Delft University of Technology

Delft Center for Systems and Control

Technical report 09-029

Optimal routing for intelligent vehicle highway systems using mixed integer linear programming*

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 intelligent vehicle highway systems using mixed integer linear programming," *Proceedings of the 12th IFAC Symposium on Transportation Systems*, Redondo Beach, California, pp. 569–575, Sept. 2009. doi:10.3182/20090902-3-US-2007.0065

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

Optimal Routing for Intelligent Vehicle Highway Systems Using Mixed Integer Linear Programming

Lakshmi Dhevi Baskar * **Bart De Schutter** ** **Hans Hellendoorn** *

* *Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands (email: {l.d.baskar,j.hellendoorn}@tudelft.nl)*
** *Delft Center for Systems and Control & Marine and Transport Technology department, Delft University of Technology, Delft, The Netherlands (email: b@deschutter.info)*

Abstract: We present a routing guidance approach that can be used in Intelligent Vehicle Highway Systems (IVHS). We consider IVHS consisting of automated highway systems on which intelligent vehicles organized 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 optimization 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 linear or a mixed integer linear problem. With a simple case study we illustrate that this results in a balanced trade-off between optimality and computational efficiency.

Keywords: Intelligent-Vehicle Highway Systems, Routing, Optimal Control.

1. INTRODUCTION

The recurring traffic congestion problems and their related costs have resulted in various solution approaches. One of these involves the combination of the 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 Intelligent Vehicle Highway Systems (IVHS), that incorporate intelligence in both the roadside infrastructure and in 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 (Sussman, 1993; Jurgen, 1991; Fenton, 1994).

In IVHS 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 organize the traffic in platoons, i.e., a closely spaced group of vehicles traveling together with short intervehicle distances (Varaiya, 1993; Shladover et al., 1991). 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 traveling 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 (Broucke and Varaiya, 1997; Li and Ioannou, 2004).

In (Baskar et al., 2007) we have proposed a hierarchical traffic management and control framework for IVHS that builds upon earlier research in this field such as the PATH framework (Shladover et al., 1991). The control architecture of Baskar et al. (2007) consists of a multi-level control structure with local controllers at the lowest level and one or more higher supervisory control levels (see also Figure 1). In this paper, we will in particular concentrate on how the area controllers can determine optimal routes for the platoons using optimal control.

The paper is organized as follows. In Section 2 we briefly recapitulate the hierarchical traffic management and control framework of Baskar et al. (2007). 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 (constant demands) and the dynamic case (time-varying demands). In general, the dynamic case leads to a nonlinear nonconvex optimization problem, but in Section 4 we show that this problem can be approximated using mixed integer linear programming (MILP). In Section 5 we present a simple example that illustrates that the MILP approximation provides a good trade-off between optimality and computational efficiency. Section 6 concludes the paper.

2. INTELLIGENT VEHICLE HIGHWAY SYSTEMS (IVHS)

We now briefly present the hierarchical control framework for IVHS we have proposed in (Baskar et al., 2007). This framework is based on the platoon concept and it distributes the intelligence between the roadside infrastructure and the vehicles using control measures such as intelligent speed adaption,

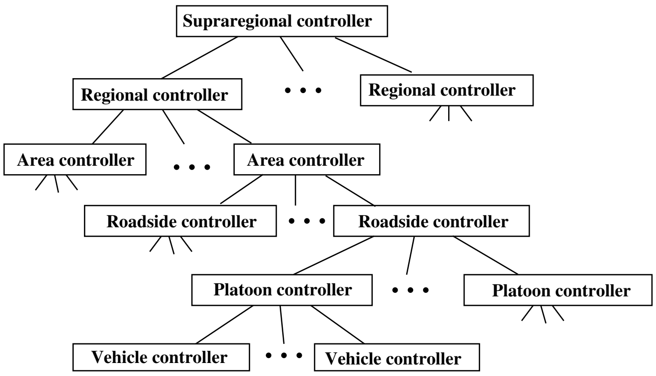

Fig. 1. The hierarchical control framework of Baskar et al. (2007) for IVHS.

