Delft University of Technology

Delft Center for Systems and Control

Technical report 13-005

Optimal routing for automated highway systems*

L.D. Baskar, B. De Schutter, and H. Hellendoorn

If you want to cite this report, please use the following reference instead:
L.D. Baskar, B. De Schutter, and H. Hellendoorn, "Optimal routing for automated highway systems," *Transportation Research Part C*, vol. 30, pp. 1–22, May 2013. doi:10.1016/j.trc.2013.01.006

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

*This report can also be downloaded via https://pub.deschutter.info/abs/13_005.html

Optimal Routing for Automated Highway Systems

Lakshmi Dhevi Baskar[a,*], Bart De Schutter[a], Hans Hellendoorn[a]

[a] *Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands*

Abstract

We present a routing guidance approach that can be used in Automated Highway Systems (AHS). We consider automated highway systems in which intelligent vehicles organised in platoons drive to their destination, controlled by a hierarchical control framework. In this framework there are roadside controllers that provide speed and lane allocation instructions to the platoons. These roadside controllers typically manage single stretches of highways. A collection of highways is then supervised by so-called area controllers that mainly take care of the route guidance instructions for the platoons and that also coordinate the various roadside controllers in their area. In this paper we focus on the optimal route choice control problem for the area controllers. In general, this problem is a nonlinear integer optimisation problem with high computational requirements, which makes the problem intractable in practice. Therefore, we first propose a simplified but fast simulation model to describe the flows of platoons in the network. Next, we show that the optimal route choice control problem can be approximated by a mixed-integer linear problem. Later, we describe a new METANET-like model to describe the flow of platoons in the AHS. With a simple case study we illustrate that both approaches result in a balanced trade-off between optimality and computational efficiency.

Highlights:
> We consider efficient route guidance control for platoons of autonomous vehicles. > The first control approach we propose is based on mixed-integer linear programming. > The second approach uses real-valued nonlinear programming. > Both provide a balanced trade-off between optimality and computational efficiency.

Keywords:
Automated Highway Systems, Routing, Optimal Control

1. Introduction

Recurring traffic congestion problems and their related costs have resulted in various solution approaches. One of these involves the combination of the

*Corresponding author
Email addresses: lakshmibaskar@gmail.com (Lakshmi Dhevi Baskar), b.deschutter@tudelft.nl (Bart De Schutter), j.hellendoorn@tudelft.nl (Hans Hellendoorn)

existing transportation infrastructure and equipment with advanced technologies from the field of control theory, communication, and information technology. This results in integrated traffic management and control systems, called Automated Highway Systems (AHS), that incorporate intelligence in both the roadside infrastructure and the vehicles. Although this step is considered to be a long-term solution, this approach is capable of offering significant increases in the performance of the traffic system [12, 20, 35].

In the AHS we consider all vehicles are assumed to be fully automated with throttle, braking, and steering commands being determined by automated on-board controllers. Such complete automation of the driving tasks allows to organise the traffic in platoons, i.e., a closely spaced group of vehicles travelling together with short intervehicle distances [34, 38]. Platoons can travel at high speeds and to avoid collisions between platoons at these high speeds, a safe interplatoon distance of about 20–60 m should be maintained. Also, the vehicles in each platoon travel with small intraplatoon distances of about 2–5 m, which are maintained by the automated on-board speed and distance controllers. By travelling at high speeds and by maintaining short intraplatoon distances, the platoon approach allows more vehicles to travel on the network, which improves the traffic throughput [10, 23].

Intelligent Vehicles (IVs) in the AHS can sense the driving environment using sensors and can provide assistance to the driver (via warnings or advisories) or can take complete control of the vehicle itself to achieve an efficient vehicle operation [8, 30, 40]. These vehicle control systems can thus shift driving tasks such as steering, braking, and throttle control from drivers to the on-board controllers in the vehicles. This complete control of driving tasks leads to automated driving, which in turn allows the vehicles' activities to be fully controlled by the traffic control and management systems. Such IVs with complete automation also reduce the negative effects of driver delays and errors.

In [3], we have proposed a hierarchical traffic management and control framework for AHS that builds upon earlier research in this field such as the PATH framework [34]. The proposed architecture consists of a multi-level control structure with local controllers at the lowest level and one or more higher supervisory control levels (see Figure 1). In this framework there are roadside controllers that provide speed and lane allocation instructions to the platoons. These roadside controllers typically manage single stretches of highways. A collection of highways is then supervised by so-called area controllers that mainly take care of the route guidance instructions for the platoons and that also coordinate the various roadside controllers in their area. In this paper, we will concentrate on how the area controllers can determine optimal routes for the platoons using optimal control. We use model-based predictive control to control the traffic system. Therefore, a traffic model is required. In general, traffic flow models can be categorised as microscopic, mesoscopic, or macroscopic models [18].

The choice of the appropriate prediction model depends on the level of detail required and also on the computational requirements. Microscopic traffic flow models become mathematically intractable for large-scale traffic systems due to the high computational requirements. They are in fact more suitable for simulation purposes and for off-line evaluation of control strategies than for on-line model-based control. On the other hand, macroscopic models are less detailed and thus less accurate than microscopic traffic flow models, but due to their fast execution they are well-suited for on-line model-based control.

If we would optimise the routes for each platoon using a microscopic traffic flow model, the optimal route choice problem would lead to mixed-integer nonlinear optimisation over the routes (integer variables) and e.g., speeds (real variables). In general, such optimisation problems are hard to solve and have high computational requirements. Therefore, we will develop a macroscopic traffic model for AHS in this paper.

This paper is an extended and improved version of the conference papers [4, 5]. In particular, a more detailed and improved account of each of the routing approaches is given, the analysis of the macroscopic flow model for intelligent vehicles is expanded, and the approaches is integrated into an overall hierarchical control framework for AHS. The paper is organised as follows. In Section 2 we briefly recapitulate the new hierarchical IV-based traffic control framework of [3]. Next, we focus on the route guidance tasks of the area controllers and we present a simplified flow model and the corresponding optimal route guidance problem in Section 3. We consider both the static case (with constant demands) and the dynamic case (with time-varying demands). In general, the dynamic case leads to a nonlinear non-convex optimisation problem, but we show that this problem can be approximated using mixed-integer linear programming (MILP). We present a simple example that illustrates that the MILP approximation provides a good trade-off between optimality and computational efficiency. In Section 4 we propose an alternative approach based on a macroscopic traffic flow model for human drivers (METANET). We present the basic METANET model for human drivers and we explain how it can be adapted to platoons. Next, we use this model to determine optimal splitting rates at the network nodes. This yields an optimisation problem with real-valued variables only, which thus results in a computation complexity that is much lower than the original mixed-integer optimisation problem. This approach is also illustrated using a case study. Section 6 concludes the paper.

2. Automated highway systems (AHS)

We now briefly present the hierarchical control framework for AHS we have proposed in [3]. This framework is inspired on the PATH platoon concept [19, 38, 39] and it distributes the intelligence between the roadside infrastructure and the vehicles using control measures such as intelligent speed adaption, adaptive cruise control, lane allocation, on-ramp access control, route guidance, etc. to prevent congestion and to improve the performance of the traffic network. The control architecture consists of a multi-level control structure with local controllers at the lowest level and one or more higher supervisory control levels as shown in Figure 1. The layers of the framework can be characterised as follows:

- The *vehicle controllers* present in each vehicle receive commands from the platoon controllers (e.g., set-points or reference trajectories for speeds (for intelligent speed adaption), headways (for adaptive cruise control), and paths) and they translate these commands into control signals for the vehicle actuators such as throttle, braking, and steering actions.

- The *platoon controllers* receive commands from the roadside controllers and are responsible for control and coordination of each vehicle inside the

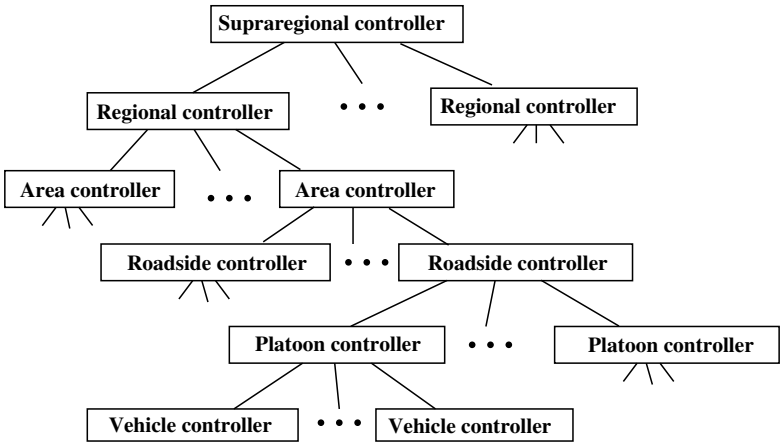


Figure 1: The hierarchical control framework for AHS

platoon. The platoon controllers are mainly concerned with actually executing the interplatoon manoeuvres (such as merges with other platoons, splits, and lane changes) and intraplatoon activities (such as maintaining safe intervehicle distances).

- The *roadside controllers* control a part of a highway or an entire highway. The main tasks of the roadside controllers are to assign speeds for each platoon, safe distances to avoid collisions between platoons, appropriate platoon sizes, and ramp metering values at the on-ramps. The roadside controllers also give instructions for merging, splitting, and lane changes to the platoons.

- The *higher-level controllers* (such as area, regional, and supraregional controllers) provide network-wide coordination of the lower-level and middle-level controllers. In particular, the area controllers provide area-wide dynamic route guidance for the platoons, and they supervise and coordinate the activities of the roadside controllers in their area by providing set-points and control targets. In turn, a group of area controllers could be supervised or controlled by a regional controller, and so on.

The lower levels in this hierarchy deal with faster time scales (typically in the milliseconds range for the vehicle controllers up to the seconds range for the roadside controllers), whereas for the higher-level layers the frequency of updating can range from a few times per minute (for the area controllers) to a few times per hour (for the supraregional controllers).

In this paper, we will focus on the area controllers and in particular on how optimal routes can be determined for the platoons.

3. Optimal route choice control in AHS using mixed-integer linear programming

3.1. Approach

In principle, the optimal route choice control problem in AHS consists in assigning an optimal route to each individual platoon in the network. How-

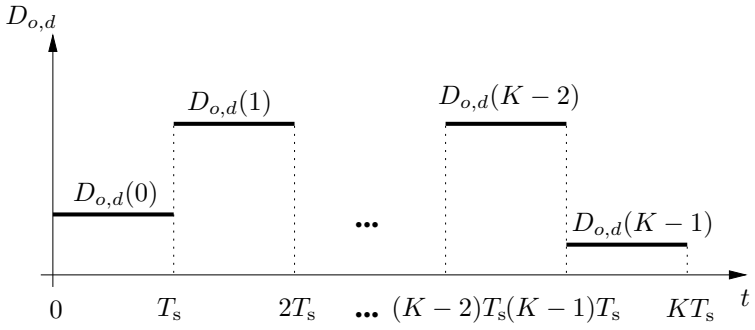


Figure 2: Piecewise constant time-varying demand profile $D_{o,d}$ for the dynamic case

ever, this results in a huge nonlinear integer optimisation problem with a high computational complexity and high computational requirements, making the problem intractable in practice. So, since considering each individual platoon is too computationally intensive, we consider streams of platoons instead (characterised by (real-valued) demands and flows expressed in vehicles per hour). The routing problem will then be recast as the problem of determining the flows on each link as explained later on in this section. For larger networks, we use the spatial division offered by the overall control framework that apply area and regional controllers [6].

Once these flows are determined, they can be implemented by roadside controllers at the links and at the nodes. So the area controllers provide flow targets to the roadside controllers, which then have to control the platoons that are under their supervision in such a way that these targets are met as well as possible. This corresponds to slowing down or speeding up platoons on the links if necessary (in combination with lane allocation and on-ramp access timing), and to steering them into a certain direction depending on the splitting rates for the flows. In case of unpredictable congestion caused by incidents or accidents on the road, the roadside controller will inform the area controller about the congestion in terms of reduced capacities of the links.

3.2. Set-up

We consider the following set-up. The highways in the traffic network are considered to be divided into links. We have a transportation network with a set of origin nodes $\mathcal{O}$, a set of destination nodes $\mathcal{D}$, and a set of internal nodes $\mathcal{I}$. Define the set of all nodes as $\mathcal{V} = \mathcal{O} \cup \mathcal{I} \cup \mathcal{D}$. Nodes can be connected by one or more (unidirectional) links. The links correspond to freeway stretches. The set of all links is denoted by L.

