This document is a preprint version of the following article:

Title: A combined method to forecast and estimate traffic demand in urban networks

By: Pohlmann, Tobias; Friedrich, Bernhard

Published in: Transportation Research Part C: Emerging Technologies - Volume 31, June 2013, Pages 131-144

Publisher: Elsevier | Year: 2013

DOI: 10.1016/j.trc.2012.04.009

Brought to you by the Scientific Information Service for Mobility and Transport Research

Fachinformationsdienst Mobilitäts- und Verkehrsforschung

Website: www.fid-move.de

Repository: publish.fid-move.de

Contact: publish@fid-move.de

© 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license.

http://creativecommons.org/licenses/by-nc-nd/4.0/
A Combined Method to Forecast and Estimate Traffic Demand in Urban Networks

Tobias Pohlmann  
(corresponding author)  
Institute of Transportation and Urban Planning  
Technische Universität Carolus-Wilhelmina zu Braunschweig  
Rebenring 31, 38106 Braunschweig, Germany  
t.pohlmann@tu-braunschweig.de  
phone: +49 (0) 531 391-7930  
fax: +49 (0) 531 391-8100

Bernhard Friedrich  
Institute of Transportation and Urban Planning  
Technische Universität Carolus-Wilhelmina zu Braunschweig  
Rebenring 31, 38106 Braunschweig, Germany  
friedrich@tu-braunschweig.de

Abstract

This paper presents a combined method for short-term forecasting of detector counts in urban networks and subsequent traffic demand estimation using the forecasted counts as constraints to estimate OD flows, route and link volumes. The method is intended to be used in the framework of an adaptive traffic control strategy with consecutive optimization intervals of 15 minutes. The method continuously estimates the forthcoming traffic demand that can be used as input data for the optimization. The forecasting uses current and reference space-time-patterns of detector counts. The reference patterns are derived from data collected in the past. The current pattern comprises all detector counts of the last four time intervals. A simple but effective pattern matching is used for forecasting. The subsequent demand estimation is based on the information minimization model that has been integrated into an iterative procedure with repeated traffic assignment and matrix estimation until a stable solution is found. Some enhancements including the improvement of constraints, redundancy elimination of these constraints and a travel time estimation based on a macroscopic simulation using the Cell Transmission Model have been implemented. The overall method, its modules and its performance, which has been assessed using artificially created data for a real sub-network in Hannover, Germany, by means of a microsimulation with Aimsun NG, are presented in this paper.

Keywords
forecasting, traffic demand estimation, information minimisation model

1. Introduction

Adaptive traffic control strategies (ATCS) for online optimization of traffic signal settings highly depend on a preferably precise estimation of the current or upcoming traffic demand in a network. In general, these applications continuously optimize network-wide traffic signal settings for the next time interval to come. Therefore they need an always updated estimated traffic demand that can be used as input data for the optimization.

This paper presents a method for a continuous estimation of traffic demand in an urban sub-network for consecutive time intervals. The method is intended to be used in the framework of a newly developed ATCS. However, the focus of the paper is on the demand estimation only. The method estimates OD matrices and, more precisely, traffic volumes on all alternative and reasonable routes and on all links in the sub-network to be optimized. The only information assumed to be available in order to estimate the mentioned data are detector counts at signalized intersections. Common ATCS use optimization intervals that may last from only one or a few minutes up to half an hour or even more. In this work the duration of each optimization interval has been set to 15 minutes. The traffic demand during this interval is assumed to be static.
The method consists of two main modules. The first module forecasts detector counts based on an approach by Förster (2008). It uses so-called space-time-patterns of detector counts. This module along with a minor modification and some first results is presented in paragraph 3.

The second module estimates the aforementioned OD matrices and traffic volumes based on the forecasted counts. It is based on the research by Wang (2008) and Friedrich/Wang (2006, 2008) respectively who base themselves on van Zuylen/Willumsen (1980). Their method has been adopted, examined and enhanced with regard to further modification for online control strategies. The enhanced method, its several steps and some results are described in paragraph 4.

An extensive microsimulation has been used in order to generate artificial data that could be used for testing the overall method and its modules. In anticipation of the detailed descriptions of the method and its modules the setup of the simulation study is sketched first in paragraph 2. On the one hand an example network helps to better understand the correct application of the modules. On the other hand the artificially created data will be referred to in paragraphs 3 and 4 when the performance of the modules is presented. Therefore it should be clear how this data has been produced.