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 of Baskar et al. (2007) 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 characterized 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 platoon. The platoon controllers are mainly concerned with actually executing the interplatoon maneuvers (such as merges with other platoons, splits, and lane changes) and intraplatoon activities (such as maintaining safe intervehicle distances).
- The *roadside controllers* may 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 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 few times per minute (for the area controllers) to a few times per hour (for the supraregional controllers).

In (Baskar et al., 2008a,b, 2009) we have proposed model predictive control methods for the roadside controllers to determine optimal speeds, lane allocations, and on-ramp release times for the platoons. In the remainder of the paper we will focus on the area controllers and in particular on how optimal routes can be determined for the platoons.

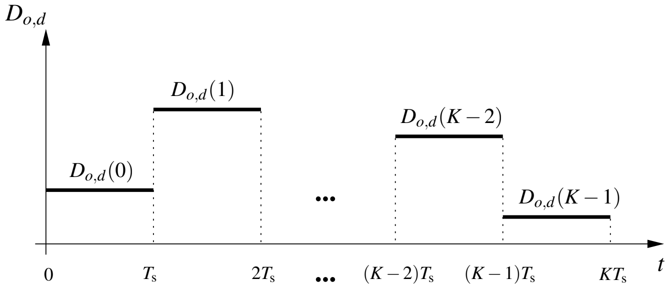

Fig. 2. Piecewise constant time-varying demand profile $D_{o,d}$ for the dynamic case.

3. OPTIMAL ROUTE CHOICE CONTROL IN IVHS

3.1 Approach

In principle, the optimal route choice control problem in IVHS consists in assigning an optimal route to each individual platoon in the network. However, this results in a huge nonlinear integer optimization problem with high computational complexity and requirements, making the problem intractable in practice. So, since considering each individual platoon is too computationally intensive, we will consider streams of platoons instead (characterized by (real-valued) demands and flows expressed in vehicles per hour). The routing problem will be recast as the problem of determining the flows on each link.

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 in the links if necessary (in combination with lane allocation and on-ramp access timing), and to steering them in a certain direction depending on the splitting rates for the flows.

3.2 Set-up

We consider the following set-up. We have a transportation network with a set of origin nodes $\mathscr{O}$, a set of destination nodes $\mathscr{D}$, and a set of internal nodes $\mathscr{I}$. Define the set of all nodes as $\mathscr{V} = \mathscr{O} \cup \mathscr{I} \cup \mathscr{D}$. Nodes can be connected by one or more (unidirectional) links. The set of all links is denoted by L.

For each origin-destination pair $(o,d) \in \mathscr{O} \times \mathscr{D}$ 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 $\mathscr{S}_{\mathrm{od},l}$ of origin-destination pairs $(o,d) \in \mathscr{O} \times \mathscr{D}$ such that l belongs to some route going from o to d.

For each pair $(o,d) \in \mathscr{O} \times \mathscr{D}$, 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 $k = 0, \dots, K-1$ 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 $l \in L$ in the network [1] 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 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 $v \in \mathscr{V}$ by L_v^{in}, and the set of outgoing links by L_v^{out}. Note that for origins $o \in \mathscr{O}$ we have $L_o^{\text{in}} = \emptyset$ and for destinations $d \in \mathscr{D}$ 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) for every pair $(o,d) \in \mathscr{O} \times \mathscr{D}$ and every $l \in L_{o,d}$, in such a way that the capacity of the links is not exceeded and such that the given performance criterion (e.g., total time spent) is minimized. 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 $[kT_\text{s}, (k+1)T_\text{s})$.

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

- Static case with sufficient network capacity,
- Static case with queues at the boundaries of the network only,
- Dynamic case with queues at the boundaries of the network only,
- Dynamic case with queues inside the network.

3.3 Static case with sufficient network capacity

Here 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.

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

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

For every internal node $v \in \mathscr{I}$ and for every pair $(o,d) \in \mathscr{O} \times \mathscr{D}$ we have

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

We also have the following condition for every link l:

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

Finally, the objective function is given as follows [2]:

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

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

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

Clearly, this is a linear programming problem.