For each origin-destination pair $(o, d) \in \mathcal{O} \times \mathcal{D}$ we define the set $L_{o,d} \subseteq L$ of links that belong to some route going from o to d. For every link $l \in L$ we define the set $\mathcal{S}_{\mathrm{od},l}$ of origin-destination pairs $(o, d) \in \mathcal{O} \times \mathcal{D}$ such that l belongs to some route going from o to d.

For each pair $(o, d) \in \mathcal{O} \times \mathcal{D}$, there is a constant demand $D_{o,d}$ (in the static case) or a dynamic, piecewise constant demand pattern $D_{o,d}(\cdot)$ as shown in Figure 2 with $D_{o,d}(k)$ the demand of vehicles at origin o with destination d in the time interval $[kT_s, (k+1)T_s)$ for $k = 0, \ldots, K-1$ with K the simulation

horizon and T_s the simulation time step (we assume that beyond $T = KT_s$ the demand is 0).

For each link[1] $l \in L$ in the network there is a maximal capacity C_l. We assume that there is a fixed average speed v_l on each link l. Let τ_l denote the average travel time on link l: $\tau_l = \frac{\ell_l}{v_l}$ where ℓ_l is the length of link l. We denote the set of incoming links for node $v \in \mathcal{V}$ by L_v^{in}, and the set of outgoing links by L_v^{out}. Note that for origins $o \in \mathcal{O}$ we have $L_o^{\text{in}} = \emptyset$ and for destinations $d \in \mathcal{D}$ we have $L_d^{\text{out}} = \emptyset$.

The aim is now to assign actual (real-valued) flows $x_{l,o,d}$ (in the static case) or $x_{l,o,d}(k)$ (in the dynamic case[2]) for every pair $(o, d) \in \mathcal{O} \times \mathcal{D}$ and every $l \in L_{o,d}$, in such a way that the given performance criterion (e.g., the total time spent in the network) is minimised subject to operational constraints (e.g., the capacity of the links should not be exceeded).

For the optimal route choice problem we now consider four cases with a gradually increasing complexity:

1. static case with sufficient network capacity,
2. static case with queues at the boundaries of the network only,
3. dynamic case with queues at the boundaries of the network only,
4. dynamic case with queues inside the network.

Note that Cases 2 and 3 with queues at the boundaries of the network can actually occur in practice if the admission and routing control of vehicles is such that the vehicles are only allowed to enter the network if they will later on not encounter any bottleneck[3] on their prescribed route to their destination. In these cases the network itself will always be congestion-free and queues can only arise at the boundaries of the network.

For the dynamic cases (Cases 3 and 4) we will focus on optimal control for the sake of simplicity of the expositions, but the proposed approach can also be included in a model predictive control framework (MPC) [26]. MPC is an on-line, sampling-based, discrete-time receding horizon control approach that uses (numerical) optimisation and an explicit prediction model to determine the optimal values for the control measures over a given prediction period. One of the main advantages of MPC is that it can handle various hard constraints on the inputs and states of the system. In addition, MPC has a built-in feedback mechanism due to the use of a receding horizon approach, and it is easy to tune. We will discuss MPC in more detail in Section 4.4.

3.3. Static case with sufficient network capacity

In this case we assume that there is a constant demand for each origin-destination pair and that the total network capacity is such that the entire demand can be processed, so that there will be no queues at the boundaries or inside the network. Let us now describe the equations to model this situation.

---

[1] This approach can easily be extended to the case where also the internal *nodes* $v \in \mathcal{I}$ have a finite capacity.

[2] More specifically, in the dynamic case $x_{l,o,d}(k)$ denotes the flow of vehicles from origin o to destination d that enter link l in the time interval $[kT_s, (k+1)T_s)$.

[3] Bottlenecks represent the places on network where capacity is restricted, which in their turn may result in queues at or near those locations.

For every origin node $o \in \mathcal{O}$ we have:

$$\sum_{l \in L_o^{\text{out}} \cap L_{o,d}} x_{l,o,d} = D_{o,d} \quad \text{for each } d \in \mathcal{D}. \tag{1}$$

For every internal node $v \in \mathcal{I}$ and for every pair $(o,d) \in \mathcal{O} \times \mathcal{D}$ we have the following conservation relation:

$$\sum_{l \in L_v^{\text{in}} \cap L_{o,d}} x_{l,o,d} = \sum_{l \in L_v^{\text{out}} \cap L_{o,d}} x_{l,o,d} \ . \tag{2}$$

We also have the following condition for every link l:

$$\sum_{(o,d) \in \mathcal{S}_{\text{od},l}} x_{l,o,d} \leqslant C_l \ . \tag{3}$$

Finally, the objective function is given as follows[4]:

$$J_{\text{links}} = \sum_{(o,d) \in \mathcal{O} \times \mathcal{D}} \sum_{l \in L_{o,d}} x_{l,o,d} \tau_l T \ , \tag{4}$$

which is a measure for the total time the vehicles or platoons spend in the network[5]. In order to minimise J_{links} we have to solve the following optimisation problem:

$$\min \ J_{\text{links}} \quad \text{s.t. (1)–(3)} \tag{5}$$

Clearly, this is a linear programming problem. The linear programming problems are efficiently solvable using (a variant of) the simplex method or an interior-point method [32, Chapter 1].

Remark 1. *In the derivation above we keep open the option of including origin-destination dependent weights $w_{o,d}$ into the objective function J_{links} (e.g., to influence long-distance traffic in a different way than local traffic). If such an origin-destination dependent weighting is not required, the setting and the resulting optimisation problem can be simplified by aggregating the flows for a given destination d over all origins, i.e., by considering $x_{l,d} = \sum_{o \in \mathcal{O}} x_{l,o,d}$ instead of $x_{l,o,d}$. A similar remark also holds for the other cases considered in the remainder of the paper.*

3.4. Static case with queues at the boundaries of the network only

In case the capacity of the network is less than the demand, then problem (5) will not be feasible. In order to be able to determine the optimal routing in

[4]Recall that $T = KT_{\text{s}}$ is the length of the simulation period.

[5]Note however that in general, we could also consider other performance criteria such as emissions, noise, fuel consumption, safety, ... Moreover, if we want to put more emphasis on certain routes or certain links, route weight factors $w_{o,d}$ or link weight factors w_l could be added in (4).

this particular case, we have to take into account that queues might appear at the origin of the network.

Let us first write down the equations for the flows inside the network.

For every origin node $o \in \mathcal{O}$ we have:

$$\sum_{l \in L_o^{\text{out}} \cap L_{o,d}} x_{l,o,d} \leqslant D_{o,d} \quad \text{for each } d \in \mathcal{D}. \tag{6}$$

Equations (2) and (3) also hold in this case.

Let us now describe the behaviour of the queues. Since the actual flow out of origin node o for destination d is given by

$$F_{o,d}^{\text{out}} = \sum_{l \in L_o^{\text{out}} \cap L_{o,d}} x_{l,o,d} \ ,$$

the queue length at the origin o for vehicles or platoons going to destination d will increase linearly with a rate $D_{o,d} - F_{o,d}^{\text{out}}$ (note that by (6) this rate is always non-negative). At the end of the simulation period (which has length T) the queue length will be $(D_{o,d} - F_{o,d}^{\text{out}})T$, and hence the average queue length is $\frac{1}{2}(D_{o,d} - F_{o,d}^{\text{out}})T$. So the total time spent in the origin queues is

$$\begin{aligned} J_{\text{queue}} &= \sum_{(o,d) \in \mathcal{O} \times \mathcal{D}} \frac{1}{2}(D_{o,d} - F_{o,d}^{\text{out}})T^2 \\ &= \sum_{(o,d) \in \mathcal{O} \times \mathcal{D}} \frac{1}{2}\left(D_{o,d} - \sum_{l \in L_o^{\text{out}}} x_{l,o,d}\right)T^2 \ . \end{aligned}$$

In order to minimise the total time spent we have to solve the following optimisation problem:

$$\min \ (J_{\text{links}} + J_{\text{queue}}) \quad \text{s.t. (2), (3), and (6).} \tag{7}$$

This is also a linear programming problem.

3.5. Dynamic case with queues at the boundaries of the network only

Now we consider a piecewise constant demand pattern for every origin-destination pair. Moreover, we assume that the travel time τ_l on link l is an integer multiple of T_{s}, say[6]

$$\tau_l = \kappa_l T_{\text{s}} \quad \text{with } \kappa_l \text{ an integer.} \tag{8}$$

Let $q_{o,d}(k)$ denote the partial queue length of vehicles at origin o going to destination d at time instant $t = kT_{\text{s}}$. In principle, the queue lengths should be integers as their unit is "number of vehicles", but we will approximate them using reals.

For every origin node $o \in \mathcal{O}$ we now have:

$$\sum_{l \in L_o^{\text{out}} \cap L_{o,d}} x_{l,o,d}(k) \leqslant D_{o,d}(k) + \frac{q_{o,d}(k)}{T_{\text{s}}} \quad \text{for each } d \in \mathcal{D}, \tag{9}$$

[6]Alternatively we could define the approximate travel delay κ_l on link l as round$(\frac{\tau_l}{T_{\text{s}}})$.

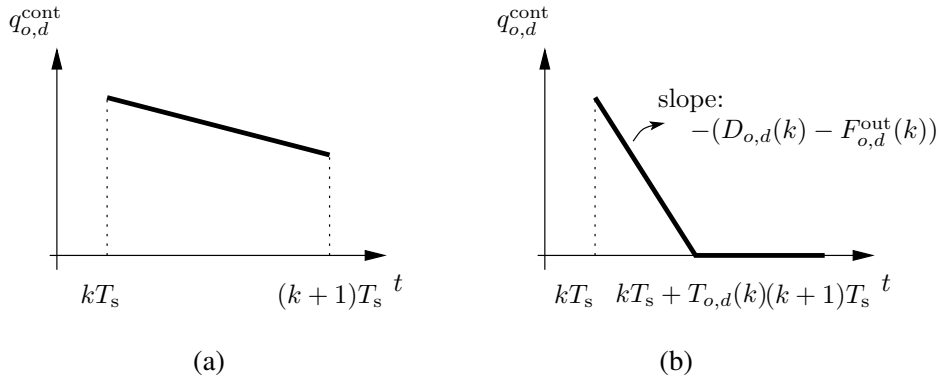


Figure 3: Two possible cases for the evolution of the continuous-time queue length $q_{o,d}^{\text{cont}}$ in the time interval $[kT_{\text{s}}, (k+1)T_{\text{s}})$

with by definition $D_{o,d}(k) = 0$ for $k \geqslant K$, and $q_{o,d}(k) = 0$ for $k \leqslant 0$. Note that the term $\frac{q_{o,d}(k)}{T_{\text{s}}}$ in (9) is due to the assumption that whenever possible and feasible the queue is emptied in the next sample period, with length T_{s}.
Taking into account that every flow on link l has a delay of κ_l time steps before it reaches the end of the link, we have

$$\sum_{l \in L_v^{\text{in}} \cap L_{o,d}} x_{l,o,d}(k - \kappa_l) = \sum_{l \in L_v^{\text{out}} \cap L_{o,d}} x_{l,o,d}(k) \tag{10}$$

for every internal node $v \in \mathcal{I}$ and for every pair $(o,d) \in \mathcal{O} \times \mathcal{D}$, with $x_{l,o,d}(k) = 0$ for $k \leq 0$.
We also have the following condition for every link l:

$$\sum_{(o,d) \in \mathcal{S}_{\text{od},l}} x_{l,o,d}(k) \leqslant C_l \ . \tag{11}$$

Let us now describe the behaviour of the queues. Since the actual flow out of origin node o for destination d in the time interval $[kT_{\text{s}}, (k+1)T_{\text{s}})$ is given by

$$F_{o,d}^{\text{out}}(k) = \sum_{l \in L_o^{\text{out}} \cap L_{o,d}} x_{l,o,d}(k) \ , \tag{12}$$

the queue length at the origin o for vehicles going to destination d will increase linearly with a rate $D_{o,d}(k) - F_{o,d}^{\text{out}}(k)$ in the time interval $[kT_{\text{s}}, (k+1)T_{\text{s}})$. Moreover, queue lengths can never become negative. Hence,

$$q_{o,d}(k+1) = \max\big(0,\, q_{o,d}(k) + (D_{o,d}(k) - F_{o,d}^{\text{out}}(k))T_{\text{s}}\big) \tag{13}$$

In order to determine the time $J_{\text{queue},o,d}(k)$ spent in the queue at origin o in the time interval $[kT_{\text{s}}, (k+1)T_{\text{s}})$ for traffic going to destination d, we have to distinguish between two cases depending on whether or not the continuous-time queue length $q_{o,d}^{\text{cont}}$ becomes equal to zero *inside*[7] the interval $[kT_{\text{s}}, (k+1)T_{\text{s}})$

[7]So we are only in Case (b) if $q_{o,d}^{\text{cont}}$ becomes equal to zero for some time t with $kT_{\text{s}} < t < (k+1)T_{\text{s}}$, i.e., if $q_{o,d}(k) > 0$ and $q_{o,d}(k) + (D_{o,d}(k) - F_{o,d}^{\text{out}}(k))T_{\text{s}} < 0$. All other situations belong to Case (a).

