Modeling and control of congestion phenomena

Levering, N.A.C.

Publication date
2024

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 5

Online velocity measurements for optimal routing decisions

5.1 Introduction

With typically multiple routes connecting a driver’s current location and their desired destination, route selection is a key problem in transportation research. Drivers wish to be advised an optimal route, i.e., a route whose travel time minimizes a given objective function. However, as we have seen in the previous chapters, the identification of such an optimal route is non-trivial, where an important complication lies in the fact that travel times are strongly affected by the permanently fluctuating congestion levels along the candidate routes. The Markov model introduced in Chapter 2 captures the impact of traffic events on these congestion levels with an environmental background process, and it is assumed that we are able to observe the state of this process at certain points in time. In this chapter, we again consider optimal routing in stochastic road networks, but with a different perspective concerning the availability of information: it is assumed that, besides historic speed data, we only have access to real-time velocity measurements of a part of the vehicles in the road network.

If we focus solely on the impact of recurrent congestion, having access to online velocity measurements allows employing models that describe the relation between traffic densities and vehicle speeds explicitly. In this chapter, we therefore examine the use of a stochastic fundamental diagram in the context of routing, where we wish to identify the route which minimizes the expected travel time. This stochastic fundamental diagram (SFD) maps each traffic density to a velocity distribution. Therefore, the recorded velocity measurements can be used to get an impression of the network densities. In the same way, the diagram provides an indication of the (average) vehicle-to-vehicle dynamics, giving predictions for the network densities at future points in time. The objective is to develop a data-driven framework that estimates the current and future traffic densities, and that uses these estimates so as to devise an efficient algorithm that outputs the optimal route.

Relevant literature

As described in Chapter 1, fundamental diagrams outline the relationship between traffic density, traffic flow, and vehicle velocities. The seminal work by Greenshields in [90] proposed a linear relation between traffic densities and vehicle velocities (see, e.g., Figure 1.2),
but many other fundamental diagrams have been proposed that aim to provide a better fit (e.g., in \cite{57, 89, 214}). Regardless of the fundamental diagram under consideration, the vehicle velocities in the network can be seen as a function of the traffic densities. Therefore, given that one would know the fundamental diagram that locally applies, wherein each velocity value corresponds to a unique traffic density, one may infer the traffic densities by observing the vehicle velocities in the network.

The fundamental diagrams mentioned above are of a deterministic nature: a given traffic density corresponds to a deterministic velocity. However, due to, e.g., individual driver characteristics, one could observe various vehicle velocities for a given traffic density. This phenomenon is confirmed by Helbing \cite{103}, who found backing for the claim that the velocity distribution for a given traffic density is approximately Gaussian. An alternative stochastic model is introduced in the work of Knoop and Hoogendoorn \cite{136}, involving the introduction of spatial variation in the traffic density.

The approach of taking the speed–density relationship to a stochastic domain is proceeded by Ni et al. \cite{166}, Qu et al. \cite{183}, and Wang et al. \cite{215}. In \cite{215}, the velocity given the traffic density follows a Gaussian distribution, whose mean may be inferred from a deterministic fundamental diagram, and whose variance is some parametric function of the traffic density. Instead of assuming Gaussianity, a more general approach is followed in \cite{166}, where the traffic density is mapped to the moments of the velocity, and these moments are then related to the parameters of the chosen probability distribution. Finally, a non-parametric approach for obtaining the velocity distribution given a traffic density is presented in \cite{183}.

The procedures in \cite{166, 183, 215} give rise to a so-called stochastic fundamental diagram (SFD), in which each traffic density is mapped to a velocity distribution (rather than a single, deterministic velocity, that is). If one knows the stochastic fundamental diagram that applies at a given location in the network, then one may devise a statistical procedure to use local real-time velocity measurements to estimate the corresponding current local traffic density. These real-time velocity measurements can be obtained from local GPS velocity data \cite{119}, which is nowadays widely available from, e.g., cell phones.

Real-time measurements primarily serve to capture the current traffic densities on the network. For routing purposes, however, one needs to have a handle on future traffic densities as well \cite{96, 137}. In the previous chapter, we operationalized a vehicle speed model that predicts the future state of the network from both its current state and historic data on vehicle velocities and incident characteristics. However, we assume full information on the current network state, and do not consider the relation between traffic densities and velocities explicitly. Instead of historic velocity data, the work of Neumann et al. \cite{165} proposes the use of a historic traffic flow estimate for every hour of each day of the week. However, due to the highly dynamic behavior of traffic densities, one should not base optimal routing algorithms solely on historic estimates, but, in addition, use real-time congestion information.

In this chapter, we resort to the use of the kinematic wave model of traffic flow theory, pioneered by Lighthill and Whitham \cite{148} and Richards \cite{185}. It allows the modeling of traffic waves, describing the evolution of the current congestion level towards future congestion levels. A numerical scheme to solve the partial differential equation underlying the dynamics of \cite{148} and \cite{185}, known as the Godunov scheme, can be found in \cite{87}.
Assuming a trapezoid-shaped fundamental diagram, an easily implementable version of [87] is the cell transmission model (CTM) presented by Daganzo [45]. While [45] only concerns segments of road, the CTM can also be employed in more general network structures if the turning proportions, priority fractions, and rate of incoming traffic at every junction are known [46]. A nice interpretation of the CTM is provided in [141], which results in an overarching framework, encompassing a broad set of local demand and supply functions. Importantly, this general setup does not impose any strict assumptions on the fundamental diagram used. Lastly, whereas the aforementioned frameworks apply to deterministic traffic flows, an approach to model traffic propagation in a stochastic scenario is presented in [200].

A main step in our approach is that we convert traffic density estimates, as obtained from the Godunov scheme, into predictions of the expected per-link travel times at future time points, which can then serve as time-dependent weights in a routing algorithm. Even if it is not allowed to reroute drivers (i.e., the full path to travel is determined upon departure), it is known that, in general, this problem is NP-hard [170]. Only under some constraints concerning the ordering of these weights (including the FIFO property discussed in Chapter 2), an optimal path can be found in polynomial time, using label-updating procedures in the spirit of the seminal Dijkstra algorithm [55]. An overview of such procedures can, e.g., be found in [51].

The obvious idea behind using real-time information is that one could reduce travel times by rerouting. Evidently, in case one allows rerouting, the problem becomes even more complex. That is, it would yield a decision problem at each junction the vehicle reaches, viz. the choice which next node to travel to, dependent on the time the vehicle reaches the junction. The state space thus consisting of junctions and potential arrival times at these junctions, the underlying dimensionality poses a huge challenge on the computation of an optimal routing procedure, severely limiting the derivation – and as a consequence, application – of such procedures in real-life large-scale networks.

Contributions

Summarizing, there is a substantial body of literature both on relating velocity measurements to current traffic densities, and on inferring future traffic densities from current traffic densities. Yet, to the knowledge of the authors, no procedure has been developed that utilizes velocity measurements with the purpose of providing optimal routing advice through traffic density predictions. It is the objective of this chapter to fill this void. A more detailed account of the contributions of our work is as follows:

- In order to relate real-time velocity measurements to current traffic densities, we propose to use an underlying model inspired by [166, 215]. That is, for a deterministic relation between average velocity and traffic density, and a corresponding covariance function, we construct an SFD without imposing strict parametric assumptions.