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

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

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 this 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 $o \in \mathscr{O}$ we have:

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

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

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

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

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 nonnegative). 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

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

In order to minimize the total time spent we have to solve the following optimization 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

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

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 the sake of simplicity we also assume that initially the network is empty (i.e., $q_{o,d}(k) = 0$ and $x_{l,o,d}(k) = 0$ for $k \leqslant 0$).

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

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

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.

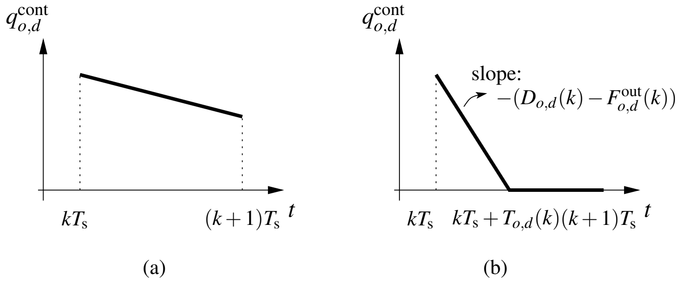

Fig. 3. Two possible cases for the evolution of the (continuous-time) queue length $q_{o,d}^{\mathrm{cont}}$ in the time interval $[kT_\mathrm{s},(k+1)T_\mathrm{s})$.

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

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

for every internal node $v\in\mathscr{I}$ and for every pair $(o,d)\in\mathscr{O}\times\mathscr{D}$, with $x_{l,o,d}(k)=0$ for $k\leq 0$.

We also have the following condition for every link l:

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

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

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

the queue length at the origin o for vehicles going to destination d will increase linearly with a rate [3] $D_{o,d}(k)-F_{o,d}^{\mathrm{out}}(k)$ in the time interval $[kT_\mathrm{s},(k+1)T_\mathrm{s})$. Hence,

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

In order to determine the time $J_{\mathrm{queue},o,d}(k)$ spent in the queue at origin o in the time interval $[kT_\mathrm{s},(k+1)T_\mathrm{s})$ for traffic going to destination d, we have to distinguish between two cases depending on whether or not the continuous-time queue length $q_{o,d}^{\mathrm{cont}}$ becomes equal to zero *inside* [4] the interval $[kT_\mathrm{s},(k+1)T_\mathrm{s}]$ (see Cases (a) and (b) of Figure 3). For Case (b) we define

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

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

$$J_{\mathrm{queue},o,d}(k)=\begin{cases}\frac{1}{2}(q_{o,d}(k)+q_{o,d}(k+1))T_\mathrm{s} & \text{for Case (a),}\\ \frac{1}{2}q_{o,d}(k)T_{o,d}(k) & \text{for Case (b).}\end{cases} \tag{15}$$

Due to the denominator term in (14) $J_{\mathrm{queue},o,d}(k)$ is in general a nonlinear function. Now assume that we simulate the network until time step $K_\mathrm{end}\geqslant K$ (e.g., until all queues and all flows have become [5] equal to zero). Then we have

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

The time spent in the links is now given by

$$J_{\mathrm{links}}=\sum_{k=0}^{K_\mathrm{end}-1}\sum_{(o,d)\in\mathscr{O}\times\mathscr{D}}\sum_{l\in L_{o,d}} x_{l,o,d}(k)\kappa_l T_\mathrm{s}^2 \ . \tag{16}$$

In order to minimize the total time spent we have to solve the following optimization problem:

$$\min\ \big(J_{\mathrm{links}}+J_{\mathrm{queue}}\big) \qquad \text{s.t. (9)–(13).} \tag{17}$$