(see Cases (a) and (b) of Figure 3). For Case (b) we define

$$T_{o,d}(k) = \frac{q_{o,d}(k)}{F^{\text{out}}_{o,d}(k) - D_{o,d}(k)} \tag{14}$$

as the time offset after time instant kT_s at which the queue length becomes zero. Then we have

$$J_{\text{queue},o,d}(k) = \begin{cases} \frac{1}{2}(q_{o,d}(k) + q_{o,d}(k+1))T_\text{s} & \text{for Case (a),} \\ \frac{1}{2}q_{o,d}(k)T_{o,d}(k) & \text{for Case (b).} \end{cases}$$

Due to the denominator term in (14) $J_{\text{queue},o,d}(k)$ is in general a nonlinear function. Now assume that we simulate the network until time step $K_\text{end} \geq K$ (e.g., until all queues and all flows have become[8] equal to zero). Then we have

$$J_\text{queue} = \sum_{k=0}^{K_\text{end}-1} \sum_{(o,d)\in\mathcal{O}\times\mathcal{D}} J_{\text{queue},o,d}(k) \ .$$

The time spent in the links is now given by (cf. also (4))

$$J_\text{links} = \sum_{k=0}^{K_\text{end}-1} \sum_{(o,d)\in\mathcal{O}\times\mathcal{D}} \sum_{l\in L_{o,d}} x_{l,o,d}(k)\kappa_l T_\text{s}^2 \tag{15}$$

In order to minimise the total time spent we have to solve the following optimisation problem with J_links still defined by (15):

$$\min \ (J_\text{links} + J_\text{queue}) \qquad \text{s.t. (9)–(13).} \tag{16}$$

Due to the nonlinear expression for $J_{\text{queue},o,d}(k)$ in Case (b) and due to the presence of constraint (13) this is a nonlinear, non-convex, and non-smooth optimisation problem. In general, such problems are difficult to solve and require multi-start local optimisation methods (such as Sequential Quadratic Programming (SQP)) or global optimisation methods (such as genetic algorithms, simulated annealing, or pattern search) [32]. However, in Section 3.7 we will propose an alternative approximate solution approach based on mixed-integer linear programming.

3.6. Dynamic case with queues inside the network

Now we consider the dynamic case with queues inside the network. If queues are formed, we assume that they are formed at the end of the links and that the queues are vertical. In fact, for the sake of simplicity and in order to obtain linear equations, we assign the queues to the nodes instead of the links.

This case is similar to the case of Section 3.5, the difference being that (10) is now replaced by (cf. also (9)):

$$\sum_{l\in L^{\text{out}}_v \cap L_{o,d}} x_{l,o,d}(k) \leqslant \left(\sum_{l\in L^{\text{in}}_v \cap L_{o,d}} x_{l,o,d}(k-\kappa_l) \right) + \frac{q_{v,o,d}(k)}{T_\text{s}} , \tag{17}$$

[8] If this is not the case we have to add an end-point penalty on the queue lengths and flows at time step K_end.

for every internal node $v \in \mathcal{I}$ and for every pair $(o,d) \in \mathcal{O} \times \mathcal{D}$, where $q_{v,o,d}(k)$ is the partial queue length at node v for vehicles or platoons going from origin o to destination d at the time instant $t = kT_\mathrm{s}$. Moreover,

$$q_{v,o,d}(k+1) = \max\big(0, q_{v,o,d}(k) + (F^{\mathrm{in}}_{v,o,d}(k) - F^{\mathrm{out}}_{v,o,d}(k))T_\mathrm{s}\big)$$

with the flow into and out of the queue being given by

$$F^{\mathrm{in}}_{v,o,d}(k) = \sum_{l \in L^{\mathrm{in}}_v \cap L_{o,d}} x_{l,o,d}(k - \kappa_l) \tag{18}$$

$$F^{\mathrm{out}}_{v,o,d}(k) = \sum_{l \in L^{\mathrm{out}}_v \cap L_{o,d}} x_{l,o,d}(k) \ . \tag{19}$$

Similar to $J_{\mathrm{queue},o,d}(k)$ we also define the time $J_{\mathrm{queue},v,o,d}(k)$ spent in the queue at node v in the time interval $[kT_\mathrm{s}, (k+1)T_\mathrm{s})$ for traffic going from origin o to destination d, and we extend the definition of J_{queue} into

$$J_{\mathrm{queue}} = \sum_{k=0}^{K_{\mathrm{end}}-1} \sum_{(o,d)\in\mathcal{O}\times\mathcal{D}} \Big(J_{\mathrm{queue},o,d}(k) + \sum_{v\in\mathcal{I}} J_{\mathrm{queue},v,o,d}(k) \Big) \ .$$

with $J_{\mathrm{queue},v,o,d}(k)$ defined with Case (a) and Case (b) in the same way as in Section 3.5:

$$J_{\mathrm{queue},v,o,d}(k) = \begin{cases} \dfrac{1}{2}\big(q_{v,o,d}(k) + q_{v,o,d}(k+1)\big)T_\mathrm{s} & \text{for Case (a),} \\ \dfrac{1}{2} q_{v,o,d}(k) T_{v,o,d}(k) & \text{for Case (b).} \end{cases}$$

For Case (b) we define

$$T_{v,o,d}(k) = \frac{q_{v,o,d}(k)}{F^{\mathrm{out}}_{v,o,d}(k) - F^{\mathrm{in}}_{v,o,d}(k)} \tag{20}$$

as the time offset after time instant kT_s at which the queue length becomes zero. This also results in a nonlinear, non-convex, and non-smooth optimisation problem. However, in the next section we will show that this problem can also be approximated using mixed-integer linear programming.

Remark 2. *For the dynamic case with queues at the boundaries of the network only, the area controllers will be designed to provide optimal flows in such a way that once the vehicles enter the network, they will travel at their free-flow speed without encountering traffic jams. When the traffic demand exceeds the capacity of the network, queues will be created at one or more origins. Such a congestion-free network comes at the price of increased queue lengths at the origins. However, when the considered network is larger, then it would be a better option for the control system to allow the vehicles to enter the network rather than to create queues at the origin. In larger networks, the vehicles that are allowed to enter the network would be more closer to their destinations than when they would be waiting at origins. In a bigger network, making vehicles wait at the periphery until the entire traffic congestion gets cleared throughout*

the entire network would be highly disadvantageous. On the other hand, allowing vehicles to stay closer to their destination will result in less residual travel time, which in particular is beneficial in case of dynamic traffic demands (with both high and low demands over time) or when different parts of the network get congested while other parts stay uncongested.

Therefore, we have also considered the dynamic case with queues inside the network. Recall that we use vertical queues. There exist two possibilities to represent the creation of queues inside the network. They can be assumed to be created either at the nodes or at the end of the links. In the latter case, the queue at the end of each link will be created by considering its own dynamics and also the behaviour of flows on the adjacent links. However, assuming that queues may arise on every link leads to nonlinear equations and a nonlinear optimisation problem (see, e.g., [37] where a similar routing problem is considered for human drivers). Therefore, we consider the queue formation to occur at the nodes rather than on the links. Queues at the nodes, to a certain extent, can also include spillback effects by using capacity constraints on links. When a link in the network is so full, then the capacity constraint on that link would prevent extra vehicles coming from upstream links to enter the given link. In this way we can model spillback from one link to another

In addition, the model can also include restrictions on the number of vehicles waiting in the queues, by imposing a maximum allowed queue length.

3.7. Approximation based on mixed-integer linear programming

Recall that the dynamic optimal route guidance problems of Sections 3.5 and 3.6 are nonlinear, non-convex, and non-smooth. Now we will show that by introducing an approximation these problems can be transformed into mixed-integer linear programming (MILP) problems, for which efficient solvers have been developed [13].

First we consider the case with queues at the origins only, i.e., we consider the optimisation problem (16). Apart from (13) this problem is a linear optimisation problem.

Now we explain how (13) can be transformed into a system of linear equations by introducing some auxiliary boolean variables δ. To this aim we use the following properties [7], where δ represents a binary-valued scalar variable, y a real-valued scalar variable, and f a function defined on a bounded set X with upper and lower bounds M and m for the function values:

P1: $[f \leqslant 0] \Leftrightarrow [\delta = 1]$ is true if and only if

$$\begin{cases} f \leqslant M(1-\delta) \\ f \geqslant \epsilon + (m-\epsilon)\delta \end{cases},$$

where ϵ is a small positive number[9] (typically the machine precision),

[9]We need this construction to transform a constraint of the form $y > 0$ into $y \geqslant \epsilon$, as in (mixed-integer) linear programming problems only non-strict inequalities are allowed.

P2: $y = \delta f$ is equivalent to

$$\begin{cases} y \leqslant M\delta \\ y \geqslant m\delta \\ y \leqslant f - m(1-\delta) \\ y \geqslant f - M(1-\delta) \ . \end{cases}$$

Depending on the order in which these properties are applied and in which additional auxiliary variables are introduced, we may end up with more or less binary and real variables in the final MILP problem. The number of binary variables — and to a lesser extent the number of real variables — should be kept as small as possible since this number has a direct impact of the computational complexity of the final MILP problem.

To reduce the number of real variables in the final MILP problem, we first eliminate $F^{\text{out}}_{o,d}(k)$ and we write (13) as

$$q_{o,d}(k+1) = \max\Big(0, q_{o,d}(k) + \big(D_{o,d}(k) - \sum_{l \in L^{\text{out}}_o \cap L_{o,d}} x_{l,o,d}(k)\big)T_{\text{s}}\Big) . \qquad (21)$$

Note that this is a nonlinear equation and thus it does not fit the MILP framework. Let $D_{\max,o,d} = \max_k D_{o,d}(k)$ be the maximal demand for origin-destination pair (o,d), let $F_{\max,o,d} = \sum_{l \in L^{\text{out}}_o \cap L_{o,d}} C_l$ be the maximal possible flow out of origin node o towards destination d, and let $q_{\max,o,d} = D_{\max,o,d} T_{\text{s}} K_{\text{end}}$ be the maximal origin queue length at origin o for traffic going to destination d. If we define $m_{o,d} = -F_{\max,o,d} T_{\text{s}}$ and $M_{o,d} = q_{\max,o,d} + D_{\max,o,d} T_{\text{s}}$, then we always have

$$m_{o,d} \leqslant q_{o,d}(k) + \Big(D_{o,d}(k) - \sum_{l \in L^{\text{out}}_o \cap L_{o,d}} x_{l,o,d}(k)\Big)T_{\text{s}} \leqslant M_{o,d} \ .$$

Next, we introduce binary variables $\delta_{o,d}(k)$ such that

$$\delta_{o,d}(k) = 1 \quad \text{if and only if} \quad q_{o,d}(k) + \big(D_{o,d}(k) - \sum_{l \in L^{\text{out}}_o \cap L_{o,d}} x_{l,o,d}(k)\big)T_{\text{s}} \geqslant 0 \ .$$

Using Property **P1** with the bounds $m_{o,d}$ and $M_{o,d}$ this condition can be transformed into a system of linear inequalities. Now we have (cf. (21))

$$q_{o,d}(k+1) = \delta_{o,d}(k)\Big(q_{o,d}(k) + \big(D_{o,d}(k) - \sum_{l \in L^{\text{out}}_o \cap L_{o,d}} x_{l,o,d}(k)\big)T_{\text{s}}\Big) . \qquad (22)$$

This expression is still nonlinear since it contains a multiplication of a binary variable $\delta_{o,d}(k)$ with a real-valued (linear) function. However, by using Property **P2** this equation can be transformed into a system of linear inequalities.
So by introducing some auxiliary variables $\delta_{o,d}(k)$ we can transform the original nonlinear equation (13) into a system of additional linear equations and inequalities.

