Service level-oriented route guidance during evacuations

R. Landmana,*, A. Hegyi*, S. Hoogendoorna

*aDelft University of Technology, CitG, Transport & Planning, Stevinweg 1, 2628CN, Delft, the Netherlands

Abstract

Research in the field of evacuation management puts a strong focus on the development of optimal evacuation and control strategies. However, in operational evacuation management the means to monitor, manage, evaluate and control are limited. The authorities nevertheless have a strong need to understand the system, in order to properly guide evacuating traffic. This indicates the need for a simple, yet efficient control approach to operationalize the evacuation objectives based on real-time available data. This paper presents a predictive route guidance approach that is able to do this by degrading and restoring target service levels of evacuation routes according to the prevailing negative effects of queue spill back. The control approach consists of a finite-state machine that determines target service levels based on predicted traffic conditions. These target service levels are used in a feedback controller as setpoints, resulting in the corresponding output signal of a Variable Message Sign. By means of a test case, the finite-state machine is compared with model predictive control-based route guidance (that realizes system optimal conditions), with a user equilibrium feedback controller (that realizes user optimal conditions) and with a do-nothing scenario. The results show that the finite-state machine is able to prevent or limit the negative effects of blocking back in a comprehensible and efficient way, while the interests of the road users are also taken into account.

© 2011 Published by [PUBLISHER CO NAME HERE] Ltd. Selection and/or peer-review under responsibility of [name organizer]

Keywords: evacuation management, service levels; dynamic route guidance; finite-state machine; feedback control.

* Corresponding author. Tel.: +31 (15) 27 82760; fax: +31 (0)15 278 31 79.
E-mail address: r.l.landman@tudelft.nl.
1. Introduction

A powerful means to mitigate the negative effects of natural and man-made disasters (Walter et al., 2001) is to evacuate the population of threatened areas. To minimize the evacuation times, loss of life and other negative impacts, plans and protocols for evacuation management (Urbina and Wolshon, 2003; Wolshon et al., 2005b) are setup by the authorities. The design of robust evacuation plans and strategies is often supported by state-of-the-art modeling, control and analysis techniques from the field of science.

However, since these plans account for many uncertainties (e.g. evacuation demand, infrastructure supply, hazard etc.) they offer a rather conservative solution in which a relatively low performance is guaranteed. Better network performance can be expected when the control signals for available dynamic traffic management (DTM) measures are based on real-time data. Moreover, to better utilize evacuation routes in practice, the available means are often limited to static route guidance, information provision and reversing lanes. Even though data is gathered for the monitoring of evacuation routes, no operational control systems are available that intelligently compose route guidance instructions based on prevailing traffic conditions to operationalize evacuation objectives dynamically.

In this context we propose a service level-oriented route guidance approach that is able to realize evacuation objectives in a comprehensible and systematic way based on available monitoring data (speed profiles over the evacuation routes). Examples of such objectives are: increasing network outflow, respecting equity constraints and realizing specific functional utilization of the infrastructure during process of network degradation and recovery. The evacuation objectives (often qualitatively defined) first need to be translated into feasible control targets (network states) that can be maintained. The approach is of the predictive feedback control type (Wang et al., 2003), hence it properly deals with changes in traffic demand and infrastructure supply and it prevents undesired system oscillations. The computational demand is limited and scales linearly with the number of controlled actuators, because the algorithm is simple, and because a single (one-shot) prediction of future traffic conditions is needed. This makes the approach well applicable in real-time (Landman et al., 2012).

The paper is structured as follows. In Section 2 a short overview is given of the ongoing developments in the field of evacuation management that are relevant for this paper. The important requirements are discussed on the application of dynamic route guidance in large-scale networks, in order to determine useful control signals. The proposed control approach is discussed in Section 3 and evaluated in Section 4 by means of a simulation test case. A comparison is made with model predictive control (MPC)-based route guidance that realizes System Optimal (SO) conditions, with a feedback controller that realizes user equilibrium (UE) conditions and with a do-nothing scenario. Hence, in the UE scenario evacuees are well informed on the quality of the alternative evacuation routes, such that they can optimize their own travel time. In the do-nothing scenario where no information or guidance is given, evacuees do not update their initial route choice. The results are presented in Section 5 followed by a discussion in Section 6.

