with a goal of maximizing a cumulative reward. The feedback to the actions is obtained through the reward signal, and the exploitation concept is applied to choose the next action. An agent may also choose to explore the environment to discover additional or better action candidates. This is known as the exploration concept. Many RL approaches use a dynamic programming technique. Therefore one of the main objectives is to estimate the degree of importance that an agent assigns to a given state.

As a rule, a reinforcement learning model consists of 5 elements: S, the set of states; A the set of actions, policy or mechanism to transit between states; $P : A \times S \to [0, 1]$, usually given by the probability of taking an action a while at state s; a scalar reward of a transition, and a method for observing the agents.

Q-Learning is a reinforcement learning method applied to solve path planning problems, but that does not require a model for an environment. Moreover, it is able to handle problems with stochastic transitions and rewards. The basic idea of Q-Learning is to find the optimal control policy by maximizing the expected total reward over all future steps. If a UAV action a yields a real return r, then the objective is to obtain a strategy $Q : S \to A$ that maximizes these returns. Denote by $\gamma \in [0, 1]$ a discount factor that weighs earlier rewards heavier then those obtained later, and by δ a map between the current state-action pair and the next state. The training is carried out based on the immediate return and the long term return of the action given by,

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(\delta(s, a), a') \tag{9}$$

The UAV repeatedly observes the current state s, selects and executes a certain action a, observes the returned result $r = r(s, a)$ and the new state $s' = \delta(s, a)$. Any action a can be found then by solving,

$$a = \arg\max_{a'} [r(s, a) + Q(\delta(s, a), a')] \tag{10}$$

Geometric Reinforcement Learning (GRL) is introduced in [C30]. GRL exploits a specific reward matrix for a simple and efficient path planning of multiple UAVs. Candidate points are selected along the geometric path from the current point to the target. The reward matrix is adaptively updated based on the geometry and the risk information shared by other UAVs. The convergence of the calculations is theoretically proven, and the path length and risk measure are determined. Experiments validate the effectiveness and feasibility of GRL for the UAV navigation.

V. Sensing and Perception

Intelligent Transportation Systems rely on the information gathering from various types of sensors. Fused and distilled information allows to make smart decisions across the different layers of the ITS architecture. In the ITS ecosystem sensing is carried out by a broad range of sensors that can be vehicle-, infrastructure- or user- based.

Vehicle based perception is a fundamental source of information that enables autonomous vehicle maneuvers. Such sensors include video, radar, lidar, GPS and inertial sensor types. A vast body of works focuses on advancing the state of these sensors and on developing perception algorithms based on sensor outputs. Notable examples of such algorithms are object detection, tracking, collision avoidance, mapping, localization and SLAM.

Infrastructure based perception consists of sensors that collect data from the ITS environment. Typically such sensors are embedded in roads and in road infrastructure, e.g. on traffic lights, dedicated installations etc, and monitor traffic state or augment vehicle sensing capabilities. Sensors of this kind range from cameras to inductive loop detectors, street parking sensors to speed detectors, laser or infrared sensors to air quality sensing devices.

User based perception is the gathering of information from smart devices carried by users of the ITS. A source of information could be a mobile device which collects GPS data, a travel history accumulated by transportation related applications or social media. Furthermore, a valuable information could be obtained from the smart ticket usage in the public transportation systems. An information that comes from both pedestrians and drivers could be used in traffic monitoring and route suggestion. Insights gained by analyzing travelling information let [D12] outline a number of goals that would be addressed by improved sensing and perception algorithms.

A. Cooperative Sensing

In the connected settings, vehicles will be able to share and obtain a vital information from peers or an infrastructure. Better decisions could be made as autonomous vehicle cooperation is expected to improve individual local sensing capabilities. Moreover, necessary safety guarantees could be met as an outcome of more accurate perception and understanding of the vehicle surroundings.

1) Cooperative Localization: In order to generate a more accurate representation of a vehicle's surroundings, a sensory information from vehicles in the vicinity and an information from infrastructure sensors should be combined. Since measurements always contain some degree of noise, algorithms that fuse measurements from different sources mitigate this effect and reduce perception uncertainty. Likewise, access to the information, that is not available to a certain vehicle, but could be acquired from its neighbors, improves an individual vehicle performance.

A map merging approach that aligns multiple local sensing maps to make observations consistent with each other is proposed in [D6]. Frequent transmission of a local sensing map is computationally inefficient and bandwidth consuming. Hence, authors choose to explore a more efficient approach : automatic alignment is gained once a vehicle is localized on the global map. Cooperative localization is also possible through the exploitation of correlations in joint and relative observations, such as relative range and relative bearing.

a) Minimal Sensor Configuration: The problem of multi-vehicle cooperative localization where all vehicles are equipped with a complete sensor suit has been thoroughly investigated. However, if vehicles were to operate in a connected environment under shared sensory information paradigm, the individual sensing capabilities of each vehicle can be reduced. Quality and quantity sensor requirements that allow vehicles cooperatively localize themselves up to an acceptable error is an interesting open question that is addressed in [D17]. The work tackles the cooperative localization of a distributed set of robots. Authors design a minimal and scalable sensor configuration which allows cooperative localization of a vehicle fleet on urban road.

2) Motion Coordination: Sharing the information on the planned vehicle trajectory should help autonomous vehicles predict dynamic changes in the environment more accurately. If trajectories overlap or violate safety requirements conflict detection and resolution algorithms must be applied. One type of distributed conflict resolution mechanisms is introduced in [D10]. The conflict resolution algorithm for each vehicle is decoupled temporally and allows connected vehicles to navigate safely and efficiently through intersections without a traffic manager. A vehicle computes the desired time slots to pass the conflict zone by solving a conflict graph locally based on the information that was broadcasted from other vehicles. In the motion planner part, a vehicle computes the desired speed profile by solving a constrained optimization problem. The paper provides theoretical guarantees, that the combination of local vehicle decisions solves the conflicts globally.

B. Infrastructure Free Traffic Monitoring

Floating Car Data(FCD) is a traffic data that comes from vehicles and includes speed measurements and travel times. FCD in combination with infrastructure based traffic monitoring devices are typically used to monitor traffic, identify congestion and provide estimated travel time information. However, the cost of deploying and maintaining traffic sensing devices, such as loop detectors and video cameras, in a dense enough manner is very high. Therefore, cooperative sensing devices that transmit the information they gather from the environment can improve traffic monitoring capabilities at low cost. Moreover, the data from these sensors could be utilized for traffic monitoring at no additional cost. It is expected that the ability to build a more accurate and reliable state of traffic improves with an increased penetration of cars that possess cooperative sensing capabilities.

An adaptive smoothing interpolation method for estimating spatio-temporal properties of highway traffic such as flow, speed and density is presented in [D19]. The approach is based on the fusion of an input data obtained from stationary infrastructure sensors with a floating car data. The method successfully detects transitions between free and congested traffic and identifies traffic structures such as stop-and go waves.

An online freeway traffic estimation and monitoring algorithm based on real floating car data is proposed in [D13]. The idea is based on the generalized adaptive smoothing interpolation method and aims to determine the accuracy of traffic speed prediction using only delayed sparse floating car data. The method is evaluated with respect to varying data densities and delays.

A spatio-temporal freeway traffic diagrams based on a mapping-to-cells method are constructed in [D5]. At first, the traffic network is partitioned into small square cells. Then, a traffic flow direction of each cell is determined. At last, a traffic diagram that depends on speed is constructed. A similar approach that models traffic flow using the Lighthill-Whitham-Richards PDE is analyzed in [D8, D15]. The model is based on three fundamental variables: *flow*, defined as the rate at which vehicles pass thorough a point; *density*, defined as the spatial concentration of vehicles and *speed* which is the average rate of travel. These traffic models are usually coupled with data assimilation techniques such as Kalman filtering in order to combine model and observations. Unfortunately, some approaches require additional data inputs that are not available through floating car data such as *flow* or *density information* that can be obtained from additional sensors such as loop detectors.

A strikingly different approach to estimate a velocity field based on the GPS data only, provided by moving vehicles is investigated in [D20]. The authors use vehicle GPS data readings only as an input to a Lighthill-Whitham-Richards based PDE. A Godunov discretization scheme then casts the PDE into a Velocity Cell Transmission Model. Finally, the Ensemble Kalman Filtering technique is applied to this nonlinear dynamical system in order to estimate the velocity field.

C. Parking Monitoring and Parking Management Algorithms

Finding an available parking spot, especially in the dense urban areas, can sometimes be a time-consuming process that increases travel time and creates unnecessary congestion and emissions. Therefore, a fast and accurate detection and monitoring of vacant parking places accompanied by the efficient information dissemination is of great importance. One viable approach to monitor available parking spots is embedding sensors in the infrastructure [D14]. Among the different types of sensors that could be used are magnetometers [D1], video cameras [D16], optical sensors [D3], ultrasonic sensors [D7] and many more. Some works suggest to use vehicle based detectors to cooperatively update a map of available parking spots and distribute this information through vehicular ad-hoc networks. However, most currently deployed solutions are infrastructure based monitoring solutions.

A cooperative reservation protocol in a vehicular ad-hoc network (VANET) is developed in [D4]. The protocol is designed to efficiently allocate parking spaces and avoid competition created by its simultaneous broadcasting nature. An Encounter Probability (EP) function is employed to assess the relevance of each event in the network. In the proposed protocol a vehicle, often the one that leaves its parking spot, is chosen to coordinate the parking of other cars by sending a message over the VANET. The role of the coordinator is to collect information from nearby vehicles seeking to park and assign the parking spot in a way that will be both fair and efficient.

A predicting occupancy model is introduced in [D2]. The model estimates the future parking situation at the time of arrival and is based on Queuing Theory and continuous-time homogeneous Markov models. Drivers can decide if they wish to park in this particular parking lot, based on the predicted parking lot occupancy, or head elsewhere. The model's performance is evaluated in a simulation of a city in Germany.

Since VANETs are limited in bandwidth and the transmission range, the importance of only the relevant information transmission is paramount. A machine learning approach to determine what is the *relevant information* in the context of parking lot assignment is investigated in [D18]. The learned relevance of the parking location is then used to assist vehicle decision-making process by estimating the probability that the parking spot will still be available at the time of arrival. Much lower parking discovery times were achieved in simulations, as the method was evaluated against a blind search approach.

Smart parking solutions could be classified into three macro-themes: system deployment, information collection, and information dissemination. Parking information could be collected in centralized or decentralized manner by various sensors. Unfortunately, a concurrent access to the available information leads to a competition between drivers over the attractive free parking spots. Therefore, it is worthwhile to explore broadcast algorithms that simultaneously minimize the number of empty parking spots and do it in a fair manner. Moreover, a driver's parking behavior should also be factored into the solution.