- Using this SFD and given real-time velocity measurements, we outline a conventional maximum likelihood procedure with the aim to identify the most likely current traffic densities in the network. We show that this method is often superior to the more basic approach of first averaging the velocities and then identifying the corresponding traffic densities.
We proceed by setting up a procedure to predict the expected travel time of any route in the network. For this, we heavily rely on the CTM, as it allows us to evaluate future traffic density estimates, using only the current traffic density estimates, and the turning proportions, priority fractions, and rate of incoming traffic (as estimated from historic data).

Putting everything together, we present a routing algorithm that aims to minimize the expected travel time. This algorithm also allows for an online version, in which new velocity measurements are used to obtain more accurate travel-time estimates while the driver has not yet reached their destination. As this online version facilitates efficient rerouting, the expected travel time can be further reduced.

Finally, we provide a selection of numerical experiments in which we implement our procedure. In particular, we validate our method by studying its accuracy and robustness. The experiments show that the routing algorithm is highly efficient, in that it can be used in large-scale networks. We conclude by demonstrating that a reduction in expected travel time can be achieved by taking expected changes in traffic densities into account.

Organization

The chapter is now structured as follows. Section 5.2 reviews the concept of a stochastic fundamental diagram, to then describe the problem of minimizing expected travel times in highway networks whose dynamics are modeled through a random velocity-density relation. A procedure to estimate both the current and future traffic densities from observed velocities is detailed in Section 5.3. Section 5.4 contains a routing algorithm that takes these traffic density estimates as input, and aims to find the path with minimum expected travel time for a given origin-destination pair. Numerical examples that demonstrate the applicability of our procedure are presented in Section 5.5. Finally, Section 5.6 contains concluding remarks.

5.2 Preliminaries

A substantial number of vehicles is equipped with in-vehicle GPS systems that may communicate their speed to a central unit. These recorded speeds may be used so as to obtain an impression of the current traffic densities in the network. Indeed, by the fundamental diagram, relating vehicle flow, vehicle speed, and vehicle density, one can aim to translate speed measurements into traffic density estimates. Traffic density estimation from speed measurements, however, is an intrinsically difficult problem. That is, the fundamental diagram only provides a one-to-one correspondence between speed and link density, whereas one may observe various speeds for a given traffic density. The concept of the stochastic fundamental diagram captures this phenomenon by mapping the traffic density to a velocity distribution.

In this chapter, our goal is to use current speed information, in combination with historic information about daily routing profiles, so as to estimate future travel times. These estimates are ultimately used to select the best route between a given origin and destination.
In Section 5.2.1, we review the concept of the stochastic fundamental diagram. Then, Section 5.2.2 provides a problem description of determining routes with minimum expected travel times in highway networks, in the setting that online velocity measurements and historic routing patterns are available.

### 5.2.1 Stochastic fundamental diagram

Consider a network whose underlying graph structure is represented by $G = (N, A)$. The set of nodes $N$ represents the junctions in the network (i.e. intersections and ramps), while the set of directed arcs $A$ represents the roads connecting these junctions. Thus, $(\ell_1, \ell_2) \in A$ only if there is a directed road segment from the junction represented by node $\ell_1$ to the junction represented by node $\ell_2$. In accordance with the notation of the previous chapters, we set $n \equiv |A|$, write $A = \{a_1, \ldots, a_n\}$ for the set of arcs in $G$, with $a_i \equiv (\ell_j, \ell_j')$ for some $\ell_j, \ell_j' \in N$, and let $d_{a_i}$ denote the distance of link $a_i$.

Clearly, for a vehicle traversing link $a \in A$, its velocity, and consequently also the time it takes this vehicle to traverse the link, is affected by the number of vehicles present on the link. As described in Section 5.1, the fundamental diagram of traffic flow captures this relation between the traffic density and the vehicle speeds. Since the seminal linear density-velocity relation suggested by Greenshields [90], numerous studies have developed more detailed diagrams. Notably, these studies are predominantly deterministic in nature, and focus on the expected functional form of the relation. The speeds recorded under a fixed density, however, are subject to various other factors than just the traffic density itself; think of driver-specific behavior and vehicle size. To account for these additional features, we model the relation between traffic density and vehicle speeds via a stochastic fundamental diagram (SFD). Specifically, for each link in the network, we use the SFD notion proposed in [166, 183], whereby the SFD of the link is such that it maps the link’s traffic density (in, say, veh/km) to a velocity distribution.

Let $V_a(k)$ denote the speed at which a vehicle traverses link $a \in A$ whenever the traffic density on link $a$ equals $k$. The proposed SFD is such that $V_a(k)$ is a random variable, the distribution of $V_a(\cdot)$ stemming from a parametric family. We denote the probability density function of $V_a(k)$ by $f(x; \theta_a(k))$, with $\theta_a(k)$ the vector of parameters in case the traffic density on link $a$ equals $k$. As conventional fundamental diagrams are designed to describe the expected density-velocity relationship, the random velocity $V_a(k)$ is modeled such that its mean, $\mu_a(k)$, is described by such a deterministic diagram. To incorporate the variability in the velocity, and to take into account that this variability may depend on the traffic density $k$, we also specify a variance function $\sigma_a^2(k) \geq 0$. The parameter $\theta_a(k)$ is then chosen such that $\mathbb{E}V_a(k) = \mu_a(k)$ and $\text{Var}V_a(k) = \sigma_a^2(k)$. Importantly, to ensure identifiability, we require the mapping $k \mapsto \theta_a(k)$ to be injective: each traffic density $k$ should correspond to a unique $\theta_a(k)$ for all links $a \in A$. Evidently, instead of only specifying functions for the mean and the variance, in principle one may also specify functions that match higher-order moments.

**Example 5.2.1.** Suppose that for each traffic density $k$, $V_a(k) \sim \mathcal{N}(\theta_{a,1}(k), \theta_{a,2}(k))$, i.e., travel velocities are normally distributed. Now, if one wishes to model the speed-density relation to closely resemble the Greenshields model, but additionally aims to incorporate that the travel-time variance is a parabolic function of $k$ (such as in [215]), one could, e.g.,
choose \( \mu_a(k) \) and \( \sigma_a^2(k) \) such that

\[
\begin{align*}
\mu_a(k) &= v_f \cdot \left(1 - \frac{k - k_j}{k_j}\right), & 0 \leq k \leq k_j, \\
\sigma_a^2(k) &= \sigma^2 \cdot \left(-\left(k - \frac{k_j}{2}\right)^2 + \left(\frac{k_j}{2}\right)^2\right), & 0 \leq k \leq k_j,
\end{align*}
\]

with \( \sigma^2 > 0 \), and \( v_f > 0 \) and \( k_j > 0 \) respectively denoting the free-flow speed and jam density. Trivially, we should then choose \( \theta_{a,1}(k) = \mu_a(k) \) and \( \theta_{a,2}(k) = \sigma_a^2(k) \).

Modeling velocities as normally distributed random variables, as in Example 5.2.1, gives rise to some obvious issues, such as the fact that probability mass is assigned to negative velocities, and that the travel time (which is inversely proportional to the speed) does not have a well-defined mean. These complications do not arise in the following two examples, that consider gamma and log-normal distributed velocities.