The paper closes with a description of the achieved performance of the overall method in paragraph 5 and some conclusions in paragraph 6.

2. Generation of artificial test data by microsimulation

The necessary data for testing the forecasting and demand estimation and its modules has been produced using the micro-simulation software Aimsun NG, version 5.1.8. An urban sub-network in Hannover, Germany, with eight signalized intersections, two pedestrian lights and two non-signalized intersections has been chosen and modeled. Its graph is shown in figure 1. Links equipped with a loop detector are highlighted as bold lines. A total of 55 detectors are located on the lanes in front of traffic signals. Since there are also mixed lanes for more than one turning movement and non-signalized intersections some turning flows are thus not detected. The network has nine origins and destinations. 108 plausible alternative routes between 71 reasonable OD relations have been selected. At signalized intersections the real fixed time signal plans have been modeled.

![Figure 1: graph of the test network in Hannover, Germany](image)

The traffic demand used for the study is semi-fictitious. An OD matrix for the morning peak interval of 15 minutes and one for the afternoon peak interval of the same duration have been deduced from real
measurements on-site. The percentages of usage of the alternative routes between the OD pairs have been set manually in such a way that they appeared reasonable and at the same time reproduce the real measurements as well as possible when assigning the OD matrix to the network accordingly. The total traffic demand for the study comprises 56 consecutive intervals of 15 minutes, starting at 6 am and ending at 8 pm. The OD matrices of the intervals are linear combinations of the aforementioned morning and afternoon peak matrices and reflect the profile over time observed on-site in an acceptable way.

30 replications with different random seeds have been simulated to generate average detector counts, average OD flows and average route and link volumes for every 15-minute time interval. Furthermore, data from each replication has been stored separately. This artificial data could then be used for intensive tests of the methods described in the following paragraphs.

3. Forecasting of detector counts

3.1. Methodology

Before the traffic demand of the next optimization interval can be estimated, detector counts serving as constraints for the estimation process have to be forecasted. The forecasting technique used for this purpose is a slightly adapted version of the approach proposed by Förster (2008). He used two-dimensional reference space-time-patterns of average detector counts to forecast traffic counts of all \( j_{\text{max}} \) detectors in a sub-network simultaneously for a relatively short prognosis horizon of 20 minutes. The reference patterns are derived from available data collected in the past. Over a certain time period of several weeks or months, counts \( q_{i,j} \) of every detector \( j \) in the sub-network are collected daily for all time intervals \( i \) of a day \( (i_{\text{max}} = 72 \) in the case of 20-minute intervals). The data is clustered into several relevant groups (e.g. different weekdays, Sundays, holidays), each containing data of \( K \) days. The reference pattern of each group consists of reference values \( q_{i,j}^{\text{ref}} \) that are the averages of all counted values of a specific time interval \( i \) for a specific detector \( j \), so that:

\[
q_{i,j}^{\text{ref}} = \frac{1}{K} \sum_{k=1}^{K} q_{i,j}^k
\]  

Thus, each reference pattern contains \( i_{\text{max}} \) times \( j_{\text{max}} \) reference values. Once the patterns are derived, they can be used in the future to forecast detector counts. It is certainly advisable to keep the reference patterns up to date by continuously including recent counts while dismissing outdated ones.

In order to forecast the counts of the upcoming time interval the real detector counts \( q_{i,j}^{\text{real}} \) that have been observed in the currently ended time interval \( i = T_0 \) and the previous \( N \) time intervals \( T_{-1} \) to \( T_{-N} \) are used as current traffic count pattern with \( N+1 \) times \( j_{\text{max}} \) values. All reference patterns are scanned for a sub-pattern of the same size that matches best the current count pattern. While Förster used either the correlation coefficient \( r_{xy} \) (that has to be preferably high) or the mean squared error MSE (that has to be preferably low) to identify the best matching sub-pattern within the reference patterns a combination of both has been used in this work in order to reduce systematic over- or underestimation of traffic counts. These systematic errors can be observed occasionally for some time intervals where the method fails to find the appropriate sub-pattern. Details on this effect and its origin can be found in Pohlmann/Friedrich (2009). The adapted version chooses the sub-pattern with the lowest MSE among those three sub-patterns with the highest \( r_{xy} \).