Recall that $J_{\text{queue},o,d}(k)$ is in general a nonlinear function due to the occurrence of Case (b) of Figure 3. However, if we also use the expression of Case

(a) for Case (b), then we can approximate $J_{\text{queue},o,d}(k)$ as[10]

$$J_{\text{queue},o,d}(k) = \frac{1}{2}(q_{o,d}(k) + q_{o,d}(k+1))T_{\text{s}} \ ,$$

which is a linear expression. Since J_{links} is also a linear expression (cf. (15)), this implies that the overall objective function $J_{\text{links}} + J_{\text{queue}}$ is now linear. So the problem (16) can be approximated by an MILP problem.

Several efficient branch-and-bound MILP solvers [13] are available for MILP problems. Moreover, there exist several commercial and free solvers for MILP problems such as, e.g, CPLEX, Xpress-MP, GLPK, or lp_solve (see [1, 25] for an overview). In principle, — i.e., when the algorithm is not terminated prematurely due to time or memory limitations, — these algorithms guarantee to find the global optimum. This global optimisation feature is not present in the other optimisation methods that can be used to solve the original nonlinear, non-convex, non-smooth optimisation problem (16) such as multi-start SQP, genetic algorithms, pattern search, etc. [32] Moreover, if the allowed computation time is limited (as is often the case in on-line real-time traffic control), then it might occur that the MILP solution can be found within the allotted time whereas a global or multi-start local optimisation algorithm still did not converge to a good solution. As a result, the MILP solution — even though it solves an approximated problem — might even perform better than the solution returned by the prematurely terminated global and multi-start local optimisation method. In general, we can say that the MILP solution often provides a good trade-off between optimality and computational efficiency, as will be illustrated in the case study of Section 3.8.

Using a similar reasoning as above we can also transform the routing problem with queues inside the network of Section 3.6 into an MILP problem. Note however that in this case the number of binary variables may become quite large.

Some ways to reduce the complexity of the above approach are to embed it in a model predictive control framework (cf. Section 4.4) (because then the optimisation horizon is much shorter than the full simulation horizon currently used due to the optimal control setting) or to reduce the number of partial queues by only considering a limited set of most probable/allowed routes between each origin-destination pair. Moreover, if there are no origin-destination dependent performance functions or constraints, we can combine the flows and queues for the various origins at each link and node (see also Remark 1). This will considerably reduce the number of variables.

3.8. Case study

In this section we present a simple case study involving a basic set-up to illustrate the area-level control approach for AHS proposed in Section 2. In particular, we will consider the dynamic case with queues at the origins of the network only (cf. Section 3.5). First, we will describe the set-up and the details of the scenario used for our simulation and next solve the resulting optimisation problem (16). Finally, we will discuss and analyse the obtained results.

[10] This is exact for Case (a) and an approximation for Case (b). However, especially if T_{s} is small enough, the error we then make is negligible.

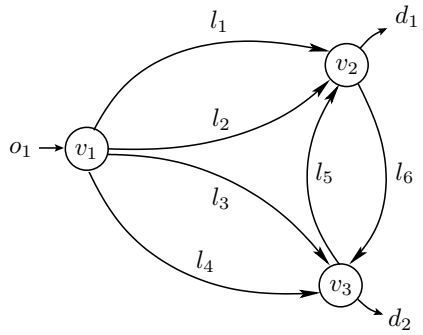


Figure 4: Set-up of case study network of Section 3.8

Period (min)	0–10	10–30	30–40	40–60
D_{o_1,d_1} (veh/h)	5000	8000	2500	0
D_{o_1,d_2} (veh/h)	1000	2000	1000	0

Table 1: Demand profiles used in the case study of Section 3.8

3.8.1. Scenario

We consider a simple network of highways with one origin o_1 and two destinations d_1, d_2, and three internal nodes v_1, v_2, and v_3 (see Figure 4). The network consists of three high-capacity links connecting o_1 to v_1, v_2 to d_1, and v_3 to d_2, as well as six links connecting the internal nodes, allowing four possible routes to each destination (e.g., d_1 can be reached via l_1, l_2, l_3+l_5, and l_4+l_5).

We simulate a period of 60 min. The simulation time step T_s is set to 1 min. The demand pattern is piecewise constant during the simulation period and is given in Table 1. Note that the demand to be processed in the period [10,30] is higher than the capacity of the network, giving rise to an origin queue for each destination. The capacities on the links directly connected to the origin and destination nodes are assumed to be high enough so that no queues are formed on them, and the travel time on these links is assumed to be negligible. The maximum capacities associated with the links between the internal nodes are C_1=1900 veh/h, C_2=2000 veh/h, C_3=1800 veh/h, C_4=1600 veh/h, C_5=1000 veh/h, and C_6=1000 veh/h. Depending on the speed and length of each link, different travel times can be obtained, which are characterised by (cf. (8)) κ_1=10, κ_2=9, κ_3=6, κ_4=7, κ_5=2, and κ_6=2. For the proposed scenario the initial state of the network is taken to be empty.

We consider three different cases:

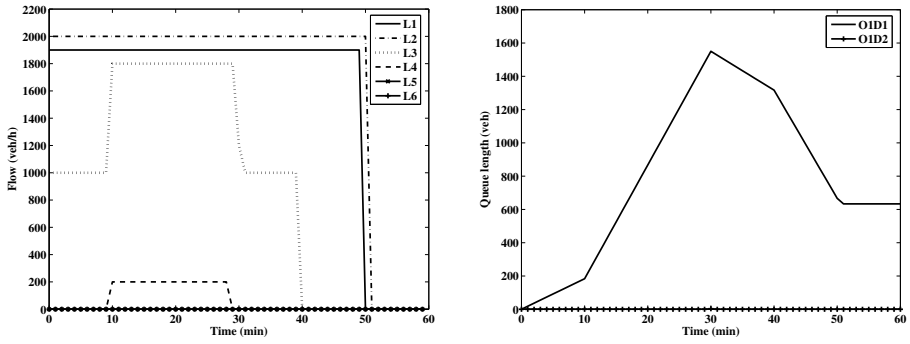
- Case A: no control,
- Case B: controlled using the MILP solution,
- Case C: controlled using the exact solution.

3.8.2. Results and analysis

We have used Matlab to compute the optimal route choice solutions in Cases B and C. More specifically, the MILP problem of Case B has been solved using CPLEX, implemented through the `cplex` interface function of the Matlab

Case	J_{queue} (veh.h)	improvement	CPU time[11] (s)
no control	1434	0 %	–
MILP	1081	24.6 %	0.27
SQP (5 initial points)	1067	25.6 %	90.0
SQP (50 initial points)	1064	25.8 %	983
SQP (with the MILP solution as initial point)	1064	25.8 %	1.29

Table 2: Results for the case study of Section 3.8. The improvement is expressed with respect to the no-control case



(a) Flow for the uncontrolled case.

(b) Queue length for the uncontrolled case.

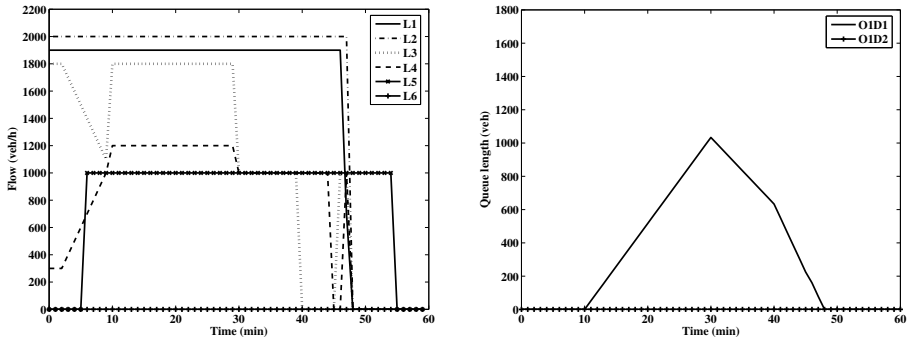
Figure 5: Simulations for the uncontrolled case.

Tomlab toolbox. For Case C we have used the SQP function SNOPT, implemented via the function `snopt` of the Matlab Tomlab toolbox. For Case C we have considered three different choices for the starting points: 5 random initial points, 50 random initial points, and the MILP solution as the initial point. The results of the numerical experiments are listed in Table 2.

In case of no control (Case A), the capacities of the direct links l_1, l_2, l_3, and l_4 are consumed up to their maximum while the links l_5 and l_6 are not used due to the fact that all vehicles and platoons want to take the shortest routes as shown in Fig. 5(a). At the point when the demand exceeds the maximum capacity of the links (i.e., during the period from 10 to 30 minutes, where the total demand 10000 veh/h exceeds the maximum capacity 4000 veh/h) an origin queue is formed as shown in Fig. 5(b). As the simulation advances further, the queue length also increases linearly with time, thus leading to a large total time spent of 1434 veh.h.

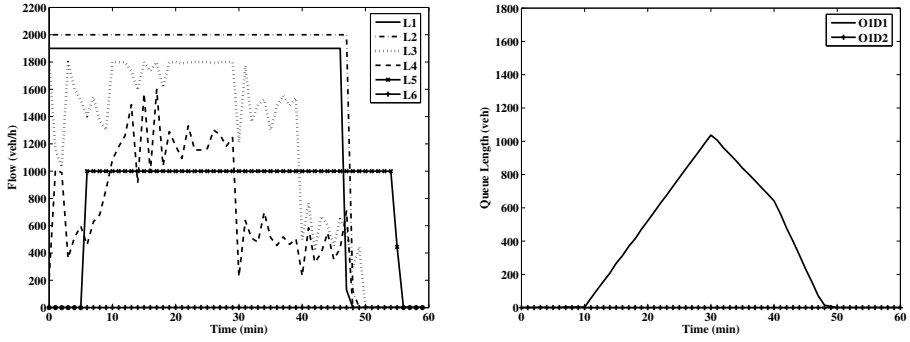
When control is applied, the area controller assigns the routes to the platoons in a system optimum manner. By system optimum, we mean that some of the platoons and vehicles can even be assigned a longer route rather than the direct or shortest routes, if this leads to an improvement of the total traffic performance. This results in a performance improvement of 24.6 % for the MILP solution (Case B), and — depending also on the number of initial points con-

[11] On a 1 GHz Athlon 64 X2 Dual Core 3800+ processor with 3 GB of RAM.



(a) Flow for the controlled case. (b) Queue length for the controlled case.

Figure 6: Simulations for the controlled case using MILP solution.



(a) Flow for the controlled case. (b) Queue length for the controlled case.

Figure 7: Simulations for the controlled case using exact solution.

sidered — in a performance improvement of up to 25.8 % for the exact solution (Case C). For the controlled cases (Case B and C) as shown in Fig. 6(a) and Fig. 7(a), the link l_5 is consumed as this route is considered to be optimum by the controller. The oscillatory behaviour observed in Fig. 7(a) can be removed by adding a penalty on variations in the control signal. Note that for this case study using the MILP solution as the starting point for SQP yields the optimal solution at very low computational costs (1.29 s). In Fig. 6(b) and Fig. 7(b), we can clearly see that the number of vehicles that are queued at the origin are less when compared to the uncontrolled case and the vehicles in the queue reduces as the simulation advances further.

Although the exact solution performs better than the MILP solution, this comes at the cost of an increased computation time due to the multi-start SQP, which results in a total computation time that can be much larger than T_s (1 min). In practice, where the approach will typically be applied on-line in a moving horizon approach, this excessive computation time makes the multi-start SQP approach infeasible, whereas the MILP solution can be computed within the sampling time interval T_s while having almost the same performance as the multi-start SQP solution.

4. Optimal routing for AHS using a macroscopic traffic flow model

In Section 3, we have used a rough approximation of the real network traffic dynamics. An alternative but somewhat more refined way to obtain a simplified model to describe the traffic flows in AHS is to use some of the existing macroscopic traffic flow models for human drivers such as the METANET model and to adapt them to fit the AHS framework.