Due to the presence of constraint (13) and the nonlinear expression for $J_{\mathrm{queue},o,d}(k)$ in Case (b) this is a nonlinear, nonconvex, and nonsmooth optimization problem. In general, these problems are difficult to solve and require multi-start local optimization methods (such as Sequential Quadratic Programming (SQP)) or global optimization methods (such as genetic algorithms, simulated annealing, or pattern search) (Pardalos and Resende, 2002). However, in Section 4 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 case with queues inside the network. If there are queues 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 previous case, the difference being that (10) is now replaced by (cf. also (9)):

$$\sum_{l\in L_v^{\mathrm{out}}\cap L_{o,d}} x_{l,o,d}(k)\leqslant\left(\sum_{l\in L_v^{\mathrm{in}}\cap L_{o,d}} x_{l,o,d}(k-\tau_l)\right)+\frac{q_{v,o,d}(k)}{T_\mathrm{s}}, \tag{18}$$

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,

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

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

$$F_{v,o,d}^{\mathrm{in}}(k)=\sum_{l\in L_v^{\mathrm{in}}\cap L_{o,d}} x_{l,o,d}(k-\tau_l) \tag{19}$$

$$F_{v,o,d}^{\mathrm{out}}(k)=\sum_{l\in L_v^{\mathrm{out}}\cap L_{o,d}} x_{l,o,d}(k) \ . \tag{20}$$

Similar to the way $J_{\mathrm{queue},o,d}(k)$ has been defined in (15) 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

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

In order to minimize the total time spent we have to solve the following optimization problem:

$$\min\ \big(J_{\mathrm{links}}+J_{\mathrm{queue}}\big) \qquad \text{s.t. (9), (11)–(13), and (18)–(20),}$$

with J_{links} still defined by (16). This also results in a nonlinear, nonconvex, and nonsmooth optimization problem. However, in the next section we will show that this problem can also be approximated using mixed integer linear programming.

---

[3] In contrast to Section 3.4 this rate can now also be negative.

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

[5] 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.

4. APPROXIMATION BASED ON MIXED INTEGER LINEAR PROGRAMMING

Recall that the dynamic optimal route guidance problems of Sections 3.5 and 3.6 are nonlinear, nonconvex, and nonsmooth. 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 (Fletcher and Leyffer, 1998).

First we consider the case with queues at the origins only, i.e., we consider the optimization problem (17). Apart from (13) this problem is a linear optimization problem.

Now we explain how we can transform (13) into a system of linear equations by introducing some auxiliary boolean variables δ. To this aim we use the following properties (Bemporad and Morari, 1999), 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: $[f \leqslant 0] \iff [\delta = 1]$ is true if and only if

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

where ε is a small positive number [6] (typically the machine precision),

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

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

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 $m^{\text{low}}_{o,d} = -F_{\max,o,d}T_\text{s}$ and $m^{\text{upp}}_{o,d} = q_{\max,o,d} + D_{\max,o,d}T_\text{s}$, then we always have

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

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

$\delta_{o,d}(k) = 1$ if and only if

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

Using Property **P1** with the bounds $m^{\text{low}}_{o,d}$ and $m^{\text{upp}}_{o,d}$ this condition can be transformed into a system of linear inequalities. Now we have (cf. (21))

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

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 [7]

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

which is a linear expression. This implies that the overall objective function $J_\text{links} + J_\text{queue}$ is now linear. So the problem (17) can be approximated by an MILP problem.

Several efficient branch-and-bound MILP solvers (Fletcher and Leyffer, 1998) 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 (Atamtürk and Savelsbergh, 2005; Linderoth and Ralphs, 2005) 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 optimization feature is not present in the other optimization methods that can be used to solve the original nonlinear, nonconvex, nonsmooth optimization problem (17). Moreover, if the computation time is limited (as is often the case in online real-time traffic control), then it might occur that the MILP solution can be found within the allotted time whereas the global and multi-start local optimization algorithm still did not converge to a good solution. As a result, the MILP solution — even although it solves an approximated problem — might even perform better than the solution returned by the prematurely terminated global and multi-start local optimization 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 5.

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.

5. CASE STUDY

In this section we present a simple case study involving a basic set-up to illustrate the area-level control approach for

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

[7] 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.