Once the best reference sub-pattern comprising the time intervals \( T_{r}^\prime \) to \( T_{-N}^\prime \) has been found the values \( q_{i,j}^{T_{r+1}\prime} \) of the time interval following the sub-pattern are taken and the forecasted values \( q_{i,j}^{T_{r+1}\prime} \) are determined as follows:

\[
q_{i,j}^{T_{r+1}\prime} = \frac{1}{K} \sum_{k=1}^{K} q_{T_{r+1}\prime,j}^k - \frac{1}{K} \sum_{k=1}^{K} q_{i,j}^{T_{r+1}\prime}^k \]  

3.2. Performance of the module

In order to assess the quality of the described forecasting technique including the modified approach to identify the best matching sub-pattern the artificial data generated by the aforementioned simulation has been used. For
each replication all detector counts have been logged for every time interval. Contrary to Förster (2008), 15-minute intervals instead of 20-minute intervals have been used. Even though the same defined traffic demand is fed into the network in every simulation run, detector counts vary in a certain range due to the random behavior of the microsimulator. The logged detector counts can thus be interpreted as \( K = 30 \) samples of the daily traffic demand of a specific weekday. A reference pattern based on these samples has been created using equation (1).

The artificial data could then be used to analyze how well the “real” detector counts from every simulation run can be forecasted. The number \( N \) of previous time intervals has been set to 3 as recommended by Förster (2008), i.e. the current detector count pattern comprises a total of four intervals. Forecasting has not only been done for the next interval \( T_{+1} \) but also for the subsequent interval \( T_{+2} \). The latter is of special interest because of the fact that a sophisticated ATCS optimization algorithm takes some computational time to generate new parameter settings. While it is running, the next time interval has already started. Therefore, it is necessary to forecast not only the traffic demand of the next time interval but rather of the interval after next.

Figure 2 shows the result of the forecasting for interval \( T_{+1} \) using the logged data of an arbitrarily selected simulation run. It comprises the forecasted counts of all 55 detectors and 52 intervals (with \( N=3 \), the first four intervals cannot be forecasted). The left part shows a comparison between the forecasted counts and the corresponding “real” counts. Since the reference pattern consists of averaged counts and therefore the forecasted counts are likely to be close to these due to the nature of the method the left part of figure 2 compares the forecasted counts to the average counts of the corresponding interval which are obtained by considering all 30 simulation runs. It can be seen clearly that the method produces good estimates of the average detector counts of the next prognosis interval whereas the comparison to the real counts observed during a specific replication run reveals a reduced but still acceptable quality of the forecasted counts.

The very good correlation with average counts must not gloss over the fact that systematic over- or underestimations could not be entirely avoided in all replications. However, this effect could be reduced noticeably compared to the original way of identifying the reference sub-pattern.

Table 1 gives an overview of the quality of the forecasted counts for all 52 time intervals and all 30 replications, i.e. the considered quality criteria \( r_{xy} \), root mean square error (RMSE) and relative root mean square error (RRMSE) have been determined for each of the \( n = 30 \cdot 52 = 1560 \) intervals based on the 55 forecasted detector counts of the respective interval. Again, both real counts and average counts have been used as reference, and the forecasting has been done both for one and for two intervals ahead. While good average values for RMSE and RRMSE have been achieved over all 1560 intervals, the respective maximum values show that the forecasted counts of some intervals suffer from a loss of accuracy. The percentage of intervals with an RMSE exceeding 10 veh/h ranges between 0.38 and 2.24 percent depending on the reference and the number of intervals to look ahead. An RRMSE exceeding 0.2 has been observed for 0.26 to 3.66 percent of all intervals.

Figure 2: quality of forecasted counts for an arbitrarily selected replication
### 4. Traffic demand estimation

#### 4.1. Information minimization model

Once the detector counts of the next optimization interval have been forecasted they can be used to estimate the upcoming traffic demand (i.e. OD flows, route and links volumes) of this interval. For this purpose the information minimization (IM) model is used as underlying method in this work. It has been presented by van Zuylen/Willumsen (1980). The network has to be modeled as a directed graph consisting of links $a$ (cp. figure 1). The IM model uses available traffic counts $q_a$ on several of these links as constraints for the estimation. Furthermore, it is assumed that the portions $p_{aij}$ of the trips from origin $i$ to destination $j$ passing link $a$ are known. Based on this input data and the flows $f_{ij}^0$ taken from a historic matrix, all current flows $f_{ij}$ between all origins and destinations are estimated in an iterative procedure by using the following equation:

$$ \begin{align*}
  f_{ij} &= f_{ij}^0 \prod_a X_a^{p_{aij}/q_a} \\
  q_a &= \sum_a p_{aij} \quad (3)
\end{align*} $$

At each iteration step the proportionality factors $X_a$ for each link with traffic counts are adapted according to the difference between real and estimated traffic counts $q_a$ until both match for all links used as constraints.