A parking competition behavior is addressed in [D9]. Various strategies to coupe with the arising competition problem are reviewed. Between them are ant colony optimization, cellular-automata methods, clustering algorithms, dynamic pricing, expert systems approaches, fuzzy logic, genetic algorithms, gravity based parking algorithms, game theoretic approaches, algorithms for solving Hitchcock transportation problems, multi-agent system based algorithms, Markov chain based algorithms, and maximal interval scheduling algorithms.

D. Infrastructure Based Perception

1) Infrastructure Based Traffic Monitoring: Infrastructure based traffic monitoring systems collect information about traffic parameters: vehicle speed, traffic density and other traffic flow characteristics. Monitoring systems could be installed as an intrusive solution such as inductive loop detectors or pneumatic road tubes embedded in the road infrastructure. Non-intrusive sensors, like video cameras and microwave radars, are also widely used. On the other hand, infrared instruments, ultrasonic and passive acoustic arrays are applied to a much lesser extent. After the data was collected, it is passed to the transportation authority central for processing. Furthermore, transportation authorities are able to build a map of the current traffic condition, identify congestion and inform drivers on developing road conditions.

The deployment of infrastructure-based monitoring equipment is in general costly and requires periodical maintenance. Therefore, sensors usually are not deployed in a dense enough manner for the desired level of accuracy. Moreover, different sensors bear different operation disadvantages, e.g. visible light cameras function poorly in bad weather conditions such as fog or snow, while inductive loops should be deployed in large numbers to cover a certain area and replacing them causes a lane closure for maintenance. Luckily, fusion of different sensor outputs, capitalizing on the individual advantages, while also overcoming the shortcomings, is a promising approach to build a reliable and accurate traffic monitoring systems.

A noticeable example of an infrastructure-based monitoring systems is a wireless sensor network. This is an active area of research exploiting infrastructure based sensors, most often in combination with other sensors, for traffic monitoring. Traffic data is collected by a network of sensor nodes, then passed to a gateway node, which is responsible to transmit the collected information to a base station. See [D11] for a survey on the urban traffic management systems based on wireless sensor networks and for an overview on the traffic sensing technologies.

2) Traffic Light Scheduling: Traffic lights is currently the predominant method to control and regulate the traffic flow at intersections. Traffic lights provide safety for vehicles and pedestrians crossing the intersection by temporarily halting the traffic flow on one or more directions. However, if the scheduling is not configured optimally, traffic lights lead to major delays and decrease to the throughput of the road network. Therefore, optimal dynamic scheduling algorithms for traffic lights are essential for reducing congestion in transportation networks.

VI. Traffic Analysis and Optimization

Simulation of real world scenarios plays an important role in intelligent transportation research. While various transportation systems benefit from accurate simulations, it is that simulations are an integral part of transitioning to smart and intelligent transportation.

A. Data Generation for Machine Learning Applications

Machine learning (ML) methods are recently on the fast track to become a default solution to myriad of problems. However, a shear amount of data required to train an ML system is insurmountable. The problem becomes even more acute in the transportation research areas. Millions of miles, thousands of scenarios, diverse settings etc. etc. need to be "encountered" and learned before an ML system could be allowed to handle the real world settings. Hence, simulating various and numerous scenarios is of the uttermost importance. Besides, already existing data sets could be augmented by photo realistic simulated images of precipitation (rain, fog, snow). Such images could be used to verify the robustness of image based perception and navigation algorithms of an autonomous vehicle.

B. Vehicle Safety and Driver Assistance Systems Simulations

Simulated data is expected to help testing vehicle safety and integration of driver assistance systems prior to going production in the field. Such simulations would be important for testing autonomous maneuvers. Lead vehicle malfunction in a connected vehicles platoon scenario, or sudden changes in the environment, e.g. a pedestrian jumps to the road, are readily examples. Both cases require quick vehicle response to reduce the probability of accidents in the real world context.

C. Short Term Traffic Prediction Algorithms

As mentioned before, fused and aggregated data from multiple information sources, both real-time and historical, is an invaluable asset. Traffic management systems can utilize this data to provide services to the users of the transportation systems. One such service is future traffic flow prediction using the collected data as an input to a traffic prediction algorithm. Traffic prediction algorithms enable an early bottleneck detection and allow better routing decisions thus reducing congestion and travel time. Furthermore, precise estimation of a travel time allows transportation system users to plan their journeys more accurately.

According to [E12] there are three properties that define a traffic prediction system. The first is the scope of the prediction system. The scope determines if a prediction system will be used in a traffic management system or in a traveler information system, as well as a target road type. The second is the data resolution property, i.e. the forecasting horizon and the step. The horizon is the prediction time ahead and the step is the time interval of the prediction update. The forecasting accuracy is directly influenced by the values of the horizon and step , therefore selecting an appropriate values for both of them is a necessary condition to obtain an accurate prediction. Accuracy degrades as the horizon and step values increase, however decreasing the step sharply increases the computation cost. The last factor that affects the forecasting accuracy is the underlying traffic model.

We briefly discuss several prediction approaches. At the beginning, in the late 1970's most works modeled the flow of traffic using statistical and time series tools, especially the Autoregressive integrated Moving Average Model (ARIMA) [E1, E3]. Spatial and temporal interactions could then be introduced to capture the nature of the road traffic. The multivariate autoregressive model based on spatio-temporal correlation matrices [E8] realized transient traffic effects in the traffic network while performing lighter computations than ARIMA forecasting algorithms. Other works employed time series analysis [E4] based on a real-time data collected from road side terminals combined with a historic data using a Kalman filter to estimate travel times.

Multivariate time-series state space model that incorporates real time data from loop detectors [E11] was applied to estimate traffic volume, travel speeds and road occupancy in urban areas.Later, GPS data was fused with data obtained from the fixed sensors on the road. Though, GPS data is characterized by the spatial and temporal sparsity, it does not require costly infrastructure installations to monitor the traffic. Through the combination of two data sources real-time traffic predictions were made [E10].

When applying a new traffic model two competing requirements should be met. Completeness of the model, i.e. how accurate the model predicts the traffic. Usually, completeness is a function of the number and type of the model parameters. On the other hand, we require the model to scale nicely with the number of modeled vehicles. These two goals usually have opposite effects on predictions. The first goal results in a specification of a greater number of estimation parameters, while the second goal results in a decrease of that number. The models strive to find a well-balanced compromise between the conflicting goals using the structure of the transportation network.

Research based on data driven empirical algorithms, has been blooming alongside classical mathematical models based on macroscopic and microscopic theories of traffic flow. Neural networks, pattern recognition and machine learning approaches were successfully applied in the area. Artificial intelligence (AI) approaches neglect classical theory and focus on the empirical testing of the model on the available traffic data.

Long short-term memory (LSTM) network to predict short-term traffic is proposed in [E15]. The suggested two-dimensional LSTM network composed of many memory units was used to investigate tempo-spatial correlations in traffic system. Due to the availability of data and with the expressive power of neural networks, the proposed LSTM network achieved superior performance in comparison with the previous classical approaches.

The temporal graph convolutional network model is used to predict traffic in an urban setting [E14]. The proposed network consists of a combination between the graph convolutional network and a gated recurrent unit. The graph convolutional network is used in the model to learn complex topological structures that express the spatial dependencies while the gated recurrent unit's role is to learn the temporal connections. The model predicts short term traffic based on the learned spatio-temporal correlations and shows promising results on real world data.

Short-term traffic forecasting trends are discussed in [E13]. Amongst the outlined research directions are the development of responsive algorithms and prediction schemes, freeway, arterial and network traffic predictions, model selection and testing, and model comparison. A recent review on the state of field is provided in [E7].

D. Microscopic Traffic Simulation

Large-scale real world communication protocol tests are extremely expensive, considering high cost of communication equipment that should be installed on vehicles and in the infrastructure. Therefore, simulations remain one of the most important tools for evaluating ITS applications and the performance of VANET communication protocols. Moreover, microscopic simulations are essential in solving intelligent logistics problems, e.g. such as when choosing the locations of battery recharging stations for electric vehicles in a city. Testing the public transportation management systems in bus route selection, mass transit scheduling, railway and metro congestion scenarios are other notable simulation applications.

1) Microscopic Traffic Simulators: Several popular road traffic microscopic simulation tools and their applications are discussed in [E6]. Each entity (car, train, person) is simulated individually, including interactions between the entities.

The prominent microscopic simulators are listed below:

a) SUMO: Simulation of Urban Mobility: SUMO [E2], is an open-source traffic simulation tool that allows modelling of inter-modal traffic systems including road vehicles, public transport and pedestrians. SUMO can be used to simulate traffic conditions in large road networks, assist in traffic prediction, performance evaluation of traffic lights algorithms and evaluation of route selection algorithms. The tool can provide traffic forecasts and test communication protocols that include movement of vehicles on a real road network. It can simulate the influence of the autonomous vehicles and platoons on the traffic flow, help to determine traffic safety and provide risk analysis. Furthermore, it can calculate noise and pollutant emissions of vehicles and aid in developing ecologically friendly routing algorithms. The movement of vehicles in SUMO can be modeled in both continuous and time discrete manner. Traffic rule configuration is a notable advantage of the tool. Furthermore, SUMO supports simultaneous movement of up to $100K$ vehicles on tens of thousands of streets.

b) iTETRIS: Integrated Wireless and Traffic Simulation Platform for Real- Time Road Traffic Management Solutions: iTETRIS [E9], is an open-source simulation platform integrating two widely adopted traffic and wireless communication simulators SUMO and NS3. iTETRIS supports the implementation of cooperative ITS applications that rely on a dynamic exchange of information between vehicles and between vehicles and infrastructure. iTETRIS allows to model the interaction between a large number of different cooperative entities in the ITS prior to real world tests with high accuracy.

c) STRAW: Street Random Waypoint: STRAW [E5] is an open-source traffic simulator to evaluate wireless ad-hoc network protocols by using a realistic vehicular mobility model on actual US cities road maps. It is implemented using the JIST/SWANS discrete event simulator. STRAW improves the accuracy of the random waypoint model that simulates the movement of entities in the network by randomly changing their location, velocity and acceleration over time. In the random waypoint model, each entity selects a random destination and a random speed up to a maximum speed, moves freely and without restrictions to the destination point and pauses for a fixed amount of time before repeating the process. By constraining the movements of entities to actual streets and by limiting the entities' mobility according to traffic conditions, STRAW provides the insights that are directly comparable to the real world experiment results.

d) VISSIM: PTV Vissim is a microscopic interaction based traffic simulator. VISSIM allows simulation of heterogeneous entities that interact with each other, e.g. cars and trucks, bicycles and motorcycles and pedestrians, etc. It uses a discrete event simulator to model road traffic and allows detailed 2 and 3 dimensional visualizations. VISSIM is mainly used to study transportation planning and vehicles-to-pedestrian interactions in urban environment.