**Example 5.2.2.** Consider \( \mu_a(k) \) and \( \sigma_a^2(k) \) as in (5.1) and (5.2), respectively, but now suppose that \( V_a(k) \) has a gamma distribution, or \( V_a(k) \sim \Gamma(\theta_{a,1}(k), \theta_{a,2}(k)) \) for a shape parameter \( \theta_{a,1}(k) \) and a scale parameter \( \theta_{a,2}(k) \). Solving

\[
\begin{align*}
\mu_a(k) &= \frac{\theta_{a,1}(k)}{\theta_{a,2}(k)}, & \sigma_a^2(k) &= \frac{\theta_{a,1}(k)}{\theta_{a,2}(k)^2},
\end{align*}
\]

yields

\[
\theta_{a,1}(k) = \frac{\mu^2(k)}{\sigma_a^2(k)}, \quad \theta_{a,2}(k) = \frac{\mu_a(k)}{\sigma_a^2(k)}.
\]

**Example 5.2.3.** Let \( \mu_a(k) \) and \( \sigma_a^2(k) \) be as in (5.1) and (5.2), respectively, but now suppose that \( V(k) \) has a log-normal distribution, i.e., \( V_a(k) \sim \mathcal{N}(\theta_{a,1}(k), \theta_{a,2}(k)) \). Then, solving

\[
\begin{align*}
\mu_a(k) &= e^{\theta_{a,1}(k) + \theta_{a,2}(k)/2}, & \sigma_a^2(k) &= (e^{\theta_{a,2}(k)} - 1) e^{2\theta_{a,1}(k) + \theta_{a,2}(k)} = (e^{\theta_{a,2}(k)} - 1) (\mu_a(k))^2,
\end{align*}
\]

yields

\[
\begin{align*}
\theta_{a,1}(k) &= \log \mu_a(k) - \frac{1}{2} \log \left(\frac{\sigma_a^2(k)}{\mu^2(k)} + 1\right), & \theta_{a,2}(k) &= \log \left(\frac{\sigma_a^2(k)}{\mu^2(k)} + 1\right).
\end{align*}
\]

### 5.2.2 Velocity-based optimal routing

We consider a traveler that, at time \( t = 0 \), wants to travel from their current location to a chosen destination, aiming to limit the time spent in the system. Therefore, our goal is to determine the path that minimizes the expected travel time. Besides their origin-destination (OD) pair and the SFDs that correspond to the different network links, we assume that we have access to both online velocity data and daily routing profiles. The online velocity data consists of pairs of recorded velocities and locations, provided
by, e.g., GPS-measurements. Specifically, for each link $a \in A$, denote the number of velocity measurements on link $a$ at time $t$ by $m_a(t)$, and the corresponding vector of recorded velocities by $v_a(t) = (v_{a,1}(t), \ldots, v_{a,m_a}(t))$. For notational brevity, we will omit the argument of $m_a(t)$ and $v_a(t)$ for $t = 0$.

The daily routing profiles as inferred from historical data reveal the turning proportions at the junctions, and the rate of incoming traffic at these junctions. Specifically, for each junction involving roads indexed by $a_1, \ldots, a_I$, let $p_{a_i a_j}(t)$ be the probability that a vehicle (i.e., one of the other vehicles using the network) that enters the junction on road $a_i$ at time $t$ exits this junction at road $a_j$. At each junction, such a vehicle may also leave the network through a fictional link, denoted by $a_{\text{out}}$. Then, $p_{a_i a_{\text{out}}}(t)$ denotes the probability that a vehicle that enters the junction on road $i$ at time $t$ will leave the network. Moreover, the rate at which new traffic enters the network at arc $a$ (via a fictional link $a_{\text{in}}$) is denoted by $\lambda_a(t)$.

Evidently, due to the intrinsic fluctuations in the speeds the vehicles in the network drive at, optimal routing is highly non-trivial. That is, to compute expected travel times, we need to get a handle on the current and future traffic densities. Estimating current densities is a delicate task, as there is no one-to-one correspondence between the speeds that vehicles drive and the corresponding link density. Predicting future densities is complex as well, as the vehicles move through the network with random speed. In the next section, we tackle these challenges by combining two well-known phenomena. We first propose a maximum likelihood estimator for the prediction of the current traffic densities (based on speed measurements), and then we model (with the obtained densities as initial values) the traffic flows according to a discrete counterpart of the average dynamics of the SFD.

### 5.3 Traffic density estimation

Now that we have outlined our objectives, we move on to the estimation of the travel-time densities, both at the current time ($t = 0$, Section 5.3.1) and at future time instances ($t > 0$, Section 5.3.2), by which we can determine current and future expected travel times. Having constructed the SFD as in Section 5.2, it suffices to obtain an estimate for the traffic densities at each $t \geq 0$, as the travel-time distribution can be inferred from these densities.

#### 5.3.1 Current densities

For each link $a$, we have access to an independent sample $v_{a,1}, \ldots, v_{a,m_a}$ of velocities from the GPS data. As the sample is observed in real-time, each velocity can be seen as a realization of the random variable $V_a(k)$ with density function $f(a; \theta_a(k))$. To estimate the current density on link $a$, we propose to use the likelihood of $k$ under the sample $v_{a,1}, \ldots, v_{a,m_a}$:

$$L(\theta_a(k); v_{a,1}, \ldots, v_{a,m_a}) = \prod_{i=1}^{m_a} f(v_{a,i}; \theta_a(k)).$$

The maximizer $\hat{k}_a$ in the range $[m_a/d_a, \infty)$ is the corresponding maximum likelihood estimate (MLE), which provides us with $f(x; \theta_a(\hat{k}_a))$, a prediction for the density function of
Figure 5.1: Difference between the average error (averaged over 1000 Monte Carlo runs) of the mean density estimate and the MLE, for different values of $\sigma^2$ and $m$ (i.e., the sample size), whenever $k = 55 \text{ veh/km}$, $v_f = 110 \text{ km/h}$, $k_j = 110 \text{ veh/km}$, and $V(k)$ has a log-normal distribution whose mean is defined by the fundamental diagram of Greenshields (left, see, e.g., Example 5.2.1) or Smulders (right, see, e.g., [196], with the flow capacity chosen to equal 2241 veh/h), and the variance formula is given through (5.2).

the velocities on link $a$. Lastly, in case $m_a = 0$, we will set $\hat{k}_a = 0$.

Example 5.3.1. Suppose that the SFD is given as in Example 5.2.3, and that for each link $a \in A$ we have access to a sample $v_{a,1}, \ldots, v_{a,m_a}$. A prediction for the density on link $a$ is given by the $k \in [m_a/d_a, \infty)$ that maximizes the likelihood

$$L(\theta_{a,1}(k), \theta_{a,2}(k); v_{a,1}, \ldots, v_{a,m_a}) = \prod_{i=1}^{m_a} \frac{1}{v_{a,i} \sqrt{2\theta_{a,2}(k)\pi}} e^{-\frac{1}{2\theta_{a,2}(k)}(\ln(v_{a,i}) - \theta_{a,1}(k))^2},$$

or, equivalently, the log-likelihood

$$\log L(\theta_{a,1}(k), \theta_{a,2}(k); v_{a,1}, \ldots, v_{a,m_a})$$

$$= -\frac{1}{2} m_a \log 2\pi - \sum_{i=1}^{m_a} \left( \log v_{a,i} + \frac{1}{2} \log \theta_{a,2}(k) + \frac{1}{2\theta_{a,2}(k)}(\ln(v_{a,i}) - \theta_{a,1}(k))^2 \right),$$

where $\theta_{a,1}(k)$ and $\theta_{a,2}(k)$ are as in (5.3).