4.1. Macroscopic traffic flow characteristics for human drivers

Consider a traffic network consisting a several links, each of which is divided in one or more segments. Let T_s be the sampling or simulation time step for the macroscopic network model. The traffic dynamics in each segment i of the traffic network at time instant kT_s can characterised by three macroscopic variables:

- space-mean speed $v_i(k)$,
- traffic density $\rho_i(k)$,
- traffic flow, intensity or volume $q_i(k)$.

The traffic density[12] ρ_i is the number of vehicles per kilometre i.e., it characterises at a specific point of time, how crowded the particular segment i of a road is. In relation to the microscopic traffic variables, the traffic density can be derived using the average distance headway ($\overline{s}_i$), and number of vehicles $N_{\mathrm{veh},i}$ in that segment as

$$\rho_i = \frac{N_{\mathrm{veh},i}}{L_{\mathrm{seg},i}} \approx \frac{N_{\mathrm{veh},i}}{\sum_{l=1}^{N_{\mathrm{veh},i}} s_{i,l}} = \frac{1}{\overline{s}_i} \tag{23}$$

[12]For the sake of simplicity of notation we drop the time index k in the remainder of this section.

where $L_{\text{seg},i}$ is the length of the given road segment and $s_{i,l}$ is the space headway for vehicle l in segment i (i.e., the distance difference in position between the rear of vehicle l and the rear of its predecessor). The average distance headway ($\overline{s}_i$) can be expressed as the product of space mean speed and average time headway. The time headway of a vehicle is defined as the time difference between the passing of the rear ends of the vehicle's predecessor and the vehicle itself at a certain location.

The traffic flow or volume q_i is the number of vehicles $N_{\text{pass},i}$ passing through a freeway location (marked by, e.g., a detector) per time unit. In relation to the microscopic variables, the traffic flow can be defined as a reciprocal of the average time headway ($\overline{h}_i$). Assuming a certain time period ΔT, the flow can expressed as:

$$q_i = \frac{N_{\text{pass},i}}{\Delta T} \approx \frac{N_{\text{pass},i}}{\sum_{l=1}^{N_{\text{pass},l}} h_{i,l}} = \frac{1}{\overline{h}_i} \tag{24}$$

where the time headway $h_{i,l}$ of vehicle l in segment i is the amount of time necessary for the rear of vehicle l to reach the current position of the rear of its predecessor.

The three basic macroscopic variables are related to each other by the fundamental relation

$$q_i = \rho_i v_i \tag{25}$$

This means that out of these three variables only two are independent. In the sequel we will consider v_i and ρ_i to be the independent variables.

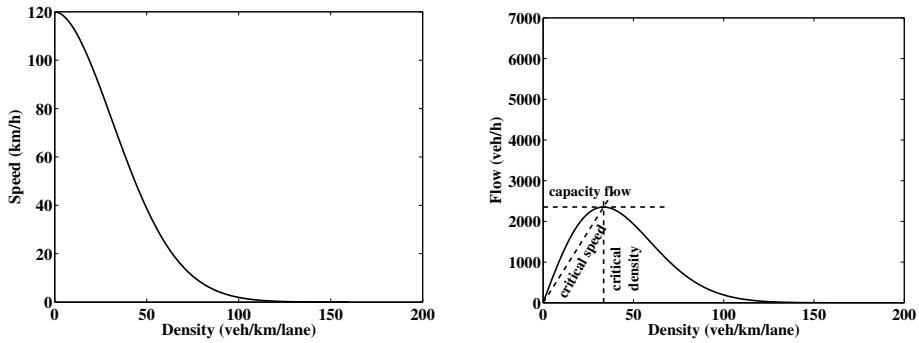
Let us now consider the (equilibrium) relation between the speed v_i and the density ρ_i. When the density on the road is very low and the average distance headway is large, the drivers travel at their desired speed. This is called free-flow driving. As the density starts to increase due to the increasing demand, the vehicles will start to reduce their speed slightly and follow their predecessor while maintaining a safe time headway. Once the critical density (i.e., the density at which the capacity of the network is being utilised at its maximum) is reached, the speed starts to decrease significantly resulting in a traffic jam. When the density is at its maximum ($\rho_{\text{max},i}$), the vehicle speed drops to almost zero. The (equilibrium) relation between the speed v_i and the density ρ_i can be modelled as [27]:

$$V(\rho_i) = v_{\text{free},i} \exp\left[-\frac{1}{a_i}\left(\frac{\rho_i}{\rho_{\text{crit},i}}\right)^{a_i}\right] \tag{26}$$

where $\rho_{\text{crit},i}$ is the critical density, $v_{\text{free},i}$ is the free-flow speed, and a_i is a model parameter. Typical values for these parameters are $v_{\text{free},i}$=120 km/h, $\rho_{\text{crit},i}$ =33.5 veh/km/lane, a_i = 1.867, and $\rho_{\text{max},i}$=180 veh/km/lane [22]. The fundamental relation given in (26) can be depicted using the so-called fundamental diagram as shown in Figure 8 for a single lane. This figure shows the maximum flow $q_{\text{max},i}$, and the critical density $\rho_{\text{crit},i}$.

4.2. Macroscopic traffic flow for IVs

When semi-automatic or intelligent vehicles are used on the road, the macroscopic traffic flow will change. An example of such a change is described by Bose and Ioannou [9], where Adaptive Cruise Control (ACC) is considered with a



(a) Speed-density plot (b) Flow-density plot

Figure 8: Fundamental diagram for human drivers

constant time headway policy. The constant time headway policy is the control form most often used for ACC [9, 36, 41]. The spacing is given in [41] as

$$d_{\mathrm{ref},i}(\ell) = S_0 + v_i(\ell) T_{\mathrm{head},i} \ , \tag{27}$$

where $d_{\mathrm{ref},i}$ is the desired distance headway, S_0 is the gap at zero speed, $T_{\mathrm{head},i}$ is the time headway, and v_i is the velocity of the vehicle of interest (the following vehicle), and L_{veh} is the length of the vehicle, which, for the sake of simplicity, is assumed to be the same for all vehicles (if this is not the case, the average vehicle length $\bar{L}_{\mathrm{veh}}$ should be used in the equations below).

Using (23), for a given speed v_i and (average) space headway $\overline{s}_i$ the maximal density with intelligent vehicles (IVs) can be expressed as the reciprocal of the inter-vehicle spacing:

$$\rho_{\mathrm{ACC},i} = \frac{1}{\overline{s}_i} = \frac{1}{h_{\mathrm{des}} v_i + L_{\mathrm{veh}}} \tag{28}$$

Rewriting (28) gives an expression for the (maximally possible) speed as

$$v_i = \frac{1}{h_{\mathrm{des}}} \left(\frac{1}{\rho_{\mathrm{ACC},i}} - L_{\mathrm{veh}} \right)$$

Now taking into account that the speed cannot exceed the free-flow speed $v_{\mathrm{free},i}$, the expression for the desired speed using 100% ACC-equipped vehicles becomes

$$v_{\mathrm{ACC},i} = \begin{cases} v_{\mathrm{free},i}, & \text{if } \rho_i \leq \rho_{\mathrm{ACC,crit},i} \\ \frac{1}{h_{\mathrm{des}}} \left(\frac{1}{\rho_i} - L_{\mathrm{veh}} \right), & \text{if } \rho_i > \rho_{\mathrm{ACC,crit},i} \end{cases} \tag{29}$$

For a situation with 100% ACC-equipped vehicles the critical density $\rho_{\mathrm{ACC,crit},i}$ at which the maximal flow is obtained, is thus given by:

$$\rho_{\mathrm{ACC,crit},i} = \frac{1}{h_{\mathrm{des}} v_{\mathrm{free},i} + L_{\mathrm{veh}}} \ . \tag{30}$$

Using (29) and (25) the relation between the flow and density becomes

$$q_{\mathrm{ACC},i} = \begin{cases} \rho_i v_{\mathrm{free},i}, & \text{if } \rho_i \leq \rho_{\mathrm{ACC,crit},i} \\ \frac{1}{h_{\mathrm{des}}} \left(1 - \rho_i L_{\mathrm{veh}} \right), & \text{if } \rho_i > \rho_{\mathrm{ACC,crit},i} \end{cases} \tag{31}$$

For typical values of $h_{\text{des}} = 0.5\,\text{s}$, $L_{\text{veh}} = 4\,\text{m}$, and $v_{\text{free},i} = 120\,\text{km/h}$, we obtain $\rho_{\text{ACC,crit}}$i=48.39 veh/km and the speed-density and flow-density curves shown in Figure 9. The flow-density curve illustrates that traffic with IVs will always yield a better performance than that of human drivers, and it also shows that the maximum flow is more than doubled.

Remark 3. *The value of $\rho_{\text{ACC,crit},i}$ in* (30) *was determined without considering the inter-platoon separations. So for platoon-based AHS the capacity flow will be somewhat smaller, also depending on the (average) platoon size.*

Therefore, we will now determine a more refined expression for the critical density $\tilde{\rho}_{\text{platoon,crit},i}$ by considering the inter-platoon separations. We consider a given platoon p and assume for the sake of simplicity that the vehicles in the platoon are numbered 1 (last vehicle), 2 (one but last vehicle), ..., n_p (platoon leader). We will determine the critical density $\tilde{\rho}_{\text{platoon,crit},i}$ by approximating the average space headway $\overline{s}_i$ in segment i in terms of inter-platoon distances and length of platoons. The speed-dependent length $L_{\text{platoon},p}$ of the platoon is given by

$$L_{\text{platoon},p} = (n_p - 1)S_0 + (n_p - 1)h_{\text{des,intra}}v_{n_p} + n_p L_{\text{veh}} \ , \tag{32}$$

where S_0 the minimum safe distance that is to be maintained at zero speed, $h_{\text{des,intra}}$ is the desired time headway for vehicles inside the platoon, v_{n_p} the speed of the platoon (leader), and L_{veh} the (average) length of the vehicles.

The inter-platoon distance $S_{\text{platoon},p}$ for the platoon can be expressed as follows:

$$S_{\text{platoon},p} = S_{\text{inter}} + h_{\text{des,inter}}v_{n_p} \ , \tag{33}$$

where S_{inter} the minimum inter-platoon safe distance that is to be maintained at zero speed, $h_{\text{des,inter}}$ is the desired time headway for platoons, and v_{n_p} the speed of the platoon (leader).

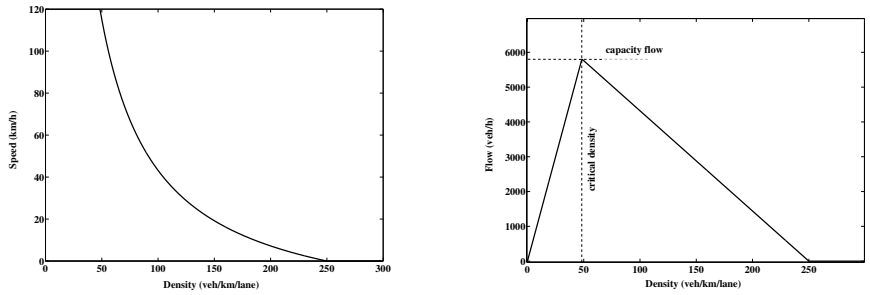
From (32) *and* (33) *it follows that the average space headway $\overline{s}_i$ in segment i is given by:*

$$\begin{aligned}\overline{s}_i &= \frac{L_{\text{platoon},p} + S_{\text{platoon},p}}{n_p} \\ &= \frac{S_{\text{inter}} + h_{\text{des,inter}}v_{n_p} + (n_p - 1)S_0 + (n_p - 1)h_{\text{des,intra}}v_{n_p} + n_p L_{\text{veh}}}{n_p} \ .\end{aligned}$$

Hence, the maximal density $\tilde{\rho}_{\text{platoon,crit},i}$ with platoons of IVs is given by

$$\begin{aligned}\tilde{\rho}_{\text{platoon,crit},i} &= \frac{1}{\overline{s}_i} \\ &= \frac{1}{\left(h_{\text{des,inter}}\frac{1}{n_p} + h_{\text{des,intra}}\frac{n_p - 1}{n_p}\right)v_{n_p} + L_{\text{veh}} + \frac{S_{\text{inter}} - S_0}{n_p} + S_0} \ .\end{aligned} \tag{34}$$

Note the similarity between this equation and (30).