2. Background

The aim during an evacuation is to guide as many people in the least amount of time away from the (potential) impact area over the available infrastructure. The challenge of finding evacuation instructions that minimize evacuation times is therefore often addressed in the field of evacuation research. In (Sbayti and Mahmassani, 2006; Chiu et al., 2007; Huibregtse et al., 2009, 2010; Pel et al., 2009) methods are presented that minimize evacuation times based on optimal departure times, route choices and destination choices. Few control approaches can be found in literature that aim at giving real-time optimal instructions during the evacuation. In (Liu et al., 2007) a MPC-based approach is presented that aims at
realizing SO conditions in the evacuation network by manipulating the turning directions of traffic in the network. Another approach to realize optimal network outflow is presented in (Tuydes and Ziliaskopoulos, 2006) in which the application of contra flow is optimized. In (So and Daganzo, 2010) a control approach is presented that ensures optimal outflow of an evacuation route, if at every inflow point the minimum downstream capacity is maintained (e.g. by means of ramp metering).

Contrary to state-of-the-art theory, the available means and controllable variables (e.g. departure time, destination choices) in operational evacuation management are limited to achieve optimal network performance. Data is gathered by means of existing data collection systems, CCTV cameras or other mobile counting units (Urbina and Wolshon, 2003; Wolshon et al., 2005a). The data is used for monitoring purposes, so that evacuation coordinators can decide upon where to guide traffic by static means (PBS&J, 2000). Demand management is done by distributing information among the evacuees by means of systems like Highway Advisory Radio (HAR) and Dynamic Message Signs (DMS). Even though staged evacuation proves to be beneficial according various researches, it is not applied in practice during large-scale evacuations. Furthermore, lanes are often reversed (contra flows) to increase the evacuation capacity of evacuation routes (Abdelgawad and Abdulhai, 2009; Theodoulou and Wolshon, 2004; Urbina and Wolshon, 2003).

The contrast between state-of-the-art and state-of-practice in evacuation management indicates the need for comprehensible control approaches that operationalize evacuation objectives systematically and in real-time. Real-time data can be made available, and the application of dynamic route guidance based on this data can be considered an interesting means to distribute traffic over the evacuation routes in line with the evacuation objectives.

However, when dynamic route guidance is applied on a network level based on arrival or measured travel times, time delays occur between the moment of determining the control signal and the corresponding impact on the network performance (Wahle et al., 2000; Pavlis and Papageorgiou, 1999; Hoogendoorn, 1997). This, in turn, may result in system oscillations that might make given control signals suboptimal or even counterproductive. The application of predicted state estimates will prevent this from happening (Wang et al., 2003), enabling stable realization of target network states.

Control approaches that take real-time information into account when determining control signals are able to take unforeseen changes in traffic demand, infrastructure supply and traffic state evolution into account (i.e. feedback controllers correct for deviations between the system output and the target output). Feedforward approaches, on the other hand, use a single state estimate to optimize control signals over the entire evacuation time period (e.g. like the techniques used to design robust evacuation plans). These plans account for many uncertainties, so that the most robust solution is a rather conservative one. By using real-time data part of the uncertainty is dealt with, so that control signals are tailored to the prevailing traffic conditions and the resulting achievable network performance expected to be better. In MPC frameworks the optimization of control signals is therefore done in a rolling horizon based on real-time data.

The computational demand is also an important aspect that needs to be taken into account when operationalizing a control system on large-scale. It is known that the computational demand of optimization based control techniques scales in general exponentially with the number of variables (i.e. the number of actuators and the size of the control horizon) that need to be optimized every control interval. Hence, these control applications are still pending to be implemented on a network scale, due to their computational demand and complexity. Nevertheless, from applying optimal controllers in an offline setting, important insights can be gained into the essential traffic phenomena that cause decreased network performance to find less complex, but efficient control heuristics (Wu and Chang, 1999).