E. Macroscopic Traffic Simulations

A different kind of models based on the "average" properties of the traffic, like flow and density, are called macroscopic models. Therefore, simulations of models with a focus shifted from an individual entity to the averaged behavior of many entities are known as macroscopic simulations. The macroscopic characteristics of traffic flow can be simulated using significant amounts of trajectory data. Recently such simulations were adopted to study traffic flow at intersections, to identify inter-vehicle conflict situations, to study operation of smart traffic lights, and to identify possible congestion bottlenecks in planned infrastructure. Traffic flow simulations are applicable beyond the ground traffic and can also be applied to study air traffic management applications, e.g. flow around airports or delivery drone services scheduling.

F. V2X Communication Coverage and Throughput Simulations

Sensing horizon of an individual vehicle is limited by the range of the vehicle own sensors. Vehicle-to-everything (V2X) communication protocols could expand that horizon well beyond that limit. Simulation of communication protocols assist in testing network throughput in vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) networks, help solve coverage issues and improve an information flow. Finally, they could provide invaluable hints When placement of range limited and costly smart infrastructure is considered. Therefore, simulations of communication protocols are an essential part of developing complete traffic automation.

VII. Smart Public Transportation

Smart public transportation solutions are expected to alleviate road congestion and lower the emissions by reducing the number of private vehicles on the road, mainly in urban or suburban areas. Ultimately, advanced, reliable and user friendly urban public transportation systems such as public buses with dynamical routes and timetables, smart rail management systems and demand responsive transportation can improve passengers' experience.

A. Adaptive Route Planning for Public Transportation

1) Railway Management Algorithms: Utilization of public transportation depends on the safety and reliability of the transit systems. Where latter is undermined by delays, their frequent occurrence and propagation through the transit network. Delays in public transportation systems can be categorized as recurrent or non-recurrent. Identifying the sources for recurrent delays and their patterns can help design intelligent public transportation where such delays are reduced. One of the promising directions to tackle the accumulated delays is development of dynamic and systematic right of way adjustments for train routes at bottleneck locations. Railway signal failure is an impactful traffic disruption. Estimating fixture time along the development of smart scheduling algorithms and train re-routing protocols is crucial to untangle and decrease congestion created by these failures.

A review of online dynamic models and algorithms for railway traffic management is presented in [F4]. Online control approaches use real-world data that includes position, timing and speed to return trains to their pre-planned schedule. Alternatively a set of control action could be applied to re-route delayed trains, thus transitioning the system into a different desired state. In general, such systems are a part of a broader iterative frameworks that adjust the forecast and the solution over time, in a closed loop control setup inspired by a model predictive control. The control actions include all measures that a controller might take to change the traffic to a certain desired state. Possible action could include adjustments of time, speed, train order, local route changes, i.e. following an adjacent to original route and arriving to possibly different platforms at the same stations, or even global route adjustments, i.e. following possibly a different path to the same destination, including service changes, like adding stops or cancelling trains.

Control approaches can be further divided into open and closed loop approaches. Open-loop approaches generate rescheduling solutions based on the perfect information about the current status of infrastructure, train positions and speeds, and precise predictions of delay characteristics and expected time of future events. They are run only once. Most optimization procedures assume this setup as a given problem is solved once and for all. In turn, in a closed-loop approach, the optimizer is iteratively called at subsequent times $t_1, t_2 \cdots, t_n$, each time defining a stage with an expected traffic situation and an actual traffic. The expected traffic at time t_{i+1} depends on the actual traffic at time t_i, which in turn depends on the control actions computed at all preceding times $t_1, \cdots t_i$.

An essential part of online dynamic approaches is the prediction of future deviations and delays from the planned schedule. Frequently this is accomplished by considering the dynamics of forecasts over time. In general road based transportation is focused in short term predictions, while railway management systems are rather keen to obtain predictions over longer horizons of time. The forecasting can be based on simulated or real world traffic information. Prediction models operate based on full future information, continuous information updates, discrete information updates or assume no knowledge of the future at all. Most real world systems that operate today adhere to the last approach as they rely on exact knowledge of the present, but largely neglect future predictions.

Mathematically dynamic train routing problems can be classified into modeling time and capacity, the objective of the railway traffic management, the degree of stochasticity and the level of detail and layout of the infrastructure. Dynamic routing problem could be represented by a coordinated sequence of actions linearly constrained based on discrete events that change the system's state. The constrained problem is then usually solved by linear programming approaches. System state events can be expressed as a time event graph, and further summarized into a compact set of expressions and constraints. Recently, the alternative graph [F7] has been used for the purpose due to its flexibility in representing most constraints and variables in a generalized job shop scheduling model.

Most of the approaches consider time-continuous variables, allowing event times to assume any value in $\mathbb{R}$. Another possibility is to model the problem variables using a discrete set of defining times, routes and connections and to address the problem as an assignment problem, thus greatly reducing the complexity of the problem.

An important aspect that has to be taken into account, while modeling a railway infrastructure is an exclusive infrastructure access. Thus when re-allocation, re-ordering or re-scheduling trains in a dynamic manner, safety considerations should be guaranteed, i.e. the algorithm should ensure no allocation of train routes will result in a conflict in the network. To avoid enumeration of all the possible conflicting orders, cumulative occupancy constraints should be used [F1]. This allows formulation of set-covering constraints to ensure resource allocation restrictions are met, and consequently in a potentially more accurate linear relaxation and a more efficient solution.

The railway traffic management objective is to improve the performance of the railway traffic by eliminating deviations from a pre-planned schedule and by minimizing passenger travel time. Some solutions simulate the railway traffic flow, then act based on the observed state. Other use heuristics to find relatively good solutions. Both approaches lack the precise definition of an objective function and fail to quantitatively evaluate own performance. On the other hand, mathematical optimization models have a well-defined objective functions that can account for factors such as delay, travel time, deviation from the schedule and operation cost. However they are much more computationally expensive.

The formulation of the problem is made either time-synchronous, implying that decisions are taken as time progresses or asynchronous, implying that choices that are taken in the present are aware of their future implications and are determined using full future information. The models differ in the level of detail and the layout of the infrastructure that is considered. Track sections, switches and the influence of safety systems are individually modeled in some works. Alternatively, only the merge and passing points are modeled explicitly as conflicting resources with finite capacity, whereas the rest of the system is modeled roughly.

The simplest models consider each station or track as a single resource with unlimited capacity analogous to a timetable problem. Consequently, depending on the model resolution, the test case could be a single line or a complex station, to a terminal area of a metro line, to a full network. The layout and the details of a model can include single or double track, crossings and a number of destinations. The modeling complexity is due to two factors. The transitioning from line to network, where continuity in routing of vehicles has to be modeled. The second comes from considering a mixed track type network, where different usage rules in single and double track have to be modeled.

A few other railway problems are deeply connected to the rescheduling problem, namely, the crew scheduling, rolling stock scheduling and delay management. The crew scheduling is a problem of assigning drivers to trains, so that all planned services can be run while minimizing the cost of stand-by personnel. Train services usually operate over hundreds of kilometers wide areas and lasts for hours, therefore solving this multi-factor problem in an optimal manner is computationally expensive. Another necessary requirement is to have an available vehicle. Rolling stock scheduling is the problem of assigning a vehicle to every train service that is intended to run. Spare vehicles are rare and costly. Therefore, services can be significantly delayed if rolling stock is not available at a given time and location, as prescribed by the plan. Finally, there is a need to decide which transfer connections should be kept or dropped while running delayed traffic. This is a delay management problem and it is inherently online, i.e., the expected delay of both feeder and connection train decide crucially an outcome of the optimization model. Most optimization approaches focused on the delay management problem under full information paradigm.

2) Algorithms for Smart Bus Transportation: Dynamic bus scheduling algorithms based on time dependent passenger information are believed to be a step in the direction of future reliable and sustainable public transportation systems. Moreover, implementation of such algorithms is expected to improve flexibility, reduce travel time, traffic congestion and emissions.

Demand responsive transit (DRT) or mobility on demand services have significantly evolved in the past few years, and are already operational as a complementary mode of public transportation around the world. DRT services are coupled to the concept of ride-sharing and ride matching, where a passenger joins a ride of another person when they both share an overlapping route segment. Real time dynamic matching between passengers and buses on public transportation lines falls directly in the above notion. Although, additional factors should be weighed in, such as passenger travel time, fleet utilization and energy consumption.

Public bus schedules usually have some form of a known expected time interval between subsequent arrivals at a given station, which is known as service reliability. However, service schedules that are independent of real time bus stop occupancy or of the actual demand on the service route are known to be sub-optimal. Several strategies are known that achieve better matching between origin-destination demands along the route and service coverage. A core idea is to design flexible routing and scheduling algorithms that combine express services that stop at few major passenger hubs, with regional services that cover all the local stops. Public bus Flexible Routing and Scheduling strategies could be classified into four categories: Zone Scheduling, Short Turning, Deadheading and Dynamic Stop Skipping.

In Zone Scheduling the whole route is divided into several zones. The inbound buses serves every bus stop within a single zone and then run uninterruptedly to the end point of the service. Outbound vehicles operate in the opposite manner. Potential advantages of this strategy are shorter travel times and a significant reduction of the required vehicles and drivers.

A Short Turning strategy assumes side-by-side operation of both short-turn and full length services along the same route. The express service is particularly suitable for routes where the demand in concentrated inside the specific zone and decreases significantly outside of it. The short-turn service covers only the high-demand zone, whereas the full length service serves all stops of the route. Short-turn services require fewer full length trips compared to a full length service, thus resulting in a lower number of service vehicles. Splitting the line into short and full length trips, designing the schedules of both trips that balance passenger loads and minimize the total fleet size and passenger wait time has a considerable impact on the solution and thus has to be carefully done.

Deadheading, is a technique to schedule some service vehicles to run empty and skip serving a number of stations at the beginning or at the end of the routes to save time and reduce the headways at later stations. Deadheading in the context of real-time transit control is investigated in [F5]. The problem is formulated as a nonlinear quadratic program, and the objective is to determine which vehicle to deadhead and at which stations.

Dynamic Stop Skipping is another practical strategy, used on routes where certain stops have a higher demand than others. The idea is to allow late and behind the schedule vehicles to skip certain low demand stops in order to compensate for accumulated delays. In contrast to the three other strategies, which are mostly off-line and designed at the service planning level, Dynamic Stop Skipping is determined and updated in real time. A negative effect of the approach is that some passengers might have their stop skipped and therefore have to wait for at least another headway to reach their destination.

The Dynamic Stop Skipping problem is usually formulated using non-linear integer programming. A real-time optimization model for dynamic scheduling of transit operations is presented in [F6]. The suggested strategy is to let service vehicles operate in pairs where lead vehicle provides an all-stop local service and the following vehicle is allowed to skip some stops as an express service. The underlying scheduling problem is formulated as a nonlinear integer programming problem with the objective of minimizing the total costs for both operators and passengers.