There are of course alternative ways to estimate the current density on a link from the sample of velocity measurements. A naive way would be to pick the density $k$ for which $E[V_a(k)] = \bar{v}$, $\bar{v}$ being the average of the sample. However, this method typically performs poorly in case the velocity distributions are skewed or the sample size is small. This can also be observed in Figure 5.1, which shows the difference in estimation error between this naive method and the likelihood method that we propose, for different sample sizes and parameters, and for two different fundamental diagrams.

Remark 5.3.2. Each vehicle whose velocity data becomes available at time $t = 0$ has typically been in the system before this time instant, and has thus also provided measurements at earlier timestamps. However, we choose to not include these in the estimation procedure, as this will give rise to some obvious correlation issues. Importantly, one cannot simply replace the velocity of the last timestamp by a weighted average of the velocities of the last few time steps, as such an average does no longer represent a sample of $V_a(k)$ for some $k \in \mathbb{R}_{\geq 0}$.

\[\diamond\]
5.3. Traffic density estimation

Remark 5.3.3. If one has distributional information, or a distributional belief, about the density $k$ at time 0, the procedure can be extended to a Bayesian approach. Specifically, if $K_a(t)$ is the vehicle density on link $a$ at time $t$, and $g_a(k) \equiv \mathbb{P}(K_a(t) \in dk)$, one may use the value of $k$ that maximizes the following likelihood function as prediction for the density:

$$L(k; v_{a,1}, \ldots, v_{a,m_a}) = \frac{g_a(k) \prod_{i=1}^{m_a} f(v_{a,i}; \theta_a(k))}{\int_0^\infty g_a(\ell) \prod_{i=1}^{m_a} f(v_{a,i}; \theta_a(\ell)) d\ell},$$

for $t \geq 0$. \hfill $\diamond$

For each link $a$, the estimate $\hat{k}_a$ is a prediction of the current traffic density on this link. For our routing objectives, we also wish to infer future traffic densities; this is the topic of the next subsection.

5.3.2 Future densities

To obtain estimates of future traffic densities, we discretize time into steps of size $\delta > 0$. Specifically, with $t_i \equiv \delta \cdot i$, we propose to predict the traffic densities at times $t = t_0, t_1, t_2, \ldots$. Moreover, we partition each link $a_j \in \mathcal{A}$ into sublinks or cells $a_{j,1}, \ldots, a_{j,n_j}$, and denote the total set of network cells by $\mathcal{A}$. For $a \in \mathcal{A}$, let $\hat{k}_a(t_i)$ denote the estimate for the traffic density on link $a$ at time $t_i$. As we only have access to an estimate of the traffic density, we will always assume that vehicles are evenly spread over the link.

Our procedure will be of an iterative nature: for any $i \geq 1$ it produces an estimate $\hat{k}_{a_i}(t_{i+1})$ using the set $\{\hat{k}_{a'}(t_i) ; a' \in \mathcal{A}\}$. It is based on the well-known cell transmission model (CTM), as initially presented by Daganzo [45, 46], which corresponds to a discretized counterpart of traffic flow dynamics adhering a trapezoid-shaped fundamental diagram. However, as extensively argued in [141], this idea can be extended to construct a model that is consistent with any fundamental diagram (that is, not necessarily trapezoid-shaped). The underlying idea is that for each time step, we first determine the flow between each pair of cells, and then use these flows to update the traffic densities in each cell by shifting time from $t_i$ to $t_{i+1}$.

For any $a \in \mathcal{A}$ we let $A^+(a)$ and $A^-(a)$ denote the set of downstream and upstream cells of $a$, respectively. Note that these sets $A^+(a)$ and $A^-(a)$ may also include (fictional) links via which vehicles leave and enter the network, respectively. To be able to directly use the CTM of [45], we require that $|A^+(a)| \leq 2$ and $|A^-(a)| \leq 2$ for any $a \in \mathcal{A}$. Notably, this assumption is realistic as we consider highway networks, in which on- and off-ramps are typically separated (albeit sometimes by a small distance); an example of such a network is the Amsterdam highway network, which is implemented in Section 5.5.

We let $k_{a,c}$ denote the critical density of arc $a \in \mathcal{A}$, which is the traffic density at which the maximum flow is obtained. If $k_a(t_i) > k_{a,c}$, we say that arc $a$ is in a congested state; otherwise, if $k_a(t_i) \leq k_{a,c}$, we say that arc $a$ is in an uncongested state. Then, the demand function $D_a$ and supply function $S_a$ of arc $a$ (representing, for a single time step, the amount of traffic that wants to enter arc $a$ and the maximum amount of new traffic that
is accepted onto $a$, respectively) are given by

$$D_a(t_i) = \begin{cases} k_a(t_i) \cdot \mu_a(k_a(t_i)) & \text{if } k_a(t_i) \leq k_{a,c}, \\ k_{a,c} \cdot \mu_a(k_{a,c}) & \text{if } k_a(t_i) > k_{a,c}, \end{cases}$$

and

$$S_a(t_i) = \begin{cases} k_{a,c} \cdot \mu_a(k_{a,c}) & \text{if } k_a(t_i) \leq k_{a,c}, \\ k_a(t_i) \cdot \mu_a(k_a(t_i)) & \text{if } k_a(t_i) > k_{a,c}. \end{cases}$$

Lastly, we let the flows on the links from which vehicles may enter the network be ‘exogenously determined’. This concretely means that $D_{ata}(t_i) = \lambda_a(t_i)$. For links from which vehicles leave the network, we use the convention that $S_{ata}(t_i) = \infty$.

Using the terminology in [45], without loss of generality, it is assumed that each $a \in \tilde{A}$ belongs to exactly one of the following types: an ordinary arc, part of a diverge, or part of a merge. We now specify the flow for each of these cases.

1. **Ordinary arcs.** Let $a \in \tilde{A}$ be an ordinary arc and write $a'$ for the downstream arc of $a$. If $a'$ is in an uncongested state, then the flow on $a$ is determined by the demand of link $a$. Otherwise, in a congested state, the flow on $a$ is determined by the supply of $a'$. As a consequence, the flow on arc $a$ is given by

$$q_a(t_i) = \min\{D_a(t_i), S_{a'}(t_i)\}. \quad (5.4)$$

2. **Diverge.** Let $a \in \tilde{A}$ be part of a diverge and write $a'$ and $a''$ for the two downstream arcs of $a$. Following [45], we let the turning proportions, which are given, be denoted by $p_{a'a'}(t)$ and $p_{a''a'}(t)$, with $p_{a'a'}(t) + p_{a''a'}(t) = 1$. The flow on arc $a$ is then

$$q_a(t_i) = \min \left\{ D_a(t_i), \frac{S_{a'}(t_i)}{p_{a'a'}(t)} \cdot \frac{S_{a''}(t_i)}{p_{a''a'}(t)} \right\}.$$

3. **Merge.** Let $a \in \tilde{A}$ be part of a merge and write $a'$ for the downstream arc of $a$. Here, arc $a'$ also has an upstream arc other than $a$, say $a_c$. In case not all vehicles on arcs $a$ and $a_c$ can flow to arc $a'$ in one time step, we need to specify the priority fractions $\pi_a$ and $\pi_{a_c}$ of vehicles that flow from $a$ and $a_c$ to $a'$, respectively. Again, we let these fractions be given. To determine the flow on arc $a$, we need to distinguish between three regimes:

- ‘forward’ regime, where $D_a(t_i) \leq \pi_a S_{a'}(t_i)$. In this case, the demand of link $a$ can be fulfilled, so that $q_a(t_i) = D_a(t_i)$.