(a) Speed-density plot (b) Flow-density plot

Figure 9: Fundamental diagram for IVs

4.3. A METANET-like macroscopic model for IVs

The METANET model is a second-order macroscopic traffic flow model for human drivers that has been proposed by Papageorgiou and his co-workers [21, 22, 28, 31]. The METANET model is discretised in time and space. As it deals with macroscopic variables rather than the variables or states of individual vehicles, it is suited for on-line computational purposes. In order to make the METANET model suitable for IVs, we will integrate the traffic flow equations for IVs presented in Section 4.2 into the existing METANET model.

The METANET model can be classified as destination-oriented or destination-independent. Since we will use the METANET model for solving routing problems, we will adopt the destination-oriented model, which explicitly models the traffic flows for different destinations. The METANET model consists of link equations and node equations (nodes can represent a junction or a bifurcation point). The link model describes the behaviour of the traffic in the highway stretches, and the node model describes the behaviour of the traffic at the nodes in the network. At each node with two or more outgoing links, the model associates splitting rates to each reachable destination from that node. These splitting rates describe how the traffic flow at the node destined to a particular destination must be distributed among the set of leaving links.

For the simulation of splitting rates with human drivers, the choice of driver's decision is based on the user optimum strategy (determined by selecting the shortest route under free-flow conditions for each user). However, in the case of AHS, the splitting rates can considered to be a controllable input, since here the traffic management system has full control over the IVs. Therefore, in the controlled cases, the splitting rates are determined by MPC-based control approach, which is primarily aimed at providing system optimum solutions (i.e., minimising the total system costs). These splitting rates are then imposed by the traffic management control centres.

4.3.1. Link model

The METANET model represents a network as a directed graph with the links corresponding to freeway stretches as shown in Figure 10. Each freeway link has uniform characteristics, i.e., no on-ramps or off-ramps, and no major changes in geometry. Where major changes occur in the characteristics of the link or in the road geometry (e.g., on-ramp or an off-ramp), a node is placed.

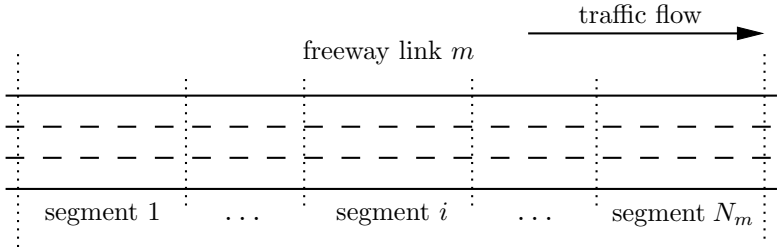


Figure 10: Freeway link in METANET

In the METANET each link m is divided into N_m segments with length L_m. The number of lanes on link m is denoted by λ_m. The traffic flow in segment i of link m destined to destination j at time step k is characterised by three macroscopic variables:

- space-mean speed $v_{m,i}(k)$ [km/h]
- partial density $\rho_{m,i,j}(k)$ [veh/km/lane]
- traffic flow $q_{m,i}(k)$ [veh/h]

where time step k corresponds to time instant $t = kT_\mathrm{s}$ where T_s is the simulation time step (typically around 10 seconds) and where we set $\rho_{m,i,j}(k) = 0$ for all i, k if destination j is not reachable from link m

Note that we use partial densities $\rho_{m,i,j}(k)$ to distinguish between traffic flows with different destinations. However, the same mean speed $v_{m,i}(k)$ is assigned to all the vehicles travelling in a segment irrespective of their destination.

The segment length L_m is usually in the range of 0.5 to 1 km. For stability reasons, a vehicle travelling in a segment at its free speed is not allowed to pass the segment in one simulation time step. So, the following condition should be satisfied:

$$L_m > v_{\mathrm{free},m} T_\mathrm{s} \tag{35}$$

where $v_{\mathrm{free},m}$ is the free-flow speed in link m (a typical value of $v_{\mathrm{free},m}$ is 120 km/h). Note that for the values of L_m, $v_{\mathrm{free},m}$, and T_s given above this condition is satisfied.

For each segment in a link, for all possible destinations reachable via the link, the conservation of vehicles in a segment can be expressed as

$$\rho_{m,i,j}(k+1) = \rho_{m,i,j}(k) + \frac{T}{L_m \lambda_m} \left(\gamma_{m,i-1,j}(k) q_{m,i-1}(k) - \gamma_{m,i,j}(k) q_{m,i}(k) \right) \tag{36}$$

where $\gamma_{m,i,j}(k)$ is the composition rate for the traffic flow in segment i of link m with destination as j at simulation time step k, defined as:

$$\gamma_{m,i,j}(k) = \frac{\rho_{m,i,j}(k)}{\rho_{m,i}(k)} \tag{37}$$

where the total density $\rho_{m,i}(k)$ in segment i of link m is defined as

$$\rho_{m,i}(k) = \sum_{j \in J_m} \rho_{m,i,j}(k)$$

with J_m the set of destinations reachable via link m.

The update of the mean speed is calculated based on a convection term, a relaxation term, and an anticipation term. More specifically, the mean speed in segment i of link m at the next time step $k+1$ is given by

$$v_{m,i}(k+1) = v_{m,i}(k) + v_{\text{free},m} \exp\left[-\frac{1}{a_m}\left(\frac{\rho_{m,i}(k)}{\rho_{\text{crit},m}}\right)^{a_m}\right] + \frac{T_\text{s}}{L_m} v_{m,i}(k)\left(v_{m,i-1}(k) - v_{m,i}(k)\right) - \frac{\eta T_\text{s}}{\tau L_m}\frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} \tag{38}$$

where τ, κ, and η are model parameters and where $V(\rho_{m,i}(k)$ can be derived either for human drivers or for IVs. For human drivers a typical value for τ is 18 s [22]. For IVs this value will be much lower, e.g., 8 s. Typical values for κ and η are κ=40 veh/km/lane and η=60 km^2/h [22]. For human drivers, $V(\rho_{m,i}(k)$ is given by (cf. (26))

$$V(\rho_{m,i}(k)) = v_{\text{free},m} \exp\left[-\frac{1}{a_m}\left(\frac{\rho_{m,i}(k)}{\rho_{\text{crit},m}}\right)^{a_m}\right] \ . \tag{39}$$

The expression of $V(\rho_{m,i}(k)$ for IVs is given by (cf. (29)):

$$V(\rho_{m,i}(k)) = \begin{cases} v_\text{free}, & \text{if } \rho_{m,i}(k) \le \rho_{\text{ACC,crit},m} \\ \frac{1}{h_\text{des}}\left(\frac{1}{\rho_{m,i}(k)} - L_\text{veh}\right), & \text{if } \rho_{m,i}(k) > \rho_{\text{ACC,crit},m} \ . \end{cases} \tag{40}$$

The partial traffic flow for each segment can be described as follows:

$$q_{m,i,j}(k) = \rho_{m,i,j}(k) v_{m,i}(k) \lambda_m \tag{41}$$

4.3.2. Origin model

Origins receive traffic demand and forward it to the freeway. Origins are modelled using a simple queue model. The queue length $w_{o,j}(k+1)$ destined to destination j at origin o can be determined from the previous queue length and the total demand $d_o(k)$ at time step k as follows:

$$w_{o,j}(k+1) = w_{o,j}(k) + T_\text{s}\gamma_{o,j}(k)\left(d_o(k) - q_o(k)\right) \tag{42}$$

with $\gamma_{o,j}(k)$ is the fraction of the demand travelling to destination j from origin o. The outflow or service rate at origin $q_o(k)$ can be expressed as:

$$q_o(k) = \min\left[d_o(k) + \frac{w_o(k)}{T_\text{s}}, \quad Q_{\text{cap},o} \min\left(1, \frac{\rho_\text{max} - \rho_{\mu,1}(k)}{\rho_\text{max} - \rho_{\text{crit},\mu}}\right)\right] \tag{43}$$

where $Q_{\text{cap},o}$ is the capacity (veh/h) of the origin o under free-flow conditions, ρ_max is the maximum density of a segment, and μ is the index of the link to which the origin is connected.

4.3.3. Node model

The node model describes how the traffic should be routed among the set of entering and leaving links of a node. For a given node n, let I_n denote the set of input links, and let O_n denote the set of output links. The traffic flow $Q_{n,j}(k)$

with destination j that enters the node n at simulation step k is distributed to the output links according to

$$Q_{n,j}(k) = \sum_{\mu \in I_n} q_{\mu,N_\mu}(k)\gamma_{\mu,N_\mu,j}(k) \tag{44}$$

$$q_{n,m,\text{out}}(k) = \sum_{j \in J_m} \beta_{n,m,j}(k)Q_{n,j}(k) \tag{45}$$

where $q_{\mu,N_\mu}(k)$ is the flow leaving the last segment of link μ, $\beta_{n,m,j}(k)$ is the splitting rate in node n that is defined as the fraction of the traffic flow heading towards destination j that leaves node n via output link m, and $q_{n,m,\text{out}}(k)$ is the total traffic flow that leaves node n via output link m at simulation step k.

The composition rate $\gamma_{n,m,\text{out},j}(k)$ of the traffic flow out of node n into link m is given by:

$$\gamma_{n,m,\text{out},j}(k) = \frac{\beta_{n,m,j}(k)Q_{n,j}(k)}{q_{m,\text{out}}(k)} \tag{46}$$

4.3.4. Downstream density

Consider a node n with input link $m \in I_n$. Note that the anticipation term of the speed update equation (38) for segment i of link m contains the downstream density $\rho_{m,i+1}(k)$. Hence, we also need an expression for the downstream density for the last segment (segment N_m) of link m. To this aim we introduce a virtual segment N_m+1 at the end of link m and we capture the effect of the downstream density of the output links leaving node n by the following expression:

$$\rho_{m,N_{m+1}}(k) = \frac{\sum_{\mu \in O_n} \rho_{\mu,1}^2(k)}{\sum_{\mu \in O_n} \rho_{\mu,1}(k)} \tag{47}$$

where $\rho_{\mu,1}(k)$ is the density of the first segment of output link μ.

4.3.5. Upstream speed

Similarly as above, when a node n has two or more input links (junction node), then the downstream speed for the first segment $i = 1$ of an outgoing link required in the convection term of (38) is captured by adding a virtual segment at the beginning of the link and by setting

$$v_{m,0}(k) = \frac{\sum_{\mu \in I_n} v_{\mu,N_\mu}(k)q_{\mu,N_\mu}(k)}{\sum_{\mu \in I_n} q_{\mu,N_\mu}(k)} \tag{48}$$

where N_μ is the index of the last segment of link μ.

4.3.6. Extensions

Hegyi et al. [16, 17] have extended the METANET model to include traffic control measures such as dynamic speed limits, and mainstream metering, to account for anticipation behaviour of drivers to varying downstream densities, and to model main-stream origins. These extensions can also be integrated in the proposed METANET-like traffic flow model for AHS.

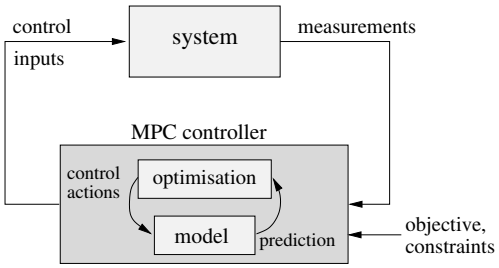


Figure 11: Schematic view of the MPC structure.

4.4. Model predictive route choice control

Now we use the model of the previous subsection to derive a model-based predictive control approach that can be used by the area controllers to determine the optimal splitting rates.

More specifically, we adopt the model predictive control (MPC) scheme [11, 26, 29, 33] (see Figure 11). In the MPC control scheme a discrete-time model is used to predict the future behaviour of the process, and the MPC controller uses (numerical) optimisation to determine the control signals that result in an optimal process behaviour over a given prediction horizon. The resulting optimal control inputs are applied using a rolling horizon scheme. Let T_c be the control sample time, i.e., the time interval between two updates of the control signal settings. For the sake of simplicity we will assume that T_c is an integer multiple of T_s, i.e., $T_\mathrm{c} = MT_\mathrm{s}$ for some integer M. A typical value for T_c is 1 min. At each control step ℓ the state of the traffic system is measured or estimated, and an optimisation is performed over the prediction horizon $[\ell T_\mathrm{c}, (\ell + N_\mathrm{p})T_\mathrm{c}]$ to determine the optimal control inputs, where N_p is the prediction horizon. Only the first value of the resulting control signal (the control signal for control time step ℓ) is then applied to the process. At the next control step $\ell + 1$ this procedure is repeated.