Information management techniques and algorithms have the potential to be used for updating public buses schedules once fed with inputs on current and predicted traffic conditions. A dynamic bus arrival time prediction model based on automatic passenger counter data is investigated in [F3]. Automatic passenger counting systems provide bus occupancy, location and travel time information in public transportation systems in real-time. This information can be further used as an input for a variety of applications including performance evaluation, operations management and service planning. The proposed model consists of two components. The first is a neural network model that produces predictions of travel time for a trip based on time, date and weather conditions. The second component is a Kalman filter based dynamic algorithm that adjusts the arrival time prediction using real time bus location information.

3) Algorithms for Intermodal Transportation: Different modes of public transportation that serves population side-by-sided should be synchronized in order to decrease overall waiting time and congestion. A higher level management system that performs this intermodal synchronization is proposed in [F2]. The authors suggest to activate a number of station outbound lines to meet the existing demand, in a way that balance operation versus passenger costs.

In order to increase the efficiency of public transportation services, passenger waiting time in hubs and terminals should be reduced. Hubs are stations where passengers change their line or transportation mode. The objective in the schedule synchronization problems is to maximize schedule synchronization, given a set of transit lines that intersect at some point in the transportation network. Which in turn will minimize passenger waiting times. Recently combinatorial optimization techniques, such as tackling the task as an assignment problem, were applied. However, most proposed models for this type of problems have a complexity that limits the applicability of the solution for real world applications.

VIII. Shared Mobility, Demand Responsive Transportation and First and Last-mile solutions

A. *Human Shared Mobility and Demand Responsive Transportation*

Shared mobility systems have recently gained a considerable attention. The research community perceives them as a sustainable and eco-friendly transportation that will drastically reduce congestion levels. The focus is on people sharing the rides, but also on combining transportation of people and freight on the same trip. According to [G4] the three major challenges to wide adoption of ride-sharing concept and consequently a decreased use of private cars are the absence of attractive mechanisms, satisfactory ride arrangements and trust among unknown passengers.

Operations research models that allow driver and rider matching in real-time, and optimization challenges that arise are reviewed in [G1]. A review on a demand-responsive ridesharing named dial-a-ride is presented in [G7]. Dial-a-ride problems are problems that focus on designing vehicle routes and time schedules in a demand-dependent modes of transportation. In the standard dial-a-ride problem, operational costs are minimized, under the full satisfaction of demand and service level requirements. Each user requests a trip from an origin to a destination of choice, subject to a number of service level requirements. The service provider suggest efficient vehicle routes and time schedules, meeting these requirements and all the additional constraints that arise in a pickup and delivery problem.

Real-life systems may require three extension types to the standard problem definition. The first is to allow additional service characteristics such as heterogeneous users, vehicles or drivers, more complex routing properties and different service level specifications. The second extension is to support a wide range of operational and service-related objectives, instead of a single operational objective. The third extension involves the information available to the service provider. The standard problem assumes that data is deterministic and known before vehicle routes and schedules are designed, however real-world problems involve dynamic or stochastic information on travel times, requests and user behavior.

The main objective of shared mobility human transportation systems is to minimize the number of used vehicles and consequently to reduce traffic congestion and environmental impact. Shared mobility comes in a variety of forms, the main ones are ridesharing, carpooling, car sharing and dial-a-ride services. Ridesharing allows people that wish to travel to close destinations at approximately the same time to share a vehicle for the trip in order to reduce the individual trip expenses. It could be either pre-arranged, when both driver and rider origins, destinations, departure and arrival times are known a-priori or dynamic when all the travelling information is generated and obtained on the fly.

Dynamic ride matching problems are solved in two stages, first, an efficient vehicle routes is computed, then, followed by an assignment of passengers to vehicles. Conflicting objectives, like maximizing the number of matched drivers and riders while simultaneously minimizing operation cost and passenger inconvenience, are inherent.

A distinct variant of the ridesharing is carpooling, where the objective is to determine the subsets of travelers that will share the same trip and the paths these shared trips should follow in order to minimize travelling costs. Flexible carpooling to battle the pre-arranged nature of carpooling services has been introduced by [G11]. The idea behind he concept is that a meeting point and departure times are known in advance to potential participants, and riders board cars on a first come first served basis. The dial-a-ride (DAR) that was described earlier provides shared trips between any origin and destination in response to advanced passenger requests within a specific area. In contrary to the other variants of shared mobility that were presented, DAR drivers are professional and do not need to get to a specific destination at a specific time.

In a shared-taxi approach [G5] one optimally assigns passengers to taxis and determines the optimal route for each taxi. The difference between shared taxi problem and dial-a-ride problem is that the objective is to minimize the response time to passenger request as opposed to minimize operation costs which is the objective of DAR. Figure 5 presents a graphical classification of shared mobility problems.

Different variants of shared mobility problems with the focus on the modelling approaches and solution algorithms are analyzed in [G8]. A set of transportation requests that should be fulfilled and which represents passenger delivery from origin to destination is introduced. Since a service could be provided by more than a single resource, the detailed planning of the shared trip under various constraints should be accomplished.

Shared Mobility objective functions could be classified into two categories: operational objectives and quality-related objectives. Operational objectives are used if we aim to optimize system-wide operating costs, e.g. the minimization of the travel distance, trip time or the number of required operational vehicle, or the maximization of fulfilled request number. Quality-related objectives are there if we aim to improve the quality of the provided service, e.g. minimize passenger ride or waiting time. Latter objectives may result in better performance from the rider's point of view, however they are not necessarily optimal from the system point of view and vice versa. In order to balance between the collective and the individual perspectives both types of objectives should be combined. Most works optimize a single operational objective, although some works do combine operational with quality-related objectives. Multi-objective systems that combine two or more objectives, most commonly, by a weighted sum approach.

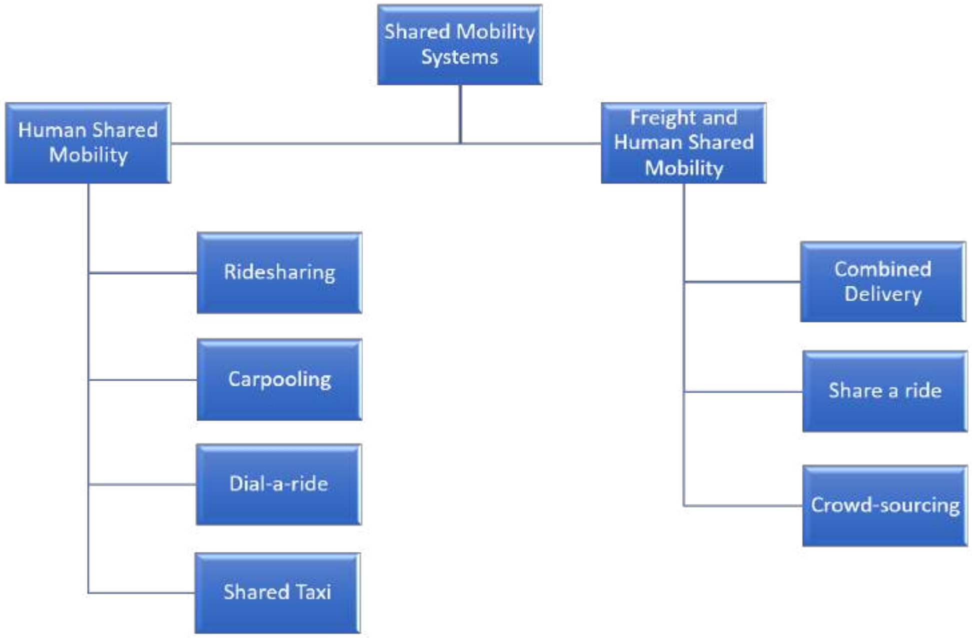

Fig. 5: Shared mobility systems problem classification.

The shared mobility problem is a generalization of the vehicle routing problem and is thus NP-hard problem. Dynamic and stochastic settings increase the complexity of the solution as well. Both exact and heuristic approaches are applied to solve variants of the problem, however due to the complexity of the problem mainly approximations or heuristic methods are economically feasible. Since urban shared mobility problems are naturally very large, even heuristic approaches are currently too slow if applied in online manner.

Stochasticity could appear in shared mobility problems in two forms. In the first, transportation demands, are updated continuously at time and location. In the second, travel time prediction errors that are accumulated due to estimation of future events such as congestion, accidents and weather conditions. Few approaches are used to solve stochastic shared mobility problems. Stochastic programming with recourse, where a decision is made and the expected resulting costs are minimized afterwards. And a multi-scenario approach where the expected costs are estimated while evaluating the solution on a set of different scenarios [G8].

B. Freight and Human Shared Mobility

Freight and human shared mobility algorithms are reviewed in [G8]. Shared mobility concepts and algorithms from the previous section could also be applied to address logistics problems such as parcel deliveries. While urban roads become heavily congested, reducing the number of freight delivery vehicles inside the cities is an appealing concept. The efficiency of people and goods ride sharing and potential challenges are investigated in [G3].

Ride sharing between humans and cargo in the context of last mile solutions and intelligent city logistics, trends, challenges and open problems are reviewed in [G10]. Crowd-sourcing logistics model to deliver goods in urban areas, that utilizes available capacity of ongoing trips, the crowd, is studied in [G9].

In a setting, where passengers and goods are to be transported, a decision whether to use a pure freight or people transportation network, or a combination of two has to be made. The share-a-ride problem in which people and parcels are handled in an integrated way by the same taxi network is analyzed in [G6]. In this model, a taxi drives in an urban area with the purpose of transporting passengers, but are also able to deliver goods, from an origin to a destination, if these deliveries do not add considerable time to the passenger trips.

The ability to share rides between humans and goods, gave birth to the concept of a crowd-source delivery. Crowd-sourced deliveries let outsource delivery services to a large number of independent individuals instead of one distribution agency. It is based on sharing excess and underused assets, i.e. the excess capacity of journey that already takes place, in order to make deliveries. The mixed ridesharing of passengers and freight shares a lot of similarities with the passenger-only ridesharing problem. However, it posses some complicating features as well, such as transfers, synchronization, capacity constraints etc. Furthermore, in order to successfully combine passengers and freight there should be no significant negative impact on people when goods are transported along the journey.

C. First and Last-Mile Solutions

Public transit is expected to take a central role in future urban transportation, mainly due to a limited infrastructure capacity and the need for low-impact transportation modes. In addition to reliability and accuracy improvements, it is essential to promote improved accessibility to public transportation hubs. Rapid and reliable services for the first and last-mile legs of the journey are necessary to reduce the time it takes passengers to complete their journey from home or office to the transportation hub and vice versa. First and last-mile services synchronized with public transportation services might eventually prove to be a cost effective alternative to private car usage.