- ‘backward’ regime, where both $D_a(t_i) \geq \pi_a S_{a'}(t_i)$ and $D_{a_c}(t_i) \geq \pi_{a_c} S_{a'}(t_i)$. Neither the demand of $a$ nor of $a'$ can be fulfilled, and hence $q_a(t_i) = \pi_a S_{a'}(t_i)$.

- ‘mixed’ regime, in which $D_a(t_i) \geq \pi_a S_{a'}(t_i)$ but $D_{a_c}(t_i) \leq \pi_{a_c} S_{a'}(t_i)$. The allocated supply for $a$ is not sufficient to fulfill its demand, but it is sufficient for $a_c$, giving $q_a(t_i) = \min\{D_a(t_i), S_{a'}(t_i) - D_{a_c}(t_i)\}$. 


Now that we have obtained the flow on each arc, we may proceed to updating the traffic density on each arc. With the suggested procedure, during each time interval \((t_i, t_{i+1})\), the number of vehicles that leave and enter link \(a\) are respectively given by

\[
B_{a,in}(t_i) \equiv \delta \cdot \sum_{a' \in \mathcal{A}^- (a)} p_{a'a}(t_i) q_{a'}(t_i) \quad \text{and} \quad B_{a,out}(t_i) \equiv \delta \cdot q_a(t_i).
\]

With \(d_a\) denoting the length of link \(a\), the number of vehicles on arc \(a\) at time \(t_i\) is \(d_a k_a(t_i)\), so that finally the density on link \(a\) on time \(t_{i+1}\) is given by

\[
k_a(t_{i+1}) = k_a(t_i) + \frac{B_{a,in}(t_i) - B_{a,out}(t_i)}{d_a}.
\] (5.5)

As our outlined estimation procedure provides us with an estimate for \(k_a(t_0)\), the estimates for future traffic densities \(k_a(t_i)\) for \(i = 1, 2, 3, \ldots\) simply follow by iteratively applying (5.5).

### 5.4 Optimal routing advice

Bearing in mind our ultimate goal to develop a real-time procedure to identify the route with the lowest expected travel time, Section 5.4.1 first explains how one can evaluate, relying on the traffic density estimates proposed in the previous section, the expected travel time for traversing a network link at current and future time points. Then, Section 5.4.2 describes a highly efficient (in terms of run-time) routing mechanism, that is based on the \(A^*\) algorithm. Section 5.5 includes various numerical examples that assess the performance of this procedure.

#### 5.4.1 Expected travel times

The traffic density estimates of the preceding section, together with the SFDs, can be used to infer the travel-time distributions in the network. To see this, note that a proxy for the random travel time on link \(a\) at time \(t_i\) is \(d_a k_a(t_i)\), whose probability density function we can infer by \(f(x; \theta_a(k_a(t_i)))\). It should be noted that the proxy assumes that the link \(a\) is traversed at constant speed \(v\), a realization of \(V_a(k_a(t_i))\), whereas in reality, while traversing the arc, \(k_a(t_i)\) – and thus the velocity distribution – may change. However, as both the time steps and arc lengths are chosen small, it can be argued that the proposed proxy will have a sufficiently small error. Moreover, a favorable consequence of working with this proxy is the fact that for a set of distributions the corresponding travel-time distribution can be determined explicitly, as pointed out in the next example.

**Example 5.4.1.** An inverse-gamma distributed random variable has the distribution of the reciprocal of a gamma distributed random variable. Suppose that \(V_a(k_a(t_i)) \sim \)
Algorithm 5.1: Time-dependent shortest path algorithm.

**Result:** path from $t_0$ to $t^*$.

Initialization: $D_{t_0} = 0, D_t = \infty$ for $\ell \neq t_0$,

$H = \{(D_{t_0} + L_{t_0}(\text{key}), D_{t_0}, t_0, \{t_0\} \ (\text{path})\}, \mathcal{V} = \emptyset$

**while** $H$ non-empty **do**

1. Extract tuple $(D_\ell + L_\ell(\text{key}), D_\ell, \ell, \text{path})$ with minimum first entry (key) from $H$;
2. **if** $\ell = t^*$ quit and **return** path, **else if** $\ell \in \mathcal{V}$ go back to step 1. **else** continue;
3. **for** neighbor $\ell'$ of $\ell$ not in $\mathcal{V}$ **do**
   a. Set $d_\ell' = D_\ell' + c_a(D_\ell')$ for $a = (\ell, \ell')$;
   b. **if** $d_\ell' < D_\ell'$ set $D_\ell' = d_\ell'$ and insert $(D_{\ell'} + L_{\ell'}(\text{key}), D_{\ell'}, \ell', \text{path} + \{\ell'\})$ in $H$;
4. Add $\ell$ to $\mathcal{V}$

$\Gamma(\theta_{a,1}(k_a(t_i)), \theta_{a,2}(k_a(t_i)))$, as in Example 5.2.2. The travel-time proxy on link $a$ at time $t_i$ then satisfies, in self-evident notation,

$$\tilde{T}_a(k_a(t_i)) \sim \Gamma_{\text{inv}} \left( \theta_{a,1}(k_a(t_i)), \frac{\theta_{a,2}(k_a(t_i))}{d_a} \right).$$

Alternatively, if $V_a(k) \sim \log \mathcal{N}(\theta_{a,1}(k), \theta_{a,2}(k))$ as in Example 5.2.3, the travel-time proxy on link $a$ at time $t_i$ is such that

$$\tilde{T}_a(k_a(t_i)) \sim \log \mathcal{N}(-\theta_{a,1}(k_a(t_i)) + \ln(d_a), \theta_{a,2}(k_a(t_i))).$$

recall that the reciprocal of a log-normal random variable is again log-normal. \hfill \diamond

Finally, in order to obtain an estimate $c_a(t_i)$ for the expected travel time on arc $a$ at time $t_i$, we substitute the estimate $\hat{k}_a(t_i)$ for the unknown $k_a(t_i)$ in the expected value of the proxy, i.e., we use

$$c_a(t_i) = \mathbb{E} \left[ \tilde{T}_a(\hat{k}_a(t_i)) \right]. \quad (5.6)$$

### 5.4.2 Efficient and near-optimal routing

Now that we have obtained estimates for the expected travel times at different arcs and timestamps, we proceed by determining the optimal route for an individual traveler. In doing so, we again distinguish between an offline and an online routing setting. In the offline setting, the driver does not deviate from the path they have chosen upon departure. In the online setting, the driver keeps receiving density updates, and may use these to update their route while driving.