To reduce the computational complexity and to improve stability often a control horizon N_c ($\leq N_\mathrm{p}$) is introduced in MPC, and after the control horizon has been passed the control signal is taken to be constant.

There are two loops involved in MPC: the rolling horizon loop and the optimisation loop inside the controller. The optimisation loop inside the controller of Figure 11 is executed as many times as needed to find the optimal control signals at control step ℓ, for the given N_p, N_c, traffic state, and expected demands. The loop connecting the controller and the traffic system is performed once for each control step ℓ and provides the state feedback to the controller. This feedback is necessary to correct for (the ever present) prediction errors, and to provide disturbance rejection (i.e., compensation for unexpected traffic demand variations). The advantage of this rolling horizon approach is that it results in an on-line adaptive control scheme that also allows us to take gradual changes in the system or in the system parameters into account by regularly updating the model of the system.

For our case the control variables in this set-up are the splitting rates β^{ctrl} at the nodes with more than one outgoing link. The optimisation signals include the control variables as well as the state variables of the macroscopic METANET-like traffic flow model for AHS derived above.

One can also include various constraints such as maximum flows on certain links, maximal speeds, maximal travel times for selected origin-destination pairs, etc. Moreover, there is a physical constraint that for a given node the sum of the splitting rates should be equal to 1:

$$\sum_{m \in O_n} \sum_{j \in J_m} \beta^{\text{ctrl}}_{n,m,j}(\ell) = 1 \tag{49}$$

for all n, k.

A typical objective function to be used is the total time spent (TTS) by all the vehicles in the network[13]. This includes both the time spent travelling through the network and the time spent waiting in the queues, if any. Since $T_\text{c} = MT_\text{s}$ with M an integer, the time instant $t = \ell T_\text{c}$ with ℓ the control step can also be expressed as $t = M\ell T_\text{s}$. So at control step ℓ we get the following expression for the total time spent in the period $[\ell T_\text{c}, (\ell + N_\text{p})T_\text{c}]$:

$$J_\text{TTS}(\ell) = \sum_{l=M\ell}^{M(\ell+N_\text{p})-1} \Bigg(\sum_{(m,i) \in \mathcal{L}_\text{ls}} \rho_{m,i}(l) L_m \lambda_m + \sum_{(o,j) \in \mathcal{O}_\text{od}} w_{o,j}(l) \Bigg) T_\text{s} \tag{50}$$

where $\mathcal{L}_\text{ls}$ is the set of all link-segment index pairs (m, i) of the network, and $\mathcal{O}_\text{od}$ the set of all origin-destination pairs (o, j). Note that in (50) the expression between the brackets gives the total number of vehicles present in the links and queues of the network. This value is multiplied by the simulation time step T_s and summed to obtain the total time spent.

The control splitting rates β^ctrl (expressed as a function of the control step ℓ) are mapped to the splitting rates β of the simulation model (expressed as a function of the simulation step k) using a zero-order hold strategy, i.e.,

$$\beta_{n,m,j}(M\ell + l) = \beta^{\text{ctrl}}_{n,m,j}(\ell) \tag{51}$$

for l=0,. . . , $M - 1$.

Minimising J_TTS then results in a nonlinear non-convex optimisation problem with real-valued variables. In general this is an NP-hard problem [15] just like the original route choice problem. In fact with (50), we now have an optimisation problem with real-valued variables, which offers a major advantage compared to the mixed-integer optimisation of the original route choice problem of Section 3.7, since the real-valued optimisation will require a lower computational effort to find (sub)optimal solutions.

To solve the nonlinear optimisation problem we can use a global or a multi-start local optimisation method such as multi-start sequential quadratic programming, pattern search, genetic algorithms, or simulated annealing [32].

The optimisation problem (cf. (49) – (51)) is a nonlinear, non-convex optimisation problem, which in general is NP-hard [15]. Therefore, the computational effort and the CPU time required for solving the problem will increase tremendously for realistic traffic scenarios. In such cases, we can use distributed controllers and high speed parallel processors to tackle the complexity of the problem. For large networks, we can use area and regional controllers as described in our paper [6], where regional controllers provide control inputs (using

[13]Note however that other (multi-objective) performance criteria can also be included easily.

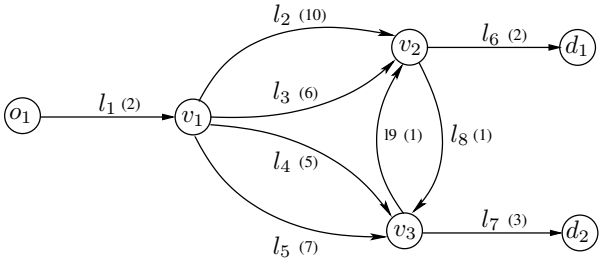


Figure 12: Set-up of case study network of Section 4.5

Period (min)	0–10	10–30	30–40	40–60
D_{o_1,d_1} (veh/h)	5000	8000	2500	0
D_{o_1,d_2} (veh/h)	1000	2000	1000	0

Table 3: Demand profiles used in the case study of Section 4.5

MPC) on links to area controllers. For sub-networks, fast MPC methods can be developed as shown in paper [24], where fast MPC is effectively applied for traffic sub-networks with up to 17 nodes and 46 links. Our proposed approach should be able to tackle the networks of a similar size. Fast MPC approaches for other application are described in [7, 14].

4.5. Case study

Now we present a simple case study involving a basic set-up to illustrate the area-level control approach for AHS proposed in this section. First, we describe the set-up and the details of the scenario used for our simulations. Next, we discuss and analyse the obtained results.

4.5.1. Scenario

We consider a simple network of highways with one origin o_1 and two destinations d_1, d_2, and three internal nodes v_1, v_2, and v_3 (see Figure 12). Note that this example is similar to the one of Section 3.8. However, due to the different type of model used, the way the case study is specified will differ.

The network of Figure 12 consists of three links[14] connecting o_1 to v_1, v_2 to d_1, and v_3 to d_2, as well as six links connecting the internal nodes allowing four possible routes to each destination (e.g., d_1 can be reached via l_2, l_3, l_4+l_9, and l_5+l_9). In Figure 12, the value within brackets indicates the number of segments (N_m) in that particular link. The length of a segment (L_m) in any link is taken to be 1 km.

We consider three different cases (due to the use of two fundamental diagrams):

- Case A: no control case with human drivers,
- Case B: controlled case with humans drivers,

[14] In contrast to the example of Section 3.8 we now explicitly consider these origin and destination links in order to account for the effect of the boundary conditions.

- Case C: controlled case with platoons.

We use the following values for the parameters of the METANET(-like) model for all the links (see also [22] and the previous subsections): v_{free}=120 km/h, $a = 1.867$, κ=40 veh/km/lane and η=60 km^2/h. For the case of human drivers we use ρ_{crit}=33.5 veh/km/lane, τ=18 s, and the fundamental V–ρ relation (39), while for the IV case we use ρ_{crit} =48.39 veh/km/lane, κ=40 veh/km/lane and η=60 km^2/h, τ=8 s [15], and the fundamental V–ρ relation (40). We have S_{inter} = 20 m, S_0 = 3 m, $h_{\text{des,intra}}$ = 0.5 s, $h_{\text{des,inter}}$ = 1.2 s for all vehicles [16].

We simulate a period of 60 min. The simulation time step T_{s} is set to 20 s and the control sample step T_{c} is set to 20 s as well. The demand pattern is piecewise constant during the simulation period and is given in Table 3. The demand to be processed in the period [10,30] will be higher than the capacity of the network, giving rise to an origin queue for each destination. For the proposed scenario the initial state of the network is taken to be empty. We choose $N_{\text{p}} = 20$ and $N_{\text{c}} = 6$. For the sake of simplicity we take the simulation model to be equal to the prediction model.

4.5.2. Control problem

The control variables considered for this case study are the splitting rates $\beta^{\text{ctrl}}_{n,m,j}(\ell)$ associated with all reachable destinations via outgoing links for each internal node for $\ell = 0, 1, \ldots, N_{\text{sim}} - 1$ where $N_{\text{sim}} = 180$ is the total number of simulation or control steps (of length $T_{\text{s}} = 20$ s) within the entire simulation period of 60 min.

Since it makes no sense to send vehicles reaching node v_2 and going to destination 1, towards link l_8 we set $\beta^{\text{ctrl}}_{v_2,l_6,1}(\ell) = 1$ and $\beta^{\text{ctrl}}_{v_2,l_8,1}(\ell) = 0$ for all ℓ. Likewise, we set $\beta^{\text{ctrl}}_{v_2,l_6,2}(\ell) = 0$ and $\beta^{\text{ctrl}}_{v_2,l_8,2}(\ell) = 1$ for all ℓ. For node v_3 we have similar expressions: $\beta^{\text{ctrl}}_{v_3,l_7,1}(\ell) = 0$ and $\beta^{\text{ctrl}}_{v_3,l_9,1}(\ell) = 1$, $\beta^{\text{ctrl}}_{v_3,l_7,2}(\ell) = 1$, $\beta^{\text{ctrl}}_{v_3,l_9,2}(\ell) = 0$ for all ℓ. So in fact the actual optimisation variables are $\beta^{\text{ctrl}}_{v_1,m,j}(\ell)$ for $m = l_2, l_3, l_4, l_5$ and $j = 1, 2$.

We have the following constraints (cf. (49)):

$$\beta^{\text{ctrl}}_{v_1,l_2,j}(\ell) + \beta^{\text{ctrl}}_{v_1,l_3,j}(\ell) + \beta^{\text{ctrl}}_{v_1,l_4,j}(\ell) + \beta^{\text{ctrl}}_{v_1,l_5,j}(\ell) = 1$$

for $j = 1, 2$ and for all ℓ.

The goal of our area controller is to improve the traffic performance. The objective that we consider for our case study is minimisation of the total time spent (TTS) by all the vehicles in the network using routing as the control measure. The TTS for the entire simulation period can be expressed as (cf. (50)):

$$J_{\text{TTS,sim}} = \sum_{k=1}^{N_{\text{sim}}-1} \left(\sum_{(m,i)\in\mathcal{L}_{\text{ls}}} \rho_{m,i}(k) L_m \lambda_m + \sum_{(o,j)\in\mathcal{O}_{\text{od}}} w_{o,j}(k) \right) T_{\text{s}} \qquad (52)$$

where $\mathcal{L}_{\text{ls}}$ is the set of all link-segment index pairs (m, i) of the network, and $\mathcal{O}_{\text{od}}$ the set of all origin-destination pairs (o, j).

[15] Determining appropriate values for τ for IVs is indeed an important topic for our future research.

[16] Please note that our framework assumes that the vehicles in the platoons are equipped with V2V and CACC.

Case	$J_{\mathrm{TTS,sim}}$ (veh.h)	improvement (%)
no control	686.28	0 %
controlled human	670.91	3 %
controlled IVs	366.07	46 %

Table 4: Results for the case study of Section 4.5. The improvement is expressed with respect to the no-control case with human drivers.

4.5.3. Results and analysis

The results of the numerical experiments are listed in Table 4.

In case of no control (Case A), the capacities of the direct links l_1, l_2, l_3, and l_4 are consumed up to their maximum while the links l_8 and l_9 are not used due to the fact that all vehicles and platoons want to take the shortest routes. At the point when the demand exceeds the maximum capacity of the links, an origin queue is formed. As the simulation advances, the queue length increases with time, thus leading to a huge total time spent.

For the controlled cases (Cases B and C) the area controller assigns the splitting rates at the internal node v_1 and routes the traffic flow (human drivers or platoons) in a system-optimum manner such that the traffic performance is improved. When platoons of IVs are deployed in the traffic system, the traffic performance is improved more than the human drivers case. For these cases we have used the SQP function SNOPT, implemented via the function `snopt` of the Matlab Tomlab toolbox and applied in a multi-start mode with 20 different random initial starting points to compute the optimal splitting rates. Compared to Case A this results in a performance improvement of about 3 % for Case B and of about 46 % for Case C. In Table 4[17], the main improvement in the controlled approach is obtained from automated platooning (along with advanced intelligent vehicle techniques such as CACC, ISA, dynamic route guidance system, etc.).