The deployment of efficient and reliable autonomous personal transport vehicle fleet will dramatically change the prevailing private car ownership mode. Fleet management, including the selection of station locations, matching user request to a vehicle that will fulfill this request based on a desired optimization criteria, will allow the successful incorporation of this transportation mode into our lives.

Autonomous mobility services and especially robo-taxi fleets are expected to revolutionize urban transportation habits completely [G2]. The control of robotic mobility-on-demand systems (robo-taxis) is investigated from a queueing theoretical perspective [G12]. In the presented work, the vehicles re-balance themselves in order to provide acceptable quality of service throughout the entire network. The problem is modeled as a closed Jackson network model with passenger loss. The authors show that an optimal rebalancing algorithm that minimizes the number of rebalanced vehicles can be found using a linear program.

IX. Human-machine Interface Algorithms

Although it is expected that decades will pass until fully autonomous vehicles will dominate the road, it is only a mater of time before vehicles with substantial autonomous capabilities will become a prevalent mode of transportation. In order to properly define the autonomous capabilities of a vehicle, the National Highway Traffic Safety Administration (NHTSA) has defined 6 levels of increasing automation.

Vehicle without any automation with all the driving tasks performed by a human is defined to have level 0 automation. Level 1 implies that some driving assistance features are provided by the vehicle. Level 2 refers to partial automation where a vehicle has combined automated functions, for example, acceleration and steering. However the driver must remain engaged in the driving task and must monitor the environment constantly. Level 3 refers to conditional automation, where the driver is not required to monitor the environment, although he has to be ready to take control of the vehicle at all times when notified by the vehicle. Level 4 is a high automation, where a vehicle is capable of performing all driving related functions under certain conditions. Level 5 is the highest degree of automation and is regarded as fully autonomous. A level 5 vehicle is capable of performing all driving functions under all conditions. Though in both level 4 and level 5, the driver still has the option to control the vehicle.

This categorization is standardized in the SAE standard for Driving Automation Systems. Since vehicles at the lower automation levels require certain degree of collaboration between the human driver and the vehicle, an interface that transitions the control over the vehicle is required. Such interface falls under the category of Human-Machine Interface (HMI). The role of a HMI is to support an interaction between the human driver and the vehicle's Advanced Driver Assistance Systems (ADAS). HMI research in transportation investigates switching of control between human and autopilot methods, situation detection when a human driver should be alerted, when the control could be shifted to the vehicle and when it needs to fallback to the human driver for safety. Additionally HMI studies are concerned with which information to display and how to effectively present it.

A. Control Transition Interfaces

The control transition interfaces between the human driver and the machine are usually divided into Handover and Take Over Requests. A Handover request is a request to transition the control of the vehicle from human driver to machine. Conversely, a Take Over request seeks to transition the control over the vehicle in the opposite direction. Research in this area includes test and creation of shift control scenarios with an emphasis on the direction of the control transition. Substantial effort is invested in the impact analysis of abrupt versus gradual transition requests, and control transitions under various driver awareness states. HMIs that allow driver to perform numerous non-driving tasks while the vehicle is in the autopilot mode, and to alert the driver to assume vehicle control, when and if such need arises, is an additional avenue of research.

A categorization framework and a control transition interfaces in partially autonomous vehicles are examined in [H10]. The surveyed interfaces are categorized by the transition initiator, transition failure fallback mechanism and by various transition request presentation methods. The purpose of this categorization is to distinguish the different attributes that are required for implementing the control transitions.

An initiator of a transition could be either the human or the vehicle. This distinction is extremely important since human driver is aware of the driving conditions when transition request is initiated by him, while in the machine initiated request case driver needs to get properly notified before assuming the control. Humans require a substantial time to assess the current driving situation, prepare and plan accordingly. Fallback mechanism determines who is responsible for vehicle control if a control transition fails and which actions should be taken to ensure the driver and environment safety. A list of possible actions include slowing down and stopping either inside or outside the vehicle's lane. Information presentation form influences the driver and HMI as well. Information could be presented in form of speech, gesture and touch commands, and notifications that form the interaction basis between human and vehicle.

B. Verification Methods for Human-Machine Interface

Verification methods for automated vehicle HMI development guidance are presented in [H12]. The work proposes a set of principles to be considered in the design of driver-vehicle interface. A formal verification-based HMI design approach is presented in [H2]. Formal verification is a technique that is used to mathematically prove that a given model of a system exhibits desired properties. Formal methods use mathematical language for unambiguous specification, modeling and verification of systems. Computational models and mathematical modelling languages that are used to model systems include Finite State Automata, directed graphs, Buchi automata, Petri nets and μ calculus. Two main methods of large systems formal verification are automated theorem proving and model checking.

A formal approach for analyzing human-machine interaction through the detection of design errors in the interaction process and through the correctness verification of the interaction in automated control systems is introduced in [H4], [H3]. The work introduces an interface evaluation methodology that assesses whether a machine operator is provided with all the necessary information to successfully perform a desired task. It also addresses the proficiency of the machine behavior information provided to the user. The proposed methodology can be applied to the verification of large systems and is illustrated through a case study of the interaction between a pilot and an aircraft autopilot.

A comprehensive review of formal verification methods for Human-Automation Interaction (HAI) is presented in [H1]. The review surveys formal verification methods that focus on abstract discrete HAI models and analyze, and detect problems in human machine interactions. It specifically addresses usability and mode confusion in human machine interface. Moreover, it focuses on why logical and valid human behavior could lead to violation of desired system properties when interacting with a system's automation interface. Authors focus on providing guarantees that the human-machine interface behaves in a safe, planned, and intended way.

Thereupon, they proceed to identify conditions which lead to mode confusion, and ultimately to undesired behavior. Such identification could be based on a formal verification of properties against a formal model of the human automation interface. Other approaches find potential mode confusions by using a model of the human-automation interface together with a basic model of the machine's behavior. This type of analysis is applied if model state space transitions may lead to abnormal states. One looks for mismatches between an anticipated result for a human's action on the resulting state of the interface, or detects interface states that do not represent faithfully the state of the automation.

However, these models do not include information regarding human operator's knowledge about current and expected system states, future modes and mode changes. Therefore, it is not clear that mode confusion will occur even if a violation is found. To address the problem, two separate approaches that allow potential mode confusion to be explicitly discovered using human operator modeling were proposed. The first approach analyzes models further referred as Human Mental Models. Those models assume that human operator creates a mental model of how the machine behaves, therefore it can be represented as a formal finite state transition system. The second approach models are known as Human Knowledge Models. Human operator is assumed to possess the proficiency of utilizing human-machine interface to achieve the their goals. Therefore, second approach models adopt formal models to represent human operator's knowledge in order to detect potential mode confusions. This approach contradicts the human mental model approach that tries to formally model the way a human operator perceives the machine's behavior.

Formal verification of system properties focuses on verification of large, detailed and complex models and results in a significantly larger state space compared to the more abstract human machine interface models. The above approaches do not explicitly model human behavior and thus can only be used to find preconditions for system failure. Another set of approaches seeks to identify the conditions under which an analytically modeled human task execution behavior might result in system problems that are not directly caused by an inappropriate human-machine interface. The developed analytic models represent the mental and physical activities performed by humans in order to cause the machine to function successfully according to its stated goal. Thus formal verification models could be categorized into Task Models and Cognitive Models. Tasks models are concerned with modeling the observable manifestation of human behavior whereas Cognitive models describe the cognitive process that drives the observable human behavior.

A formalism for human-in-the-loop control systems is defined in [H9]. The authors synthesize a semi-autonomous controller from high-level temporal specifications. The proposed controller requires occasional human intervention and provides a balance between human-in-the-loop control systems that interact frequently with a human operator and fully autonomous controllers. Fully autonomous controllers are synthesized from mathematical specifications that guarantee correct operation and can be specified in a formal language such as linear temporal logic (LTL) [H14]. However, an inadequate specification can result in a conclusion that no controller exists to perform the desired action or in a controller that is not implementable in real world settings due to specification abstractions.

In semi-autonomous vehicle scenarios the human driver should act as a fail-safe mechanism in case the automation encounters a situation it cannot handle successfully. Therefore, four criteria should be met for successful integration of a human-in-the-loop-controller in the operation of semi-autonomous vehicles. The first is a controller monitoring capability to determine if human intervention is required based on past and present readings of the system and the environment. The second is that the controller operates in a minimally intervening manner and invokes the human operator only when the need arises. The third is a requirement for a controller to be prescient, implying that it has the ability to determine ahead of time that a specification may be violated and that it notifies the human driver with a sufficient warning to let the human assume the control of the vehicle. The last criterion is that the controller should be conditionally correct, meaning that it operates correctly up to the point where it determines that the human operator should assume control.

The proposed human-in-the-loop controller consists of three components: an automatic controller, a human operator and an advisory control mechanism that is responsible for switching between the two controllers when needed. The construction of the human-in-the-loop controller is researched in the context of reactive synthesis from LTL specifications. Where latter is the process of automatic synthesis of a discrete system that reacts to environment changes in a way that satisfies a given LTL specification. The authors propose an algorithm to synthesis human-in-the-loop controllers satisfying the four specified criteria and show a potential application of the controller in driver-assistance systems.

C. Distraction Detection and Analysis

Monitoring of the human operator awareness and the ability to detect driver distraction is an important aspect in semi-autonomous vehicles in both modes of operation. This capability is essential for safe and effective collaboration between the human operator and the vehicle's ADAS systems. The driver awareness monitoring is performed by gathering and analyzing the information from in-cabin sensors. Driver distraction and inattention are defined as a lack of attention to activities that are critical for safe driving [H6]. Driver activities are classified into primary tasks that are essential to operate and control the vehicle's trajectory, and secondary tasks that distract the driver attention from the primary task and results in a degraded driving performance. Driving distraction is usually caused by a trigger activity, that can be detected once it occurs. Subsequently a warning notification can be issued by an automated system to the driver, thus drawing the driver's attention back to the primary driving task.

Secondary tasks vary greatly in the amount of the distraction determined by a number of factors. Amongst them are the duration of the distraction, its frequency, the complexity of performed distracting activity, the ease of transitioning back to the primary task and the individual ability to share the attention between the simultaneously performed tasks. Monitoring a driver awareness and performance can be completed by analyzing various signals obtained from the physical contact points of the human operator and the vehicle including the steering wheel, the pedals, the driver seat or by tracking the vehicle's speed. Additional visual sensors can monitor the driver attention and analyze the driver glance behaviour.

The analysis of distractions relies on the hypothesis that a safe and acceptable driving behaviour should result in stable and smooth vehicle dynamics. Another hypothesis states that undesired driving behaviour occurs infrequently compared to a desired one, and that both exhibit similar characteristics, even for different drivers. Therefore, it is possible to cluster desired driving features together in a theoretic feature space, and allow classification solutions to detect an undesired driving behaviour that might correspond to a driver distraction.