#### Offline routing

In the offline context, we propose to determine the optimal path between the vehicle origin and a given destination, in terms of minimum expected travel time, by Algorithm 5.1. This is a time-dependent A* algorithm, the weight $c_a(t)$ of arc $a$ at time $t \in [t_i, t_{i+1})$ given by a linear interpolation of $c_a(t_i)$ and $c_a(t_{i+1})$:

$$c_a(t) = c_a(t_i) \frac{t_{i+1} - t}{t_{i+1} - t_i} + c_a(t_{i+1}) \frac{t - t_i}{t_{i+1} - t_i}. \quad (5.7)$$
Algorithm 5.2: Online routing policy.

**Result:** procedure to travel from \( \ell_0 \) to \( \ell^* \).

**Initialization:** \( \ell' = \ell_0, t' = 0 \);

**while** \( \ell' \neq \ell^* \) **do**

1. Obtain latest velocity measurements, set \( t_i = t' + \delta_i \) \((i = 0, 1, 2, \ldots)\);
2. Determine \( \hat{k}_a(t_i) \) for all \( a \in \tilde{A} \) and \( i = 0, 1, 2, \ldots, M_{\ell'} \);
3. Use (5.6) and (5.7) to find \( c_a(t) \) for all \( t' \leq t \leq t_{M_\ell} \);
4. Let \( \ell' \) equal first node on path outputted by Algorithm 5.1. Travel to \( \ell' \), and let \( t' \) equal arrival time at \( \ell' \).

Recall that this algorithm is a label-correcting algorithm in the same spirit as Dijkstra’s algorithm, but with the additional property that it adds a lower bound for the travel time between nodes and the destination to the label [58, 123]. As in the previous chapters, we let the lower bound \( \text{LB}_k \) of a node \( k \) equal the minimum distance (in km) to the destination divided by the corresponding maximum driveable speed.

Besides the fact that the expected travel times at the different timestamps are proxies, there is an additional reason for the fact that the proposed algorithm is not exact. That is, recall that in the time-dependent setting, label-correcting algorithms such as the Dijkstra algorithm are only exact in case the weights \( \{c_a(t) | a \in \tilde{A}, t \geq 0 \} \) satisfy the FIFO property: the weights should be such that whenever the traveler decides to traverse link \( a \in \tilde{A} \) at an earlier time, they have a lower expected arrival time at the end of the link. In the considered algorithm, this entails that it should hold that for any \( t, t' \geq 0 \) for which \( t \leq t' \),

\[
t + c_a(t) \leq t' + c_a(t').
\]

Notably, for our procedure, such an ordering is not guaranteed. However, by the continuity of the traffic streams, the underlying model does satisfy the FIFO property. As the weights are expected to be close to their continuous counterparts, the loss that is incurred due to the potentially non-consistent ordering will be negligible. On the other hand, the gains of accepting such a small loss are huge, in that it allows us to use the fast A* algorithm. Specifically, by the use of efficient storing structures (such as a so-called Fibonacci heap), the algorithm has worst-case complexity \( \Theta (|A| + |N| \log |N|) \) [73].

**Online routing**

In the online setting, the driver receives new velocity recordings while driving. These may for instance reveal that it is substantially busier on the chosen path than anticipated, making it preferable to switch to an alternative route. Thus, at each location where a driver is able to make a new routing choice (i.e., at each node), they need to determine the best direction to travel, based on the latest available velocity measurements.

The online routing policy we propose is summarized in Algorithm 5.2, and is in the same spirit as the routing algorithms proposed in Chapter 2, in the sense that it can be seen as the ‘dynamic implementation’ of a shortest path algorithm. That is, whenever the vehicle arrives at a node \( \ell' \in \mathcal{N} \) at time \( t' \), the most up-to-date velocity measurements are used to
compute the density estimates at times $t_0, t_1, \ldots, t_{M'}$, with $t_i \equiv t' + \delta \cdot i$ ($i = 0, 1, 2, \ldots$), and $M'$ such that $M'\delta$ is a crude upper bound for the travel time between $\ell'$ and the destination. Then, these estimates are used to infer, for any $a \in \tilde{A}$, the expected travel times $c_a(t_i)$ ($i = 0, 1, \ldots, M'$), and, via linear interpolation, the expected travel time $c_a(t)$ for any $t' \leq t \leq t_{M'}$. Now, an application of Algorithm 5.1 yields a shortest path, and our policy prescribes to travel to the first node in this path. At each node the vehicle arrives at, the above procedure is repeated (always using the most recent velocity measurements), until they arrive at their chosen destination.

The suggested procedure is highly efficient, in that it is capable of providing real-time advice in large networks. We will corroborate this claim in the large-scale example featuring in the upcoming section. Notably, there may be alternative procedures that achieve better performance, as they may, for example, take into account that the driver knows that they will receive updated velocity measurements and can adapt the route, giving rise to dynamic programming type of algorithms. Constructing such procedures is a highly non-trivial task, as this requires the computation of – and working with – probability distributions over the continuous space of potentially recorded velocities at future time-stamps. A perhaps more important complication, that was also highlighted in Chapter 2, is that, due to the curse of dimensionality, the high computational complexity of such procedures renders their application in large-scale networks unlikely.

### 5.5 Numerical experiments

This section consists of numerical experiments in which we illustrate the applicability of our procedure. Specifically, we review the accuracy and robustness of the components of our procedure (Sections 5.5.1 and 5.5.2), focus on its properties in a routing context (Section 5.5.3), and study the efficiency of the procedure (Section 5.5.4).

In what follows we will, unless otherwise specified, model the flow-density relation on each (sub)link by the trapezoid-shaped fundamental diagram as used in [45], with the addition of a random noise term to the velocity which is parameterized as (5.2). In particular, we model the velocities as log-normal random variables with parameters such that its expectation and variance are, for $k \in [0, k_j]$, given by

$$
\mu_a(k) = \frac{1}{k} \min\{v_f \cdot k, q_c, v_f \cdot (k_j - k)\} \quad \text{and} \quad \sigma_a^2(k) = 0.2 \cdot \left( -\left( k - \frac{k_j}{2} \right)^2 + \left( \frac{k_j}{2} \right)^2 \right).
$$
5.5. Numerical experiments

Figure 5.3: The traffic input rates considered in the numerical experiments, where $T = 0.1$ h in Section 5.5.1, and $T = 1$ h otherwise.

For the traffic parameters we follow Smulders' choice for a Dutch motorway [196]: $v_f = 110$ km/h, $k_j = 110$ veh/km, and $q_c = 2241$ veh/h. See Figure 5.2a for the resulting deterministic flow-density relation. In some experiments, we additionally consider $\mu_a(k)$ to be shaped like De Romph’s fundamental diagram [188] as depicted in Figure 5.2b.

5.5.1 Selecting an appropriate granularity

Our routing procedure requires access to estimates of the future traffic densities, for which we rely on a discretization of both space and time. Following [45], the discretization should be chosen such that in light traffic conditions, each cell depletes within precisely one time step. We now proceed with establishing an appropriate discretization for space, from which the discretization of time follows.

For this, we consider a simple network consisting of a single link with a length of 10 km, and with vehicles exclusively entering and leaving the network at the beginning and end of the link, respectively. At $t = 0$, there are in total 200 evenly spread vehicles, so that $q(0) = 20$ veh/km. We consider various time-dependent rates at which vehicles enter the network, see Figure 5.3. Then, even under the deterministic flow-density dynamics, vehicles will not be evenly spread over the link at future time instances. This, in turn, may greatly impact the expected travel time. Therefore, it is important that the granularity of space is chosen sufficiently high such that the spread of vehicles can be modeled in adequate detail.