Heuristic approaches can be developed that aim at systematically preventing the phenomena that cause decreased network performance (i.e. outflow) like the capacity drop (Hall and Agyemang-Duah,
1991), blocking back of queues (Knoop et al., 2006) and underutilization of available capacity in available routes towards a destination. The capacity drop is a typical phenomenon that occurs at freeways due to congestion. The outflow of a queue is namely smaller than the maximal flow during free flow conditions. Also blocking back can severely decrease the performance of both freeway and urban road networks. It occurs when a queue spills back to a location where it starts to hinder traffic that does not need to pass the bottleneck that causes the queue. To conclude, it is desirable to utilize all available capacity in the network to maximize network outflow.

During evacuations the most promising application of route guidance to improve network performance is to prevent the blocking back phenomenon in evacuation routes. During large scale evacuations, traffic demand generally strongly exceeds the available supply over a long time period. It is therefore unlikely that evacuation routes remain available with redundant capacity. Moreover, to achieve optimal performance, the minimal capacity downstream an inflow location should not be exceeded (So and Daganzo, 2010). It will be difficult to prevent the capacity from being exceeded at every location in the route over time by means of route guidance, i.e. using an actuator whose control actions are significantly delayed before having effect. The most appropriate application of route guidance during evacuations therefore seems preventing outflow locations within a route from becoming blocked (i.e. by storing traffic at network parts where it least influences the network outflow).

3. Control approach

In this section a control approach is presented in which service levels are used to prevent blocking back. Service level-oriented route guidance can be applied to prevent or delay routes from degrading beyond critical performance values (e.g. in terms of travel time) at which queue spill back negatively influences the network outflow. To this aim, traffic is sent to the route alternative at which additional traffic has the lowest impact on network degradation. However, from a user point of view, large travel time differences are undesired if the evacuation routes are used at capacity without upstream infrastructure being threatened by spill back. It might, however, be better to relax equity requirements when spill back threatens upstream turning directions towards other important safe zones.

Aspects like the size of hazard and the size of the hindered flow determine the acceptable deviation from the UE conditions in order to prevent blocking back phenomenon in a route. To operationalize the trade-off between maximum network outflow and equity related objectives, a stepwise service level degradation and recovery scheme is applied. This scheme degrades routes towards their critical performance value while respecting user interests (i.e. some maximum travel time difference over the routes). When the most vulnerable route for spill back reaches its critical performance value, the queue is stabilized by sending traffic to the alternative route. The alternative is then allowed to further degrade until some maximum acceptable performance difference is realized. Hence, this mechanism enables the prevention of reduced network outflow while taking the equity bounds under various circumstances into account. It also ensures that the available capacity among the evacuation routes becomes fully utilized in an early stage and that during undersaturated conditions bottlenecks are released at the same time.

A finite-state machine is designed to control the switching of the target service levels. In the next section its functioning is described for a single route set consisting of two routes and for multiple route sets where routes overlap.

3.1. Service level control for two routes

The finite-state machine dynamically sets the target service levels for routes based on predicted traffic conditions. The target service levels are designed such that spill back to upstream infrastructure is
prevented, in line with its impact on the network performance. Based on the prevailing traffic conditions with respect to the target service levels, the finite-state machine decides on which route the target service level is maintained, so that the other route is allowed to further degrade or recover during respectively over- and undersaturated conditions. Oversaturated means that the traffic demand for both routes is larger than their joint capacities, resulting in increasing congestion and decreasing service levels. If the demand for both routes is smaller than this joint capacity, routes are said to be undersaturated (even though congestion can still be present), resulting in performance recovery.

In Figure 1 the control loop is shown. Note that 'process' stands for the real traffic process. The simulation time step counter $k$ and the control time step counter $k_c$ indicate time instants $kT$ and $k_cT_c$, with $T$ and $T_c$ respectively the simulation and controller time step sizes. We assume that $T_c = MT$, with $M$ being an integer. The simulation process is modeled by the discrete-time system $f$:

$$x(k+1) = f(x(k), u(k_c), d(k)), \quad (1)$$

with $Mk_c \leq k \leq M(k_c + 1)$, $x(k)$ the state vector of the system (e.g. flow, speed, density) at simulation step $k$, $u(k_c)$ the control input at controller time step $k_c$ (e.g. split fraction) and $d(k)$ the disturbance vector (e.g. demand) at simulation step $k$.