D. Maneuver Recognition

Semi-autonomous vehicles must possess several important capabilities before they could be allowed to share an environment with vehicles controlled by human drivers. Among them are an understanding of the surrounding environment and traffic, monitoring the driver awareness, and the ability to predict nearby vehicle movement. Furthermore, a human operator driving performance can be monitored, evaluated and compared to the past behaviors in order to detect deviations from a normal driving patterns. Detection of sudden variations in a vehicle dynamics might suggest that the vehicle poses a threat to nearby surroundings and that caution should be taken to prevent a potential accident.

Detection of sudden variations in own vehicle dynamics can prove beneficial. Moreover, if driver attention was distracted, a corresponding notification to the operator could steer it back to the primary driving task. Several factors may influence vehicle dynamics while a human driver is in control of the vehicle. Between them, weather and traffic conditions, driving skills and driver mental and physical conditions. Certain vehicle dynamics are common among different drivers and therefore it is possible to recognize different maneuvers executed by nearby vehicles. Predicting the type of maneuver can assist in predicting vehicle future trajectory and lead to better planning of the own vehicle course.

Maneuvers form the basis of a vehicle trajectory. Understanding the dynamics of vehicle maneuvers can incite a better understanding of how human operator controls a vehicle and how driving performance changes over time. Drawing inspiration from the structure of speech where phonemes form segments of words, [H15] suggests to call the smallest meaningful units of a driving pattern drivemes. Drivemes form maneuvers, and maneuver sequences form the entire trajectory of a vehicle. Consequently, monitoring and identifying changes in drivemes allows the vehicle's safety systems to issue warnings if an abnormal pattern is detected. It also allows to predict behavior of drivers in the vehicle environment.

Several works investigated the topic of vehicle maneuver recognition. A vehicle trajectory prediction method based on motion and maneuver recognition is presented in [H7]. The trajectory prediction is a combination of short and long-term predictions. Short term predictions are based on Constant Yaw Rate and Acceleration motion model, while long-term predictions rely on recognition of a maneuver from a predefined maneuver set. The maneuver could be classified via a maneuver recognition module based on a comparison between the center lines of road lanes and a local curvilinear model of the vehicle path. The maneuver recognition module exploits the fact that both the maneuver and the vehicle path are strongly correlated with lanes, and therefore identifies target lane using this information. The considered maneuvers consist of keep lane, change lane (right or left) or turn at an intersection. Those are the basic building blocks for more complicated maneuvers.

The trajectory can be predicted for both the ego-vehicle and the surrounding vehicles. An online maneuver recognition and multimodal trajectory prediction framework for intersection assistance using non-parametric regression models is presented in [H16]. Three dimensional Gaussian regression models are constructed from two dimensional vehicle trajectories to model the spatio-temporal attributes of vehicle trajectories. A maneuver is recognized by comparison the likelihoods of the observed vehicle trajectory to each individual maneuver regression model. The set of maneuvers is tailored for intersection handling and includes left and right turns, straight and stop and go maneuvers. This is achieved by using a sequence of successive position and velocity measurements and selection of a regression model that explains the observations the best.

A rule-based intention recognition algorithm based on lateral and longitudinal motion cues for maneuver recognition is presented in [H13]. The algorithm utilizes simple logic rules to determine keep-or-switch lane intentions of the neighbor vehicles prior to the completion of the lane change maneuver. Numerous machine-learning and statistical modelling approaches were used for maneuver recognition as well. Bayesian models are used for automatic maneuver recognition [H5] due to their ability to handle uncertain measurement data, which is often the case when information is gathered from the vehicle sensors. Probabilistic finite-state machines approach for maneuver recognition is presented in [H8]. The approach combines fuzzy logic rules to model basic maneuver elements and probabilistic state machines to capture all the possible sequences of basic maneuver elements that form a driving maneuver.

Hidden Markov Models (HMMs) are used to recognize driving patterns in [H11]. HMMs represent a stochastic Markov process over finite or infinite state space with corresponding transition probabilities. At discrete time intervals, the process may change its state, although state changes are hidden from the observer. Measurements of acceleration and speed from a real vehicle are collected, filtered, normalized, segmented and quantified to derive a symbolic data representation that can be used in a discrete HMM. Observation sequences are manually selected and classified as specific maneuvers. Then those sequences are used to train and evaluate a recognition model. Finally, a separate HMM model is trained to detect each type of maneuver, and at test time the maneuver that corresponds to the model that outputs the highest probability is chosen as the maneuver that corresponds to the observation sequence.

X. Topics for Further Investigation

As a conclusion to this survey we provide a list of topics that should be further investigated in order to promote the development of Intelligent Transportation Systems. These topics are divided based on the identified core research areas we have considered above. Advancing research in these fields will play a crucial role in the development of intelligent transportation.

1. Ground Traffic Management (Freeway and Urban) Algorithms
   - Vehicle routing problems using time varying graphs.
   - Evolution game theory.
   - Feedback dynamic traffic assignment.
   - Shortest paths on graphs.
   - Social vehicle route selection and Stackelberg algorithms.
   - Trajectory planning.
   - Speed planning.
   - Collision avoidance.
   - Multi-vehicle interaction
     - Resource allocation problems involving a wide range of possible approaches:
       - Linear and non-linear programming.
       - Conflict-region algorithms.
       - Centralized and decentralized. synchronization protocols.
       - Graph coloring.
       - Mimetic and machine-learning based optimization, transfer learning and prediction.
2. Air Traffic Management Algorithms
   - Dynamic routing problems on time varying graphs for air space sharing and dynamic air space allocation.
   - Collision avoidance for unstructured air space using conflict detection and resolution algorithms.
   - Scheduling and sequencing algorithms for vertiports traffic management for Urban Air Mobility.

- Optimal control and model predictive control.
- Reachability analysis allowing for safe platooning
- Path planning using optimization approaches such as mixed integer optimization for incorporation of logical constraints in trajectory design and non linear programming for generation of flyable paths.
- Reinforcement learning for path planning.

3. Perception, Data fusion, Processing and Aggregation Techniques
   - Map generation and fusion.
   - 3D object localization.
   - Fusion algorithms of data from heterogeneous data sources for road network traffic representation.
   - Cooperative vehicles
     - Vehicular communication.
     - Cooperative localization.
     - Motion coordination.
     - Sensor management of autonomous vehicles.
     - Message dissemination protocols and infrastructure assisted coordination.
     - Cooperative navigation using distributed consensus.
     - Model predictive, $\mathcal{H}_\infty$ and sliding mode controls.
     - Platoon coordination, virtual platoons, platoon and string stability theory.
     - Speed harmonization algorithms and optimal control based speed harmonization.
4. Traffic Simulation and Prediction Models
   - short- and long-term traffic prediction algorithms based on statistical and time series analysis, auto-correlation functions, neural networks, pattern recognition and machine learning approaches.
   - Microscopic car-following models.
   - Particle-hopping models.
   - Gas-kinetic theoretical models.
   - Macroscopic fluid models.
   - Human-driving modeling using differential equations and acceleration correction algorithms.
   - Detailed vehicle dynamics modeling.
5. Smart Public Transportation
   - Adaptive route planning and scheduling for public transportation.
   - Train re-routing algorithms that incorporate future predictions.
   - Algorithms for demand responsive passenger balancing.
   - Prediction of occupancy and saturation in public transportation using automatic passenger information collection systems.
   - Algorithms for schedule synchronization for intermodal transportation.
6. Shared mobility, Demand Responsive Transportation and First and Last-mile Solutions
   - Algorithms for Ride-sharing and Carpooling such as matching and assignment problems on graphs, scheduling algorithms, combinatorial optimization approaches and algorithms for solving dynamic and stochastic shared mobility problems.
   - Algorithms for management of fleets of autonomous vehicles for first and last-mile transportation services.
   - Algorithms for combined human and parcel transportation.
   - Parking guidance and management systems algorithms- Data sharing algorithms for combined parking routing and traffic prediction, Markov chains and queuing theory for future occupancy of parking spots and machine learning approaches.
   - optimal sensor placement for parking applications.
   - Dynamic re-balancing based on predictions.
   - Optimal trading schemes and distributed decision making.
7. Human-machine Interface Algorithms
   - Algorithms for safe hand off and take over requests for partially autonomous vehicles
   - Maneuver recognition using statistical modeling and machine learning classification algorithms such as Bayesian models, hidden Markov models and decision trees.
   - Distraction detection and analysis.
   - Verification and Validation Methods for Human-Machine Interface

References

Overview of the Field

[A1] S.-h. An, B.-H. Lee, and D.-R. Shin. "A survey of intelligent transportation systems". In: *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE. 2011, pp. 332–337.

[A2] H. F. Azgomi and M. Jamshidi. “A brief survey on smart community and smart transportation”. In: *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2018, pp. 932–939.

[A3] J. Barbaresso et al. *USDOT's Intelligent Transportation Systems (ITS) ITS strategic plan, 2015-2019*. Tech. rep. United States. Department of Transportation. Intelligent Transportation, 2014.

[A4] A. M. De Souza et al. “Traffic management systems: A classification, review, challenges, and future perspectives”. In: *International Journal of Distributed Sensor Networks* 13.4 (2017), p. 1550147716683612.

[A5] R. Faria et al. “Smart mobility: A survey”. In: *2017 International Conference on Internet of Things for the Global Community (IoTGC)*. IEEE. 2017, pp. 1–8.

[A6] Y. Lin, P. Wang, and M. Ma. “Intelligent transportation system (its): Concept, challenge and opportunity”. In: *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*. IEEE. 2017, pp. 167–172.

[A7] E. Parliament et al. “Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport Text with EEA relevance”. In: *Off. J. Eur. Union* (2010), pp. 1–13.

[A8] K. N. Qureshi and A. H. Abdullah. “A survey on intelligent transportation systems”. In: *Middle-East Journal of Scientific Research* 15.5 (2013), pp. 629–642.

Freeway Traffic Modeling

[B1] O. Biham, A. A. Middleton, and D. Levine. “Self-organization and a dynamical transition in traffic-flow models”. In: *Physical Review A* 46.10 (1992), R6124.

[B2] R. E. Chandler, R. Herman, and E. W. Montroll. “Traffic dynamics: studies in car following”. In: *Operations research* 6.2 (1958), pp. 165–184.

[B3] D. Chowdhury, D. E. Wolf, and M. Schreckenberg. “Particle hopping models for two-lane traffic with two kinds of vehicles: Effects of lane-changing rules”. In: *Physica A: Statistical Mechanics and its Applications* 235.3-4 (1997), pp. 417–439.

[B4] L. C. Edie. “Car-following and steady-state theory for noncongested traffic”. In: *Operations research* 9.1 (1961), pp. 66–76.