In Figure 5.4, we plot the average traffic density in the network at time $t = 0.1$ h as a function of the number of sublinks into which we divide the original link. We see that for any input rate, the average future traffic density already approximates its limit whenever the number of sublinks exceeds 20, or, whenever the length of a sublink is shorter than 500 m. We observe a similar phenomenon for the expected travel time of a vehicle departing at time $t = 0.05$ h. Besides Daganzo’s fundamental diagram as depicted in Figure 5.2a, we also studied the behavior under De Romph’s fundamental diagram as depicted in Figure 5.2b. This again led to similar results, indicating that the convergence is irrespective of the fundamental diagram under consideration. As such, in the following experiments we will always set the length of the sublinks to 500 m.
5.4.2 Comparison of the procedure to simulation

The estimated future traffic densities in Section 5.5.1 were obtained using the simplified procedure from Section 5.3.2. We now proceed by displaying the robustness of this procedure by comparing the estimates to future traffic densities that result from a simulation that more accurately mimics the uncertain conditions in a real traffic network.

The main difference in the simulation compared to our simplified procedure is that the randomness of the velocities is not disregarded. That is, instead of assuming that the velocities in the network during the time interval \((t_i, t_{i+1})\) are deterministic and given by the mean of the velocity distribution, we will generate velocity realizations for each individual vehicle in the network.

Specifically, we discretize time such that \(\delta = t_{i+1} - t_i = 0.5/110\) h (i.e., the time it takes a vehicle to traverse 500 m under free-flow speed), and at each time instance \(t_i\) and for each vehicle, we generate a velocity based on the traffic density on the road segment located 500 meters ahead of the vehicle under consideration. This vehicle specific generated velocity is assumed fixed in the time interval \((t_i, t_{i+1})\), and the locations of the vehicles in the network are updated according to their generated velocity.

We take a similar approach for vehicles entering the network. Whereas the simplified procedure assumes that at time \(t_i\) vehicles enter the network with a deterministic rate \(\lambda(t_i)\), in the simulation we generate the number of vehicles that want to enter the network at time \(t_i\) from a Poisson distribution with rate \(\lambda(t_i) \cdot \delta\). The actual number of vehicles that enter the link at time \(t_i\) then follows by rounding \(\delta \cdot q(t_i)\) to the nearest integer, where the flow \(q(t_i)\) is as in (5.4) with \(D(t_i) = \text{Poisson}(\lambda(t_i) \cdot \delta)/\delta\), and \(S(t_i)\) is the supply corresponding to the first 500 m of the road segment.

Lastly, recall that in Section 5.5.1 we assumed that vehicles on the link were evenly spread at time \(t = 0\). Now, we introduce randomness in this initial spread of the vehicles on the
5.5. Numerical experiments

link by drawing the initial location of each vehicle from a specified probability distribution.
We will take this approach both for the simplified procedure and for the simulation. For
the comparison of our simplified procedure to the simulation we consider two settings,
which we label as full information and partial information.

**Full information**

In the full information case, we assume that $k_{a}(0)$, the traffic density at time $t = 0$, is
known for each arc. Figure 5.5 plots the evolution of the traffic density for the same
network as considered in Section 5.5.1, both for our simplified procedure (blue) and the
simulation (yellow), where the traffic densities of the simulation are averages based on
1000 Monte Carlo runs. Each row represents a different initial placement of vehicles.
Specifically, we generate 50 draws from Uniform(0, 10) (row 1, labeled $k_{a}^{1}(0)$), 200 draws
from Uniform(0, 10) (row 2, labeled $k_{a}^{2}(0)$), and 200 draws from Beta(0, 10, 5, 5) (row 3,
labeled $k_{a}^{3}(0)$, representing a setting in which there is a high traffic density midway). The
columns correspond to the different input rates depicted in Figure 5.3.

We see that the traffic densities that follow from our procedure closely resemble the traffic
densities of the simulation. We do, however, observe a slight bias towards higher traffic
densities in our procedure. This small bias can also be seen in Table 5.1, which reports
the excess expected travel time of our procedure compared to the simulation. As the
expected travel time is no less than 5.50 min, we regard the relative error of our procedure
as acceptable. Moreover, as this bias will be present in every candidate path we consider,

**Table 5.1:** Excess expected travel time of our procedure compared to the simulation (in min.).

<table>
<thead>
<tr>
<th>$k_{a}(0)$</th>
<th>$\lambda_{1}(t)$</th>
<th>$\lambda_{2}(t)$</th>
<th>$\lambda_{3}(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{a}^{1}(0)$</td>
<td>0.05</td>
<td>0.03</td>
<td>0.07</td>
</tr>
<tr>
<td>$k_{a}^{2}(0)$</td>
<td>0.06</td>
<td>-0.07</td>
<td>0.08</td>
</tr>
<tr>
<td>$k_{a}^{3}(0)$</td>
<td>0.36</td>
<td>0.24</td>
<td>0.38</td>
</tr>
</tbody>
</table>
it is unlikely to impact the optimality of the routes.

Partial information

In the partial information case, we only have access to velocity observations of a selection of vehicles in the network. Using these observations, we apply our estimation procedure from Section 5.3 to obtain an estimate of the traffic density at time 0 for each arc. Using this initial density estimate, our simplified procedure then supplies us with estimates for future traffic densities, from which we obtain the expected travel time.

Specifically, we first generate an initial placement of a set amount of vehicles, and we compute the traffic density in each cell. Then, we generate a velocity for each vehicle based on the traffic density of their respective cell. Figure 5.6 depicts the expected travel time for different random sample sizes of the velocities, where the expected travel time is averaged over 1000 Monte Carlo runs, and the dotted lines depict an interval containing 90% of the Monte Carlo runs. In this figure we consider two different settings: vehicle placements from 50 draws from Uniform(0, 10), input rate $\lambda_1(t)$, and Daganzo’s fundamental diagram (left), and 400 draws from Uniform(0, 10), input rate $\lambda_3(t)$, and De Romph’s fundamental diagram (right).

In both settings, we observe from the dotted lines that the vast majority of the Monte Carlo runs result in acceptable expected travel time estimates, even for low sample sizes. We do see that the Monte Carlo runs display a higher variation for similar sample sizes in the right figure. This is due to the higher traffic density in this setting, which results in higher variances of the velocities. We also observe that the expected travel time is slightly increasing in the number of observations. This is due to the occasional absence of observations on some sublinks, resulting in a (sometimes faulty) traffic density estimate of 0 veh/km, which in turn results in a biased estimate for the expected travel time. As the sample size grows, this phenomenon becomes rarer, and we see that the bias quickly disappears.