When the controller is activated, the state vector $x(k)$ is the initial state for a model-based prediction that is used to define the future traffic conditions $\hat{x}(k)$ over a prediction horizon that is equal to or larger
than the maximum travel time through the routes. Based on this prediction, the departure travel time \( \tau_r(k_r) \) for each route \( r \) is determined by means of the trajectory-based method elaborated in (van Lint, 2010). In the remainder of the paper we assume that from a route set \( s \in S \) consisting of routes \( r \in \{1, 2\} \), the main route (preferred route) is indicated by \( r = 1 \) and the alternative by \( r = 2 \).

The travel times indicate the current performance of each route, and they are easily translated to an average travel speed \( v_r(k_r) \) in km/h in combination with the route length. Based on these performance indicators and the predefined service levels, the finite-state machine decides which feedback algorithm to activate to determine the control signal.

The service levels are expressed in terms of travel time or speed, and in Table 1 an example is given in terms of speed, because this gives a generic performance description that is not dependent on route lengths. With respect to the implementation, the service levels are always translated into travel times, because this prevents unrealistic and unfair travel time differences between route alternatives from being realized and maintained. Due to the relation \( \tau_r = L_r / v_r \), with \( v_r \) the speed, \( L_r \) the length and \( \tau_r \) the travel time of route \( r \), small variations in low speeds result in much larger travel time differences than small variations in high speeds.

For each route, every service level \( l_r(k_r) \) is determined by an upper boundary \( v^{ub}_r(l_r(k_r)) \) and lower boundary \( v^{lb}_r(l_r(k_r)) \). Notice from the table that the boundaries of the same service level can be different for the different routes, and that the level indices increase when the performance degrades.

<table>
<thead>
<tr>
<th>Levels</th>
<th>Main route</th>
<th>Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_r(k_r) )</td>
<td>( v^{ub}_r(l_r(k_r)) )</td>
<td>( v^{lb}_r(l_r(k_r)) )</td>
</tr>
<tr>
<td>1</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

At the moment the finite-state machine is activated, it updates its state \( S(k_r) \in \{1, 2\} \) and the active service levels \( l_r(k_r) \), based on a comparison between the actual route performance \( v_r(k_r) \) and the active service level boundaries \( v^{ub}_r(l_r(k_r - 1)) \) and \( v^{lb}_r(l_r(k_r - 1)) \) from the previous control interval \( k_r - 1 \). This process is formalized in Figure 2, with on the arrows the triggers that trigger a state transition including the actions of updating the target service levels. The outer loop over the finite-state machine states is followed during the degradation process and the inner loop during the recovery process. If the performance of the route that is allowed to degrade or recover remains within its service level boundaries, no state transition is triggered and the active service levels remain the same. This is indicated by the triggers and actions within each state.
To prevent frequent switching, an extra threshold is added to the boundaries that trigger a state transition. This threshold is a constant value $\mu$ defined in terms of travel time. However, since the process description is in terms of speed, $\mu$ is translated into the terms $\varepsilon^{lb}$ and $\varepsilon^{ub}$ expressing the threshold as a function of the route length $L_r$, the considered reference value $v^b_r(l(k_c))$ or $v^b_r(l'(k_c))$, and the defined travel time difference $\mu$. The upper and lower boundaries become respectively $v^b_r(l'(k_c)) + \varepsilon^{ub}$ and $v^b_r(l'(k_c)) - \varepsilon^{lb}$.

The updated state $S(k_c)$ is used to select and execute the corresponding feedback algorithm with the setpoint $v^b_r(l'(k_c))$ to determine the control signal. In state $S(k_c) = 1$ and $S(k_c) = 2$ the performance of respectively the main route and alternative is kept constant, allowing the other route to degrade or recover. The applied feedback control laws are given in (2). They determine the desired split fraction $\beta^d_n(k_c)$ for the controllable traffic flow at the node $n$ directly downstream the VMS towards destination $d$ in control interval $k_c$.