[B5] D. W. Erickson et al. “Spurious phase in a model for traffic on a bridge”. In: *Journal of Physics A: Mathematical and General* 38.41 (2005), p. L659.

[B6] M. Fukui and Y. Ishibashi. “Traffic flow in 1D cellular automaton model including cars moving with high speed”. In: *Journal of the Physical Society of Japan* 65.6 (1996), pp. 1868–1870.

[B7] D. C. Gazis, R. Herman, and R. B. Potts. “Car-following theory of steady-state traffic flow”. In: *Operations research* 7.4 (1959), pp. 499–505.

[B8] P. G. Gipps. “A behavioural car-following model for computer simulation”. In: *Transportation Research Part B: Methodological* 15.2 (1981), pp. 105–111.

[B9] B. S. Kerner. “Experimental features of self-organization in traffic flow”. In: *Physical review letters* 81.17 (1998), p. 3797.

[B10] B. S. Kerner. “The physics of traffic”. In: *Physics World* 12.8 (1999), p. 25.

[B11] T. Nagatani. “Self-organization and phase transition in traffic-flow model of a two-lane roadway”. In: *Journal of Physics A: Mathematical and General* 26.17 (1993), p. L781.

[B12] K. Nagel and H. J. Herrmann. “Deterministic models for traffic jams”. In: *Physica A: Statistical Mechanics and its Applications* 199.2 (1993), pp. 254–269.

[B13] K. Nagel and M. Paczuski. “Emergent traffic jams”. In: *Physical Review E* 51.4 (1995), p. 2909.

[B14] K. Nagel and M. Schreckenberg. “A cellular automaton model for freeway traffic”. In: *Journal de physique I* 2.12 (1992), pp. 2221–2229.

[B15] K. Nagel et al. “Two-lane traffic rules for cellular automata: A systematic approach”. In: *Physical Review E* 58.2 (1998), p. 1425.

[B16] M. Rickert et al. “Two lane traffic simulations using cellular automata”. In: *Physica A: Statistical Mechanics and its Applications* 231.4 (1996), pp. 534–550.

[B17] G. Schutz. “Exactly solvable models for many-body systems far from equilibrium”. In: *Phase transitions and critical phenomena* (2000).

[B18] H. Spohn. *Large scale dynamics of interacting particles*. Springer Science & Business Media, 2012.

[B19] J. Vasic and H. J. Ruskin. “Cellular automata simulation of traffic including cars and bicycles”. In: *Physica A: Statistical Mechanics and its Applications* 391.8 (2012), pp. 2720–2729.

[B20] P. Wagner, K. Nagel, and D. E. Wolf. “Realistic multi-lane traffic rules for cellular automata”. In: *Physica A: Statistical Mechanics and its Applications* 234.3-4 (1997), pp. 687–698.

[B21] S. Wolfram. *Cellular automata and complexity: collected papers*. CRC Press, 2018.

[B22] S. Wolfram. "Theory and applications of cellular automata". In: *World Scientific* (1986).

Air Traffic Management

[C1] H. Balakrishnan. "Control and optimization algorithms for air transportation systems". In: *Annual Reviews in Control* 41 (2016), pp. 39–46.

[C2] H. Balakrishnan and B. Chandran. "Scheduling aircraft landings under constrained position shifting". In: *AIAA guidance, navigation, and control conference and exhibit*. 2006, p. 6320.

[C3] H. Balakrishnan and B. G. Chandran. "Algorithms for Scheduling Runway Operations Under Constrained Position Shifting". In: *Operations Research* 58.6 (2010), pp. 1650–1665.

[C4] J. Bellingham, A. Richards, and J. P. How. "Receding Horizon Control of Autonomous Aerial Vehicles". In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*. Vol. 5. IEEE. 2002, pp. 3741–3746.

[C5] F. Borrelli et al. "MILP and NLP Techniques for Centralized Trajectory Planning of Multiple Unmanned Air Vehicles". In: *2006 American Control Conference*. IEEE. 2006, 6–pp.

[C6] B. Chandran and H. Balakrishnan. "A dynamic programming algorithm for robust runway scheduling". In: *2007 American Control Conference*. IEEE. 2007, pp. 1161–1166.

[C7] D. E. Chang et al. "Collision Avoidance for Multiple Agent Systems". In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*. Vol. 1. IEEE. 2003, pp. 539–543.

[C8] M. Chen et al. "Reachability-based Safety and Goal Satisfaction of Unmanned Aerial Platoons on Air Highways". In: *Journal of Guidance, Control, and Dynamics* 40.6 (2017), pp. 1360–1373.

[C9] R. Deits and R. Tedrake. "Efficient Mixed-integer Planning for UAVs in Cluttered Environments". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 42–49.

[C10] H. Erzberger and E. Itoh. "Design Principles and Algorithms for Air Traffic Arrival Scheduling". In: (2014).

[C11] P. E. Hart, N. J. Nilsson, and B. Raphael. "A formal basis for the heuristic determination of minimum cost paths". In: *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107.

[C12] D. Jung and P. Tsiotras. "On-line Path Generation for Unmanned Aerial Vehicles using B-spline Path Templates". In: *Journal of Guidance, Control, and Dynamics* 36.6 (2013), pp. 1642–1653.

[C13] L. E. Kavraki et al. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE transactions on Robotics and Automation* 12.4 (1996), pp. 566–580.

[C14] I. C. Kleinbekman, M. A. Mitici, and P. Wei. "eVTOL Arrival Sequencing and Scheduling for on-demand Urban Air Mobility". In: *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE. 2018, pp. 1–7.

[C15] J. K. Kuchar and L. C. Yang. "A review of conflict detection and resolution modeling methods". In: *IEEE Transactions on intelligent transportation systems* 1.4 (2000), pp. 179–189.

[C16] S. M. LaValle. "Rapidly-exploring random trees: A new tool for path planning". In: (1998).

[C17] N. H. Motlagh, T. Taleb, and O. Arouk. "Low-altitude Unmanned Aerial Vehicles-based Internet of Things Services: Comprehensive Survey and Future Perspectives". In: *IEEE Internet of Things Journal* 3.6 (2016), pp. 899–922.

[C18] A. Otto et al. "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey". In: *Networks* 72.4 (2018), pp. 411–458.

[C19] P. Pradeep and P. Wei. "Energy-efficient Arrival with RTA Constraint for Multirotor eVTOL in Urban Air Mobility". In: *Journal of Aerospace Information Systems* 16.7 (2019), pp. 263–277.

[C20] M. Radmanesh et al. "Overview of Path-planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study". In: *Unmanned systems* 6.02 (2018), pp. 95–118.

[C21] A. U. Raghunathan et al. "Dynamic Optimization Strategies for Three-dimensional Conflict Resolution of Multiple Aircraft". In: *Journal of guidance, control, and dynamics* 27.4 (2004), pp. 586–594.

[C22] V. Roberge, M. Tarbouchi, and G. Labonté. "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning". In: *IEEE Transactions on industrial informatics* 9.1 (2012), pp. 132–141.

[C23] J. A. Sethian. "A Fast Marching Level Set Method for Monotonically Advancing Fronts". In: *Proceedings of the National Academy of Sciences* 93.4 (1996), pp. 1591–1595.

[C24] I. Simaiakis, M. Sandberg, and H. Balakrishnan. "Dynamic control of airport departures: Algorithm development and field evaluation". In: *IEEE Transactions on Intelligent Transportation Systems* 15.1 (2013), pp. 285–295.

[C25] I. Simaiakis et al. "Demonstration of reduced airport congestion through pushback rate control". In: *Transportation Research Part A: Policy and Practice* 66 (2014), pp. 251–267.

[C26] J. Sun, J. Tang, and S. Lao. "Collision Avoidance for Cooperative UAVs with Optimized Artificial Potential Field Algorithm". In: *IEEE Access* 5 (2017), pp. 18382–18390.

[C27] S. Wollkind, J. Valasek, and T. Ioerger. "Automated conflict resolution for air traffic management using cooperative multiagent negotiation". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2004, p. 4992.

[C28] J. Yang et al. "Distributed Cooperative Onboard Planning for the Conflict Resolution of Unmanned Aerial Vehicles". In: *Journal of Guidance, Control, and Dynamics* 42.2 (2019), pp. 272–283.

[C29] X. Yu and Y. Zhang. “Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects”. In: *Progress in Aerospace Sciences* 74 (2015), pp. 152–166.

[C30] B. Zhang et al. “Geometric Reinforcement Learning for Path Planning of UAVs”. In: *Journal of Intelligent & Robotic Systems* 77.2 (2015), pp. 391–409.

Sensing and Perception

[D1] V. K. Boda, A. Nasipuri, and I. Howitt. “Design considerations for a wireless sensor network for locating parking spaces”. In: *Proceedings 2007 IEEE SoutheastCon*. IEEE. 2007, pp. 698–703.

[D2] M. Caliskan et al. “Predicting parking lot occupancy in vehicular ad hoc networks”. In: *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*. IEEE. 2007, pp. 277–281.

[D3] J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd. “Smart parking: An application of optical wireless sensor network”. In: *2007 International Symposium on Applications and the Internet Workshops*. IEEE. 2007, pp. 66–66.

[D4] T. Delot et al. “A cooperative reservation protocol for parking spaces in vehicular ad hoc networks”. In: *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems*. 2009, pp. 1–8.

[D5] Z. He et al. “Mapping to cells: a simple method to extract traffic dynamics from probe vehicle data”. In: *Computer-Aided Civil and Infrastructure Engineering* 32.3 (2017), pp. 252–267.

[D6] S.-W. Kim et al. “Multivehicle cooperative driving using cooperative perception: Design and experimental validation”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2014), pp. 663–680.

[D7] S. Lee, D. Yoon, and A. Ghosh. “Intelligent parking lot application using wireless sensor networks”. In: *2008 International Symposium on Collaborative Technologies and Systems*. IEEE. 2008, pp. 48–57.

[D8] M. J. Lighthill and G. B. Whitham. “On kinematic waves II. A theory of traffic flow on long crowded roads”. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229.1178 (1955), pp. 317–345.

[D9] T. Lin, H. Rivano, and F. Le Mouël. “A survey of smart parking solutions”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.12 (2017), pp. 3229–3253.

[D10] Y. Lin, P. Wang, and M. Ma. “Intelligent transportation system (its): Concept, challenge and opportunity”. In: *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*. IEEE. 2017, pp. 167–172.

[D11] K. Nellore and G. P. Hancke. “A survey on urban traffic management system using wireless sensor networks”. In: *Sensors* 16.2 (2016), p. 157.

[D12] S. D. Pendleton et al. “Perception, planning, control, and coordination for autonomous vehicles”. In: *Machines* 5.1 (2017), p. 6.