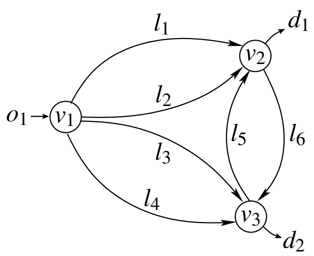

Fig. 4. Set-up of case study network.

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.

IVHS proposed in this paper. In particular, we will consider the dynamic case with queues at the origins of the network only (cf. Section 3.5) and thus solve problem (17). First, we will describe the set-up and the details of the scenario used for our simulations. Next, we will discuss and analyze the obtained results.

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 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. 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 characterized 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.

5.2 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 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.

Case	J_{queue} (veh.h)	improvement	CPU time [8] (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 MILP solution as initial point)	1064	25.8 %	1.29

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

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. At the point when the demand exceeds the maximum capacity of the links, origin queues are formed. 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 considered — in a performance improvement of up to 25.8 % for the exact solution (Case C).

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).

Although the exact solution will in general perform 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 (which is equal to 1 min here). 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.

6. CONCLUSIONS

We have considered the optimal route guidance problem for IVHS. In particular, we have proposed an optimal route guidance approach for platoons by an area controller based on a simplified flow model. Since the resulting optimization problem could still be too involved for on-line, real-time implementation in the case of dynamic demands, we have explored an approximation resulting in a mixed integer linear programming problem, for which efficient solvers exist. With a case study we have illustrated that the resulting approach can offer a balanced trade-off between computational efficiency and optimality.

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

In our future research, we will investigate other methods and approximations (such a receding horizon control, blocking, selection of routes from a restricted set, etc.) to get an even better balance between optimality and computational efficiency. We will also consider additional case studies as well as the coordination and mutual interaction between various area controllers and between the area controllers and the roadside controllers.

ACKNOWLEDGEMENTS

Research supported by the VIDI project "Multi-Agent Control of Large-Scale Hybrid Systems" (DWV.6188) of the Dutch Technology Foundation STW, Applied Science division of NWO and the Technology Programme of the Dutch Ministry of Economic Affairs, the European 7th framework STREP project "Hierarchical and distributed model predictive control (HD-MPC)" (contract number INFSO-ICT-223854), the European COST Action TU0702, the BSIK projects "Next Generation Infrastructures (NGI)" and "Transition to Sustainable Mobility (TRANSUMO)", the Delft Research Center Next Generation Infrastructures, and the Transport Research Centre Delft,

REFERENCES

A. Atamtürk and M.W.P. Savelsbergh. Integer-programming software systems. *Annals of Operations Research*, 140(1): 67–124, November 2005.

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

L.D. Baskar, B. De Schutter, and H. Hellendoorn. Model-based predictive traffic control for intelligent vehicles: Dynamic speed limits and dynamic lane allocation. In *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium (IV'08)*, pages 174–179, Eindhoven, The Netherlands, June 2008a.

L.D. Baskar, B. De Schutter, and H. Hellendoorn. Dynamic speed limits and on-ramp metering for IVHS using model predictive control. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC 2008)*, pages 821–826, Beijing, China, October 2008b. ISBN 978-1-4244-2122-1.

L.D. Baskar, B. De Schutter, J. Hellendoorn, and A. Tarău. Traffic management for intelligent vehicle highway systems using model-based predictive control. In *Proceedings of the 88th Annual Meeting of the Transportation Research Board*, Washington, DC, January 2009. Paper 09-2107.

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

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

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

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

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

K. Li and P. Ioannou. Modeling of traffic flow of automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 5(2):99–113, June 2004.

J. Linderoth and T. Ralphs. Noncommercial software for mixed-integer linear programming. *Optimization Online*, January 2005.

P.M. Pardalos and M.G.C. Resende, editors. *Handbook of Applied Optimization*. Oxford University Press, Oxford, UK, 2002. ISBN 0-19-512594-0.

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

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

P. Varaiya. Smart cars on smart roads. *IEEE Transactions on Automatic Control*, 38(2):195–207, February 1993.