\[
\beta^d_n(k_c) = \begin{cases} 
\hat{\beta}^d_n(k_c - 1) + \alpha \left( \tau_1(k_c) - \tau^b_2(l'(k_c)) \right), & \text{if } S(k_c) = 1 \\
\hat{\beta}^d_n(k_c - 1) - \alpha \left( \tau_2(k_c) - \tau^b_2(l'(k_c)) \right), & \text{if } S(k_c) = 2.
\end{cases}
\]  

(2)

It has to satisfy $0 \leq \beta^d_n(k_c) \leq 1$, and therefore might need to be truncated by $\hat{\beta}^d_n(k_c) = \min(\max(0, \beta^d_n(k_c)), 1)$. The realized split fraction towards the main route $\beta^d_n(k_c)$, however, depends on the compliance (driver response) $\gamma$ of the controlled flow, and the nominal fraction (default behavior) towards the main route $\beta^{bd}_n(k_c)$. The implemented split fraction at time step $k_c$ towards the main route then becomes

\[
\hat{\beta}^d_n(k_c) = (1 - \gamma)\beta^{bd}_n + \gamma \hat{\beta}^d_n(k_c)
\]  

(3)

and towards the alternative

\[
\hat{\beta}^d_n(k_c) = 1 - \hat{\beta}^d_n(k_c)
\]  

(4)
3.2. Service level control for overlapping routes

In the previous section we have introduced a route guidance approach that controls service levels over two alternative routes. However, when applying the method on a network level with multiple actuators, controlled routes might overlap. Interaction between the controllers then occurs, because the traffic that flows towards overlapping route stretches can be manipulated from multiple directions. In the remainder, the interaction mechanism between the controllers is described and it is explained why the mechanism leads to the utilization of redundant capacity and stepwise performance degradation and recovery within the routes.

Route sets can share overlap with each other by their main or alternative route. The number of possible overlap combinations is determined by the number of involved sets. It is assumed that all routes \( r \in \{1, 2\} \) within the considered route sets \( s \in S \) initially perform within their first service level \( l_{s1}(0) = 1 \).

At the moment a bottleneck becomes active within a route stretch, first redundant capacity is utilized from the route sets \( z \in Z \) with \( Z \subset S \) that are directly influenced. If the problem concerns the main route of route set \( z \), the target service level \( v_{1}^{z}(l_{z1}(k_{c})) \) is maintained by sending traffic to the corresponding alternative to either use its redundant capacity or degrade it until its first service level lower boundary \( l_{z1}(k_{c}) \). In case the bottleneck concerns the alternative, the performance is allowed to degrade to \( v_{2}^{z}(l_{z2}(k_{c})) \), its service level index subsequently increased \( l_{z2}(k_{c}) = l_{z2}(k_{c} - 1) + 1 \), and the corresponding upper boundary \( v_{2}^{z}(l_{z2}(k_{c})) \) maintained. Traffic is then sent back to the main route to use its redundant capacity or to degrade its performance to the active service level lower boundary \( v_{1}^{z}(l_{z1}(k_{c})) \).

At route sets \( y \in Y \) with \( Y \subset S \) that are not directly influenced by the bottleneck, redundant capacity is used as soon as the performance degrades within the route stretches that overlap. Again, traffic is directly sent towards the alternative within route set \( y \) if the stretch belongs to a main route by maintaining \( v_{1}^{y}(l_{y1}(k_{c})) \). The controller of set \( y \) basically turns the capacity over to the surplus of traffic from set \( z \) with which it overlaps. The same reasoning holds if the overlap is realized by the alternative of set \( y \), however, after the performance degraded to its first service level lower boundary \( v_{2}^{y}(l_{y2}(k_{c})) \).

During over- and undersaturated conditions, this mechanism realizes stepwise degradation and recovery within the routes, as long as the controllable flow and its compliance are large enough to completely relieve active bottlenecks. The reasoning is then in terms of storage space with respect to the difference between the actual performance and the target service levels.

4. Test case

By means of the test case the functioning of the proposed control approach during an evacuation is evaluated and compared to SO conditions, UE conditions (perfectly informed evacuees) and a do-nothing scenario (not informed evacuees). The test case illustrates that applying no control and even realizing user equilibrium conditions can have a negative impact on the network performance. Moreover, we illustrate that the proposed control approach is well able to approximate the system optimal solution.