5.5.3 Routing experiments

In the above experiments, we have seen that our procedure is able to translate online velocity measurements into travel-time estimates that properly match the expected traffic conditions. This enables routing algorithms to identify the path with the lowest expected
5.5. Numerical experiments

For illustrative purposes, we consider the simple graph of Figure 5.7, which consists of six nodes and six edges. Node 1 is the only junction vehicles may arrive at, and node 6 is the only junction from which vehicles may exit the system. The traffic dynamics on all roads are chosen to match those introduced at the start of this section, i.e., a trapezoidal shaped fundamental gram, velocities adhering a log-normal distribution, and traffic parameters that align those reported by Smulders [196]. First, let the turning proportions at the junction be equal for both routing options, and, analogously, let the priority fractions for the merging links be the same. Moreover, consider a non-rush hour setting, in which the input rate is 400 veh/h, and it is uncrowded at $t = 0$. Specifically, there are 50 vehicles in the network, whose locations are sampled uniformly, and whose speeds are sampled from the SFD (the input density corresponding to the density of the cell in which the vehicle is located). The input of the procedure is a random sample of velocity measurements, whose size equals 10% of the total number of measurements.

Now, let there be a vehicle that aims to travel from node 1 to node 6. Given that both the current traffic densities and expected input is low, a routing algorithm should identify the route with shortest length as optimal path. Indeed, as can be seen in Table 5.2 (‘case 1’), our procedure (‘SFD’) picks the lower, shortest path for traveling between nodes 1 and 6. Of course, this is not surprising, as the naïve procedure that simply assumes the current traffic state to be static will yield identical routing advice, as is also reported in Table 5.2.

A similar observation is true whenever the initial densities on links (2, 4) and (4, 5) are 100 veh/km, and thus close to jam density (‘case 2’), and it should thus be preferred to select the somewhat longer but certainly faster upper path via node 3.

Figure 5.7: Simple road network example.

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Expected travel time (in minutes) for the path \{1, 2, 4, 5, 6\} (resp. \{1, 2, 3, 5, 6\}) in Figure 5.7, denoted by ‘L’ (resp. ‘U’), as computed by our proposed procedure (denoted ‘SFD’) and a naïve procedure that does not take the changing dynamics into account, in four different traffic settings. The expected travel time that corresponds to the path that each procedure in each setting outputs, is marked bold.

travel time for a vehicle that aims to travel between its origin and a given destination. In this experiment, we investigate the paths and corresponding expected travel times that are selected by the algorithm under different traffic conditions. It is shown that the procedure performs as expected, in the sense that it selects a path such that it is likely that the vehicle faces favorable conditions, resulting in a lower expected travel time. By considering path-selecting procedures, our attention is directed towards the offline routing setting. However, we note that, naturally, we could construct similar examples for the online routing setting.
However, there are also settings in which our method, taking the changing vehicle dynamics into account, outperforms the more naïve procedure. A simple example of such a setting (‘case 3’) is the following: let the conditions equal those of case 1, except for the fact that the initial density on link (4, 5) is 80 veh/km. Then, even though there are unfavorable traffic conditions on the lower path between nodes 1 and 6, it is still optimal to pick this path: with low upstream traffic densities, at the time the vehicle would reach link (4, 5) when traversing the lower path, this congestive situation has already dissipated. Our methodology does indeed acknowledge this, and selects the shortest path, whereas the naïve procedure does not account for these traffic dynamics. Another example (‘case 4’) is when there are high traffic densities (i.e., 80 veh/km) on both links (3, 5) and (4, 5), but the priority fractions are 0.95 and 0.05 respectively. The naïve method does not account for these fractions, and would therefore still pick the shorter path, whereas our method observes that the dissipation on the upper route occurs substantially faster, this path therefore being more favorable. Notably, as can be observed from the expected travel time measured by our procedure, the gains of using our method compared to the more naïve procedure are in this small case already 1.9 minutes. We conclude that our procedure indeed successfully takes the expected changes in traffic densities into account, resulting in a reduction of the expected travel time compared to the use of a more naïve procedure.

5.5.4 Efficiency

To test the scalability of our procedures, we consider routing in graphs of different sizes. Specifically, we look at the run-time of the proposed methodology in linear graphs, i.e., graphs in which there is a single path from some starting link to some final link, every other link in the network being part of this path. The model setting corresponds to the one provided at the start of this section, with links that have a length of 500 meter. It should be noted that, whereas there is only one path from start to finish, such that the optimal path is trivial, the graph does serve as proper test for the run-time of the procedure. That is, given that the routing algorithm has to visit all nodes in this graph, its costs equal the worst-case complexity of any graph with the same number of nodes and edges, but a different structure. Thus, these costs provide an upper bound for the costs the algorithm would have in a real-world network of similar size.

The costs of the complete procedure, for linear networks of different total lengths, are given in Table 5.3. In addition, to corroborate the claim that the costs in these linear networks serve as upper bound, we report that computing an optimal route from the most western to the most eastern node in the Amsterdam highway system (Figure 5.8), in which the link lengths sum to 210.6 km, takes 2.61 seconds. These costs reveal that the proposed methodology has a high potential in practical applications, its run-time being real-time. It should be remarked that the computational costs are for a large extent due to the first part of the procedure: the estimation of the initial densities. The main reason is the fact that, to this end, one needs to perform a numerical optimization for each link in the graph. In our implementation, we only made use of a single computer kernel. Since the optimization

<table>
<thead>
<tr>
<th>Length (in km.)</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run time (in sec.)</td>
<td>0.15</td>
<td>0.23</td>
<td>0.35</td>
<td>0.56</td>
<td>0.87</td>
<td>1.45</td>
<td>2.13</td>
<td>3.36</td>
</tr>
</tbody>
</table>

Table 5.3: Efficiency of the routing procedure.
5.6 Concluding remarks

In this chapter, we have developed a procedure that aims to find the route with minimum expected travel time from a vehicle's origin to a chosen destination, thereby only having access to (i) real-time vectors of velocity measurements on the different network links and (ii) historic routing probabilities. The underlying road traffic model captures the randomness in vehicle speeds under given traffic densities with a stochastic fundamental diagram (SFD), which maps each traffic density to a velocity distribution. Various numerical experiments illustrated the applicability of our procedure. In particular, we have shown that our procedure indeed successfully takes the expected changes in traffic densities into account, resulting in a reduction of the expected travel time. Moreover, we have demonstrated its high potential in practical applications, given that its run-time is real-time.

In the presented SFD setup, we have not distinguished between vehicle types. However, in the literature, there are some suggestions for multi-class fundamental diagrams, see, e.g., [146, 152, 221]. Therefore, future research could focus on extending our procedure, by modeling the traffic dynamics with a multi-class SFD, and letting the recorded velocities not only have a location but also a vehicle-type tag. Note that such an extension would not be evident, as the optimization procedure should account for the correlation between the recorded velocities of the different vehicle types.

Another potential future model extension would be the incorporation of traffic incidents. Although our model does account for uncertainties, it is specified towards the uncertainties between vehicles, whereas incidents may impact the speeds of all vehicles on a road segment. Thus, considering that incidents can even change the mean vehicle speed dynamics, one could think about an SFD that does not only take the traffic density as input, but the state of the road (e.g., incident-free or not) as well. An interesting research direction...

Figure 5.8: Amsterdam highway network, in which the nodes represent the junctions (contrary to Figure 2.9, in which the nodes only show the highway intersections.)

entails to a completely separate problem for each link, we foresee that a large part of the reported costs may even vanish by the use of parallel computing.
would be to combine the ideas behind the SFD and the MVM of Chapter 2, and to use a Markov-modulated background process to model the (incident) state that determines the corresponding shape of the SFD.