[D13] F. Rempe et al. “Online freeway traffic estimation with real floating car data”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2016, pp. 1838–1843.

[D14] G. Revathi and V. S. Dhulipala. “Smart parking systems and sensors: A survey”. In: *2012 International Conference on Computing, Communication and Applications*. IEEE. 2012, pp. 1–5.

[D15] P. I. Richards. “Shock waves on the highway”. In: *Operations research* 4.1 (1956), pp. 42–51.

[D16] X. Sevillano, E. Màrmol, and V. Fernandez-Arguedas. “Towards smart traffic management systems: Vacant on-street parking spot detection based on video analytics”. In: *17th International Conference on Information Fusion (FUSION)*. IEEE. 2014, pp. 1–8.

[D17] X. Shen, S. Pendleton, and M. H. Ang. “Scalable cooperative localization with minimal sensor configuration”. In: *Distributed Autonomous Robotic Systems*. Springer, 2016, pp. 89–104.

[D18] P. Szczurek et al. “Learning the relevance of parking information in VANETs”. In: *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*. 2010, pp. 81–82.

[D19] M. Treiber, A. Kesting, and R. E. Wilson. “Reconstructing the traffic state by fusion of heterogeneous data”. In: *Computer-Aided Civil and Infrastructure Engineering* 26.6 (2011), pp. 408–419.

[D20] D. B. Work et al. “An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices”. In: *2008 47th IEEE Conference on Decision and Control*. IEEE. 2008, pp. 5062–5068.

Traffic Analysis and Optimization

[E1] M. S. Ahmed and A. R. Cook. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. 722. 1979.

[E2] M. Behrisch et al. “SUMO- Simulation of Urban Mobility: an Overview”. In: *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. 2011.

[E3] S. R. Chandra and H. Al-Deek. “Predictions of freeway traffic speeds and volumes using vector autoregressive models”. In: *Journal of Intelligent Transportation Systems* 13.2 (2009), pp. 53–72.

[E4] S. I.-J. Chien and C. M. Kuchipudi. “Dynamic travel time prediction with real-time and historic data”. In: *Journal of transportation engineering* 129.6 (2003), pp. 608–616.

[E5] D. R. Choffnes and F. E. Bustamante. “An Integrated Mobility and Traffic Model for Vehicular Wireless Networks”. In: *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*. 2005, pp. 69–78.

[E6] S. Djahel et al. “A Communications-oriented Perspective on Traffic Management Systems for Smart Cities: Challenges and Innovative Approaches”. In: *IEEE Communications Surveys & Tutorials* 17.1 (2014), pp. 125–151.

[E7] I. Lana et al. “Road traffic forecasting: Recent advances and new challenges”. In: *IEEE Intelligent Transportation Systems Magazine* 10.2 (2018), pp. 93–109.

[E8] W. Min and L. Wynter. “Real-time road traffic prediction with spatio-temporal correlations”. In: *Transportation Research Part C: Emerging Technologies* 19.4 (2011), pp. 606–616.

[E9] M. Rondinone et al. “iTETRIS: a Modular Simulation Platform for the Large Scale Evaluation of Cooperative ITS Applications”. In: *Simulation Modelling Practice and Theory* 34 (2013), pp. 99–125.

[E10] W. Shen and L. Wynter. “Real-time road traffic fusion and prediction with GPS and fixed-sensor data”. In: *2012 15th International Conference on Information Fusion*. IEEE. 2012, pp. 1468–1475.

[E11] A. Stathopoulos and M. G. Karlaftis. “A multivariate state space approach for urban traffic flow modeling and prediction”. In: *Transportation Research Part C: Emerging Technologies* 11.2 (2003), pp. 121–135.

[E12] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis. “Short-term traffic forecasting: Overview of objectives and methods”. In: *Transport reviews* 24.5 (2004), pp. 533–557.

[E13] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias. “Short-term traffic forecasting: Where we are and where we’re going”. In: *Transportation Research Part C: Emerging Technologies* 43 (2014), pp. 3–19.

[E14] L. Zhao et al. “T-gcn: A temporal graph convolutional network for traffic prediction”. In: *IEEE Transactions on Intelligent Transportation Systems* (2019).

[E15] Z. Zhao et al. “LSTM network: a deep learning approach for short-term traffic forecast”. In: *IET Intelligent Transport Systems* 11.2 (2017), pp. 68–75.

Smart Public Transportation

[F1] M. Boccia, C. Mannino, and I. Vasilyev. “The dispatching problem on multitrack territories: Heuristic approaches based on mixed integer linear programming”. In: *Networks* 62.4 (2013), pp. 315–326.

[F2] G. Bruno, G. Improta, and A. Sgalambro. “Optimization of the Departure Schedule at a Public Transit Terminal with Multiple Destinations”. In: *Multidisciplinary International Conference on Scheduling: Theory and Application*. 2007, pp. 104–111.

[F3] M. Chen et al. “A dynamic bus-arrival time prediction model based on APC data”. In: *Computer-Aided Civil and Infrastructure Engineering* 19.5 (2004), pp. 364–376.

[F4] F. Corman and L. Meng. “A review of online dynamic models and algorithms for railway traffic management”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.3 (2014), pp. 1274–1284.

[F5] X. J. Eberlein et al. “The real-time deadheading problem in transit operations control”. In: *Transportation Research Part B: Methodological* 32.2 (1998), pp. 77–100.

[F6] L. Fu, Q. Liu, and P. Calamai. “Real-time optimization model for dynamic scheduling of transit operations”. In: *Transportation research record* 1857.1 (2003), pp. 48–55.

[F7] A. Mascis, D. Pacciarelli, and M. Pranzo. “Scheduling models for short-term railway traffic optimisation”. In: *Computer-aided systems in public transport*. Springer, 2008, pp. 71–90.

Shared Mobility, Demand Responsive Transportation and First and Last-mile solutions

[G1] N. Agatz et al. “Optimization for dynamic ride-sharing: A review”. In: *European Journal of Operational Research* 223.2 (2012), pp. 295–303.

[G2] T. D. Chen, K. M. Kockelman, and J. P. Hanna. “Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions”. In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 243–254.

[G3] E. Fatnassi, J. Chaouachi, and W. Klibi. “Planning and operating a shared goods and passengers on-demand rapid transit system for sustainable city-logistics”. In: *Transportation Research Part B: Methodological* 81 (2015), pp. 440–460.

[G4] M. Furuhata et al. “Ridesharing: The state-of-the-art and future directions”. In: *Transportation Research Part B: Methodological* 57 (2013), pp. 28–46.

[G5] H. Hosni, J. Naoum-Sawaya, and H. Artail. “The shared-taxi problem: Formulation and solution methods”. In: *Transportation Research Part B: Methodological* 70 (2014), pp. 303–318.

[G6] B. Li et al. “The share-a-ride problem: People and parcels sharing taxis”. In: *European Journal of Operational Research* 238.1 (2014), pp. 31–40.

[G7] Y. Molenbruch, K. Braekers, and A. Caris. “Typology and literature review for dial-a-ride problems”. In: *Annals of Operations Research* 259.1-2 (2017), pp. 295–325.

[G8] A. Mourad, J. Puchinger, and C. Chu. “A survey of models and algorithms for optimizing shared mobility”. In: *Transportation Research Part B: Methodological* 123 (2019), pp. 323–346.

[G9] A. Sampaio et al. “Crowd-based city logistics”. In: *Sustainable Transportation and Smart Logistics*. Elsevier, 2019, pp. 381–400.

[G10] M. Savelsbergh and T. Van Woensel. “50th anniversary invited article - city logistics: Challenges and opportunities”. In: *Transportation Science* 50.2 (2016), pp. 579–590.

[G11] S. A. Shaheen, N. D. Chan, and T. Gaynor. “Casual carpooling in the San Francisco Bay Area: Understanding user characteristics, behaviors, and motivations”. In: *Transport Policy* 51 (2016), pp. 165–173.

[G12] R. Zhang and M. Pavone. “Control of robotic mobility-on-demand systems: a queueing-theoretical perspective”. In: *The International Journal of Robotics Research* 35.1-3 (2016), pp. 186–203.

Human-machine Interface Algorithms

[H1] M. L. Bolton, E. J. Bass, and R. I. Siminiceanu. “Using formal verification to evaluate human-automation interaction: A review”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43.3 (2013), pp. 488–503.

[H2] O. Carsten and M. H. Martens. “How can humans understand their automated cars? HMI principles, problems and solutions”. In: *Cognition, Technology & Work* 21.1 (2019), pp. 3–20.

[H3] A. Degani and M. Heymann. “Formal verification of human-automation interaction”. In: *Human factors* 44.1 (2002), pp. 28–43.

[H4] A. Degani and M. Heymann. “Pilot-autopilot interaction: A formal perspective”. In: *Abbott et al.[1]* (2000), pp. 157–168.

[H5] A. Gerdes. “Automatic maneuver recognition in the automobile: the fusion of uncertain sensor values using bayesian models”. In: *Proceedings of the 3rd International Workshop on Intelligent Transportation (WIT 2006)*. 2006, pp. 129–133.

[H6] J. H. Hansen et al. “Driver modeling for detection and assessment of driver distraction: Examples from the UTDrive test bed”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 130–142.

[H7] A. Houenou et al. “Vehicle trajectory prediction based on motion model and maneuver recognition”. In: *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2013, pp. 4363–4369.

[H8] T. Hülnhagen et al. “Maneuver recognition using probabilistic finite-state machines and fuzzy logic”. In: *2010 IEEE Intelligent Vehicles Symposium*. IEEE. 2010, pp. 65–70.

[H9] W. Li et al. “Synthesis for human-in-the-loop control systems”. In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2014, pp. 470–484.

[H10] A. G. Mirnig et al. “Control transition interfaces in semiautonomous vehicles: a categorization framework and literature analysis”. In: *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. 2017, pp. 209–220.

[H11] D. Mitrovic. “Reliable method for driving events recognition”. In: *IEEE transactions on intelligent transportation systems* 6.2 (2005), pp. 198–205.

[H12] F. Naujoks et al. “Towards guidelines and verification methods for automated vehicle HMIs”. In: *Transportation research part F: traffic psychology and behaviour* 60 (2019), pp. 121–136.

[H13] J. Nilsson, J. Fredriksson, and E. Coelingh. “Rule-based highway maneuver intention recognition”. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE. 2015, pp. 950–955.

[H14] A. Pnueli. “The temporal logic of programs”. In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. IEEE. 1977, pp. 46–57.

[H15] A. Sathyanarayana, P. Boyraz, and J. H. Hansen. “Driver behavior analysis and route recognition by hidden Markov models”. In: *2008 IEEE International Conference on Vehicular Electronics and Safety*. IEEE. 2008, pp. 276–281.

[H16] Q. Tran and J. Firl. “Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression”. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 918–923.