The prevailing evacuation objectives are: protecting the network’s outflow while keeping the travel time difference over main routes and their alternatives limited to 15 minutes. However, when upstream infrastructure becomes threatened, the equity constraint is relaxed to prevent the outflow to other safe zones from becoming blocked. In the remainder of this section, the applied traffic flow model, the performance indicators and the set-ups of the test case, the finite-state machine and the MPC controller are discussed.
4.1. Applied traffic flow model

The macroscopic first-order, multi-class, cell-based traffic flow model Fastlane (van Lint et al., 2008) has been used for the process simulation, the state predictions of the finite-state-machine, and the optimization procedure within the MPC controller. Fastlane propagates traffic flows destination dependent through the network, avoiding the incorrect manipulation of flows by route guidance in case of a non-destination oriented flow representation. This also allows for correct representation of the onset and dissolution of congestion due to blocking back.

4.2. Performance Indicators

The different control methodologies are evaluated based on the network performance indicator: the total time that vehicles have spent in the network (TTS). The time spent by \( N(k) \) vehicles in one time step is \( T N(k) \) and the total time that the vehicles spend in the network over a period \( k = \{1, 2, ..., K\} \) with \( K \) the total number of simulation time steps becomes

\[
J_{TTS} = T \sum_{k=1}^{K} \sum_{m \in M} \sum_{c \in C_m} \rho_{m,c}(k) \lambda_{m,c},
\]

with \( \rho_{m,c}(k) \) the vehicle densities over the cells \( c \in C_m \) of all links \( m \in M \) in the network, \( \lambda_{m,c} \) the corresponding cell lengths. This equation is also used as objective function to determine system optimal conditions by the MPC approach.

4.3. Test case set-up

A scenario is chosen in which city \( A \) with a total population of 160,000 inhabitants needs to be evacuated between 5:00h in the morning and 0:00h at night. In line with the assumptions used in (Friso et al., 2011), 80% of the population is assumed to be self reliant and 20% not self reliant. The car occupation rates are respectively 2.3 and 5 persons/car, resulting in a total evacuation demand of 58,000 cars. The corresponding demand and the departure curve are given in Figure 3.

Figure 3: Specification of inflow and departure curve.
The network layout and characteristics are given in Figure 4. Shelter can be found in the neighboring cities B, C, and D. City D can only be reached by a single route and the other cities each have two route alternatives; a preferred main route $r = 1$ and an alternative $r = 2$. Of the total demand, 40% moves to B, 40% to C, and 20% to D. The nominal split fractions to the main routes of cities B and C are respectively 60% and 55%, and the compliance rate of traffic to a given advice is assumed to be 100%. The route guidance actuators that are used to adjust the split fractions between the main routes and their alternatives are located in the south. The critical bottlenecks within this scenario are the off-ramps at the safe cities and their effective capacity is given in Figure 4.

These settings result in the following scenario. Until 10:00h the evacuation routes are undersaturated and from then on the routes become oversaturated until 15:00h. If no control is applied, congestion spills back beyond both bifurcation points 1 and 2 within the main routes.

4.4. Finite-state machine set-up

The target service levels are given in Table 2. The desired maximum travel time difference under noncritical conditions is reflected by the maximum travel time difference within a service level (900 seconds). By means of simulation the critical travel times are determined. Bifurcation 1 in the main route...
to city B and bifurcation 2 in the main route to city C both become blocked around 5100 seconds. Hence, when the main routes degraded to their 6th service level, their performance is kept constant at all costs to prevent blocking back by allowing the alternatives to fully degrade. The service levels in this example are degraded in steps of 900 seconds, as can be seen in Table 2, except for the first service level of the alternatives. This is done to keep the maximum travel time difference over the routes within each service level equal to 900 seconds. To conclude, the threshold $\mu$ is chosen as 10 seconds and the feedback gain $\alpha$ is chosen as 0.001.

Table 2: Service level table for the test case. The odd columns indicate the service level upper (ub) boundaries and the even columns the lower boundaries (lb) in terms of travel time (s).