5. Interface between area and roadside controllers

The area controllers will provide flow targets to the roadside controllers (e.g., using the approaches proposed in Sections 3 and 4), which then have to control the platoons that are under their supervision in such a way that these targets are met as well as possible. In practice, we could implement the control using dynamic tolling systems, interaction with on-board route guidance devices, etc. In our AHS framework, the area and roadside controllers use a different level of representation of the traffic. However, consistency between the macroscopic traffic characteristics (such as flows, splitting rates) and the platoons has to be established to ensure efficiency of the approach. Hence, our approach requires an interface between these controllers, which is implemented as follows.

Once the optimal target flows are determined by the area controller, the roadside controller has to realise the target flows on each link as well as possible by using the available control measures such as speed limits, ramp metering,

[17] On a 1 GHz Athlon 64 X2 Dual Core 3800+ processor with 3 GB of RAM.

route guidance, and lane allocation control. (see paper [2]). In order to achieve the specified optimal flows on the links, the roadside controller will use MPC and a performance criterion based on the minimisation of the difference between the reference flows and the actual flows. The roadside controller is aware of the complete information of the IVs (i.e., position, speed, lane, origin, destination) and hence can determine the actual flow $x_{l,o,d}(k)$ on each link l for all $(o,d) \in \mathcal{O} \times \mathcal{D}$. At each control step ℓ, the roadside controller will be updated with this flow information. Let $T_{\rm rd}$ be the control sample time for the roadside controller. For the sake of simplicity we will assume that $T_{\rm rd}$ is an integer multiple of $T_{\rm s}$ (and an integer divisor of the area control sample time $T_{\rm c}$). Let $T_{\rm rd} = PT_{\rm s}$ for some integer P. The performance function $J_{\rm perf}$ used by the roadside controller at control step ℓ is then given by

$$J_{\rm perf}(\ell) = \sum_{k=\ell P}^{(\ell+N_{\rm p})P-1} \sum_{(o,d)\in\mathcal{O}\times\mathcal{D}} \sum_{l\in L_{o,d}} (x^{\rm opt}_{l,o,d}(k) - x_{l,o,d}(k))^2 \qquad (53)$$

where $x^{\rm opt}_{l,o,d}(k)$ is the optimal flow — as specified by the area controller — of vehicles from origin o to destination d that enter link l in the time interval $[kT_{\rm s}, (k+1)T_{\rm s})$.

6. Summary

We have considered the optimal route guidance problem for AHS. Since the resulting optimisation problem is a nonlinear mixed-integer optimisation problem that in general is too involved for on-line, real-time implementation, we have explored approximations resulting in simplified but fast simulation models that yield mixed-integer *linear* or nonlinear *real-valued* optimisation problems, for both of which efficient solvers exist.

The first approach we have proposed is based on a simplified model that describes the movement of the platoons in the network via flows. We have shown that using this model the optimal route choice control problem can be approximated by a linear or a mixed-integer linear problem. With a case study we have illustrated that the resulting approach can offer a balanced trade-off between computational efficiency and optimality.

In the second approach we have developed a new model to describe the flow of platoons in AHS based on the macroscopic METANET traffic flow model, which has been adapted to fit the case of platoons of intelligent vehicles equipped with ACC. This model has subsequently been used in a model predictive control approach for determining optimal splitting rates of the platoon flows at the nodes in the network. This approach has also been illustrated via a simple case study.

Since our framework is a generic approach, we can also use MPC for MILP routing methods and use optimal control for the METANET case.

The proposed approach allow area controllers in a hierarchical traffic management system for AHS to determine optimal flows and splitting rates. The lower-level roadside controllers can then translate these flows and splitting rates into actual route instructions for the platoons.

In our future research, we will extensively compare the two approaches (using a microscopic model as simulation model), assess the effects of routing in addition to platooning, consider more extensive case studies, include and explore

the platoon formation mechanism, carry out an in-depth analysis of circumstances under which Remark 2 (about allowing vehicles closer to their destination than at the periphery) applies, and assess the performance improvement of the proposed approaches with respect to an approach based on mixed-integer optimisation for the original route choice problem. We will also investigate the coordination and mutual interaction between various area controllers and between the area controllers and the roadside controllers. In addition, we will perform simulations for different sets of parameter values for the platoons(for both CACC and regular ACC), investigate the scalability of our approach, and assess the size of the networks our approach can be applied to in practice.

Acknowledgements

Research supported by the European 7th framework STREP project "Hierarchical and distributed model predictive control of large-scale systems (HD-MPC)", contract number INFSO-ICT-223854, the European COST Action TU-0702, the European 7th Framework Network of Excellence "Highly-complex and networked control systems (HYCON2)", the BSIK project "Next Generation Infrastructures (NGI)", the Transport Research Centre Delft, and the Delft Research Center Next Generation Infrastructures.

References

[1] Atamtürk, A., Savelsbergh, M. W. P., Nov. 2005. Integer-programming software systems. Annals of Operations Research 140 (1), 67–124.

[2] Baskar, L., De Schutter, B., Hellendoorn, H., Jun. 2012. Traffic management for automated highway systems using model-based predictive control. IEEE Transactions on Intelligent Transportation Systems 13 (2), 838–847.

[3] Baskar, L. D., De Schutter, B., Hellendoorn, H., Jun. 2007. Hierarchical traffic control and management with intelligent vehicles. In: Proceedings of the 2007 IEEE Intelligent Vehicles Symposium (IV'07). Istanbul, Turkey, pp. 834–839.

[4] Baskar, L. D., De Schutter, B., Hellendoorn, H., Sep. 2009. Optimal routing for intelligent vehicle highway systems using mixed integer linear programming. In: Proceedings of the 12th IFAC Symposium on Transportation Systems. Redondo Beach, California, pp. 569–575.

[5] Baskar, L. D., De Schutter, B., Hellendoorn, J., Oct. 2009. Optimal routing for intelligent vehicle highway systems using a macroscopic traffic flow model. In: Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC 2009). St. Louis, Missouri, pp. 576–581.

[6] Baskar, L. D., De Schutter, B., Hellendoorn, J., Sep. 2010. Hierarchical model-based predictive control for intelligent vehicle highway systems: Regional controllers. In: Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010). Madeira Island, Portugal, pp. 249–254.

[7] Bemporad, A., Morari, M., Mar. 1999. Control of systems integrating logic, dynamics, and constraints. Automatica 35 (3), 407–427.

[8] Bishop, R., 2005. Intelligent Vehicles Technology and Trends. Artech House.

[9] Bose, A., Ioannou, P., 2003. Mixed manual/semi-automated traffic: a macroscopic analysis. Transportation Research Part C 11 (6), 439–462.

[10] Broucke, M., Varaiya, P., Nov. 1997. The automated highway system: A transportation technology for the 21st century. Control Engineering Practice 5 (11), 1583–1590.

[11] Camacho, E. F., Bordons, C., 1995. Model Predictive Control in the Process Industry. Springer-Verlag, Berlin, Germany.

[12] Fenton, R. E., Dec. 1994. IVHS/AHS: Driving into the future. IEEE Control Systems Magazine 14 (6), 13–20.

[13] Fletcher, R., Leyffer, S., May 1998. Numerical experience with lower bounds for MIQP branch-and-bound. SIAM Journal on Optimization 8 (2), 604–616.

[14] García, C., Prett, D., Morari, M., May 1989. Model predictive control: Theory and practice — A survey. Automatica 25 (3), 335–348.

[15] Garey, M. R., Johnson, D. S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, San Francisco, California.

[16] Hegyi, A., De Schutter, B., Hellendoorn, J., Jun. 2005. Model predictive control for optimal coordination of ramp metering and variable speed limits. Transportation Research Part C 13 (3), 185–209.

[17] Hegyi, A., De Schutter, B., Hellendoorn, J., Mar. 2005. Optimal coordination of variable speed limits to suppress shock waves. IEEE Transactions on Intelligent Transportation Systems 6 (1), 102–112.

[18] Hoogendoorn, S. P., Bovy, P. H. L., Aug. 2001. State-of-the-art of vehicular traffic flow modelling. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 215 (4), 283–303.

[19] Horowitz, R., Varaiya, P., Jul. 2000. Control design of an automated highway system. Proceedings of the IEEE: Special Issue on Hybrid Systems 88 (7), 913–925.

[20] Jurgen, R. K., May 1991. Smart cars and highways go global. IEEE Spectrum 28 (5), 26–36.

[21] Kotsialos, A., Papageorgiou, M., Diakaki, C., Pavlis, Y., Middelham, F., Dec. 2002. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. IEEE Transactions on Intelligent Transportation Systems 3 (4), 282–292.

[22] Kotsialos, A., Papageorgiou, M., Messmer, A., Jul. 1999. Optimal coordinated and integrated motorway network traffic control. In: Proceedings of the 14th International Symposium of Transportation and Traffic Theory (ISTTT). Jerusalem, Israel, pp. 621–644.

[23] Li, K., Ioannou, P., Jun. 2004. Modeling of traffic flow of automated vehicles. IEEE Transactions on Intelligent Transportation Systems 5 (2), 99–113.

[24] Lin, S., De Schutter, B., Xi, Y., Hellendoorn, H., Oct. 2012. Efficient network-wide model-based predictive control for urban traffic networks. Transportation Research Part C 24, 122–140.

[25] Linderoth, J., Ralphs, T., Jan. 2005. Noncommercial software for mixed-integer linear programming. Optimization Online.

[26] Maciejowski, J. M., 2002. Predictive Control with Constraints. Prentice-Hall, Harlow, England.

[27] May, A. D., 1990. Traffic Flow Fundamentals. Prentice-Hall, Englewood Cliffs, New Jersey.

[28] Messmer, A., Papageorgiou, M., 1990. METANET: A macroscopic simulation program for motorway networks. Traffic Engineering and Control 31 (9), 466–470.

[29] Messmer, A., Papageorgiou, M., mar 1995. Route diversion control in motorway networks via nonlinear optimization. ieeecs 3 (1), 144–154.

[30] Nookala, M., Estochen, B., 2002. Minnesota, USA intelligent vehicle initiative. In: Proceedings of the IEEE Intelligent Vehicles Symposium. Versailles, France, pp. 533–536.

[31] Papageorgiou, M., Blosseville, J. M., Hadj-Salem, H., Sep. 1990. Modelling and real-time control of traffic flow on the southern part of Boulevard Périphérique in Paris: Part I: Modelling. Transportation Research Part A 24 (5), 345–359.

[32] Pardalos, P. M., Resende, M. G. C., 2002. Handbook of Applied Optimization. Oxford University Press, Oxford, England.

[33] Rawlings, J. B., Mayne, D. Q., 2009. Model Predictive Control: Theory and Design. Nob Hill Publishing, Madison, Wisconsin.

[34] Shladover, S., Desoer, C. A., Hedrick, J. K., Tomizuka, M., Walrand, J., Zhang, W. B., McMahon, D. H., Peng, H., Sheikholeslam, S., McKeown, N., Feb. 1991. Automatic vehicle control developments in the PATH program. IEEE Transactions on Vehicle Technology 40 (1), 114–130.

[35] Sussman, J. M., Jun. 1993. Intelligent vehicle highway systems: Challenge for the future. IEEE Micro 1 (14-18), 101–104.

[36] Swaroop, D., Rajagopal, K. R., 1999. Intelligent cruise control systems and traffic flow stability. Transportation Research Part C 7 (6), 329–352.

[37] van den Berg, M., De Schutter, B., Hellendoorn, H., Hegyi, A., Sep. 2009. Control of day-to-day route choice in traffic networks with overlapping routes. In: Proceedings of the 12th IFAC Symposium on Control in Transportation Systems. Redondo Beach, California, pp. 556–561.

[38] Varaiya, P., Feb. 1993. Smart cars on smart roads: Problems of control. IEEE Transactions on Automatic Control 38 (2), 195–207.

[39] Varaiya, P., Shladover, S. E., Oct. 1991. Sketch of an IVHS systems architecture. In: Vehicle Navigation and Information Systems. Dearborn, Michigan, pp. 909–922.

[40] Verburg, D., van der Knaap, A., Ploeg, J., 2002. VEHIL: Developing and testing intelligent vehicles. In: Proceedings of the IEEE Intelligent Vehicles Symposium. Versailles, France, pp. 537–544.

[41] Yi, J., Horowitz, R., Apr. 2006. Macroscopic traffic flow propagation stability for adaptive cruise controlled vehicles. Transportation Research Part C 14 (2), 81–95.