<table>
<thead>
<tr>
<th>Level</th>
<th>Main route B</th>
<th>Alternative B</th>
<th>Main route C</th>
<th>Alternative C</th>
</tr>
</thead>
<tbody>
<tr>
<td>l ub</td>
<td>r ub (l ub)</td>
<td>r ub (l ub)</td>
<td>r ub (l ub)</td>
<td>r ub (l ub)</td>
</tr>
<tr>
<td>1</td>
<td>730</td>
<td>1630</td>
<td>905</td>
<td>1630</td>
</tr>
<tr>
<td>2</td>
<td>1630</td>
<td>2530</td>
<td>1630</td>
<td>2530</td>
</tr>
<tr>
<td>3</td>
<td>2530</td>
<td>3430</td>
<td>2530</td>
<td>3430</td>
</tr>
<tr>
<td>4</td>
<td>3430</td>
<td>4330</td>
<td>3430</td>
<td>4330</td>
</tr>
<tr>
<td>5</td>
<td>4330</td>
<td>5100</td>
<td>4330</td>
<td>5200</td>
</tr>
<tr>
<td>6</td>
<td>5100</td>
<td>9999</td>
<td>5230</td>
<td>9999</td>
</tr>
</tbody>
</table>

4.5. Model Predictive Control and controller setup

In a MPC scheme the optimal control signals $u^*(k)$ are determined at each control interval $k$ over a prediction horizon $N_p$. A control horizon $N_c (< N_p)$ is selected to reduce the number of variables for optimization, and improve the stability of the system. In the optimization procedure, a model is used to evaluate the system performance $J(\hat{x}(k), u(k))$ by the evolution of states $\hat{x}(k)$ over the prediction horizon, based on the current state of the system $x(k)$, the expected disturbances $d(k)$, and some planned control trajectory $u(k)$. From the resulting optimal signals only the first sample $u^*(k)$ is applied to the process and in the next control time step ($k+1$), a new optimization is performed. For more information on MPC see (Hegyi, 2004) and the references therein.

When applying MPC, it is very important to determine the correct settings for the prediction horizon $N_p$, the number of variable control signals within the control horizon $N_c$, and of course the size of the parameter $M$ that directly determines the size of the control interval $T_c = MT$ for a given simulation time step size $T$. The main rule for tuning $N_p$ is that the prediction horizon should be long enough to cover the important system dynamics to find optimal conditions. A 60 minute prediction horizon of 10 control intervals ($M=100$, $T=3.6$ seconds) and 4 variable control signals per actuator are chosen to approximate system optimal conditions.

5. Results

First the control signals of the model predictive controller and finite-state machine are discussed with respect to system optimal conditions. Then the resulting travel times, queue lengths and the total time spent by vehicles in the network are compared.
5.1. Control signals

To achieve system optimality the MPC controller aims at directing traffic such that the fastest route is used during undersaturated conditions and that bottlenecks within the main route and its alternative become active and released at the same time. To this aim, we can see in Figure 5 that traffic is first sent to the main route and then a part of the flow is sent back to the alternative to utilize its redundant capacity. When the controller detects the possibility of the upstream bifurcation becoming blocked within the main route to city B (i.e. within the considered prediction horizon) it sends even more traffic to the alternative. The controller, however, does not succeed in releasing the bottlenecks at the same time due to the following interesting difficulty of applying MPC within an evacuation network. The prediction horizon should be long enough to account for all relevant dynamics with respect to system optimality. In this case, the controller realizes a large travel time difference over the main route to city B and its alternative to prevent blocking back. However, the travel time difference during saturated conditions might become larger than the prediction horizon of the controller. This means that the controller will not be able to anticipate on a sudden decrease of evacuation demand. Hence, when the evacuation demand drops, the queue in the main route will dissolve before the queue at the alternative does, resulting in underutilization of available capacity.

The finite-state machine follows the same strategy as the optimal controller, however, instead of instantly realizing, it gradually finds the required control signal. Nevertheless, if the controller is tuned well, the oscillations within the control signal remain limited. Strong oscillations can be found when realizing user optimal conditions, because the turn fractions constantly need to be adjusted to keep travel times over the routes equal.

![Figure 5: Control signals of the Finite-state machine, the MPC-based controller, and the UE feedback controller.](image-url)
5.2. Queue lengths

The horizontal red lines in Figure 6a indicate the locations of the bifurcations within the main routes to cities B and C. It can be clearly seen that the finite-state machine stabilizes the queue in main route to city B just downstream the bifurcation to city D. To this aim we accept the queue on the alternative to become 10 km of length. The MPC controller puts an even longer queue on this alternative. Notice that it does not matter where the queues are allocated during oversaturated conditions as long as both bottlenecks are active. The disadvantage of UE conditions also becomes clearly visible. By keeping the travel times over the main route and alternative equal, the main route queue spills back beyond the bifurcation point to city D. Notice the increased speed of the queue length grow in the main route to city B at the moment the bifurcation to city D becomes blocked. This is because evacuees to city D now also flow into the queue. This makes it even harder to stabilize a queue by means of control. In the do-nothing scenario the main route queue to city B even blocks the bifurcation to city C. This, however, does not influence the total network performance as long as the bottleneck within the main route to city C remains active.

5.3. Travel times and total time spent

In Figure 6b the travel times clearly show the stepwise degradation and recovery scheme of the finite-state machine, the equal travel times over the routes for the user equilibrium controller and the extreme travel time differences between the routes if evacuees are either not informed on prevailing conditions. The total time spent of vehicles are as follows: do-nothing 78.3 kveh-h, user equilibrium 61.6 kveh-h, finite-state machine 56.7 kveh-h and MPC 56.3 kveh-h. Hence, the finite-state machine shows an improvement of 28% with respect to the do-nothing scenario and 8% with respect to user equilibrium conditions. It only performs 0.6% worse than the model predictive controller.

Figure 6: Scenario results with a) the queue lengths and b) the travel times over all routes
6. Discussion

The proposed control approach generates route guidance signals that realize the evacuation objectives of the authorities by degrading and recovering service levels of routes stepwise. This is done in a systematic and comprehensible way based on real-time speed profile information from the evacuation routes. The prerequisite for establishing network states that reflect the evacuation objectives, is that the objectives (often qualitatively defined) are carefully translated into target service level boundaries that can be maintained by the controller. The boundaries can be chosen such that a trade-off is made between the network performance and interests of the evacuees depending on the threat of the hazard.

The test case illustrates large potential network performance benefits of the finite-state compared to do-nothing scenario (28% decrease TTS) and user equilibrium conditions (8% decrease TTS). Furthermore, system optimality can be approximated (0.6% increase TTS) if the approach is tuned in line with the optimal strategy followed by the MPC controller.

With respect to the scalability of the control approach, the following remarks can be made. The test case illustrates the functioning of the control approach when preventing blocking back from a single bottleneck to an upstream bifurcation point. Within more complex situations with multiple bottlenecks, realizing a certain service level does not necessarily mean that the blocking back phenomenon is adequately dealt with, because there is no unique mapping of the queue distribution within a route and the corresponding travel time. However, protecting the performance of an important route that is vulnerable to spill back, will at least delay the moment blocking back occurs. Moreover, the finite-state machine approach can also be used to redistribute queues at multiple controlled bottleneck locations within the evacuation routes. The result is that the evolution of queues within the routes is fully controlled and that each queuing state corresponds to a unique travel time. Hence, the combination of service level-oriented route guidance and the redistribution of queues within a route will enable adequate handling of the blocking back phenomenon in a more realistic setting.

It is well-known that the computational time of the MPC approach in general increases exponentially with respect to the number of control signals that need to be optimized. This severely limits its applicability in practice. The finite-state machine, however, does not have such high computational requirements, because only one state predication is needed to determine the departure travel times over all routes as input to the finite-state machine that determines the control signals. The computational time of the finite-state machine therefore scales linearly with the number of controlled actuators. To conclude, the approach can be considered comprehensible, because the realized control signals are always easy to interpret.

Acknowledgements

The research presented in this article is part of the research program “Traffic and Travel Behavior in case of Exceptional Events”, sponsored by the Dutch Foundation of Scientific Research MaGW-NWO.

References


planning anticipating traveler compliance behavior. Proceedings of the 12th International Conference on Travel Behavior Research, Jaipur, India.