#### 4.2. Improvement of constraints

In this work the forecasted counts from detectors at signalized intersections are used as constraints for the IM model. These counts have to be assigned to the corresponding links of the graph. For best estimation results the constraints should be as detailed as possible. The most detailed information would be all turning flows (left, thru, right) at all approaches of all intersections. However, not all turning flows at intersections are measured independently or at all. Often mixed lanes for thru- and turning movements exist, i.e. only the total of both movements is available. To enhance the information obtained from the available detector counts a simple iterative algorithm is proposed here that derives as many missing counts on links without detectors as possible. This algorithm requires that all nodes of the graph either have only one input link and multiple output links or vice versa. Therefore, a node with multiple input and output links as shown in figure 3 has to be decomposed into two nodes connected by a single link if the graph does not fulfill this requirement. Nodes with only one input and output link are not allowed either, but they are dispensable anyway.

![Figure 3: decomposition of a multiple input – multiple output node](image)

Taking the following steps identifies all links without original counts whose missing traffic volumes $q_a$ can be derived exactly:

<table>
<thead>
<tr>
<th>reference intervals ahead</th>
<th>$r_{xy}$</th>
<th>RMSE</th>
<th>RRMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg</td>
<td>max</td>
<td>&gt; 10</td>
</tr>
<tr>
<td></td>
<td>[-]</td>
<td>[veh/h]</td>
<td>veh/h</td>
</tr>
<tr>
<td>real counts</td>
<td>1</td>
<td>0.98</td>
<td>6.76</td>
</tr>
<tr>
<td>counts</td>
<td>2</td>
<td>0.98</td>
<td>6.88</td>
</tr>
<tr>
<td>average counts</td>
<td>1</td>
<td>1.00</td>
<td>1.42</td>
</tr>
<tr>
<td>counts</td>
<td>2</td>
<td>1.00</td>
<td>1.55</td>
</tr>
</tbody>
</table>

Table 1: overall quality of forecasted counts for all replications

The table reveals that some inaccuracies remain and have to be accepted. It will be shown later in paragraph 5.2 how the not entirely precise forecasted counts that are used as constraints for the demand estimation affect the latter.

![Table 1: overall quality of forecasted counts for all replications](image)
Step 1: All nodes having a single input link without count are stored in set $N_1$.
Step 2: All nodes having a single output link without count are stored in set $N_2$.
Step 3: For each node in $N_1$ the missing count on the input link can be derived by summation of the counts of the output links if these are all available. In this case the node is removed from set $N_1$ and, if applicable, the start node of the input link from set $N_2$.
Step 4: For each node in $N_1$ the missing count on the output link can be derived by summation of the counts of the input links if these are all available. In this case the node is removed from set $N_2$ and, if applicable, the end node of the output link from set $N_1$.
Step 5: All nodes having multiple input links, only one of which without count, are stored in set $N_3$.
Step 6: All nodes having multiple output links, only one of which without count, are stored in set $N_4$.
Step 7: For each node in $N_3$ the missing count on one of the input links can be derived by subtraction of the available counts of the other input links from the count of the output link if the latter is also available. In this case, the start node of the link whose count has been derived is removed from set $N_3$ or $N_4$ if applicable.
Step 8: For each node in $N_4$ the missing count on one of the output links can be derived by subtraction of the available counts of the other output links from the count of the input link if the latter is also available. In this case, the end node of the link whose count has been derived is removed from set $N_1$ if applicable.
Step 9: If no (further) count could be derived in steps 3 to 8, the algorithm stops. Otherwise it returns to step 3.

Figure 4 shows an example of the algorithm applied to a simple graph. The black links in the left-most image correspond to separate turning lanes or mixed lanes at a T-junction and the adjacent junctions respectively, all equipped with detectors and thus having counts. By applying the steps described above several times, all missing traffic volumes including the desired ones on four of the six turning links of the T-junction can be derived in the example. (In figure 4 bold lines represent links whose count is derivable in the respective step. Since now these links have a derived count their link colour stays black in the following steps and they can be used to derive even more counts on other links.)

4.3. Elimination of redundant information

Van Zuylen (1981) described a drawback of the IM model that derives from redundant information. He showed that the estimation result of the flows between the four possible OD pairs in figure 5 varies depending on the link counts used for the estimation. If all links have counts and are used as constraints, a first solution is found. It should be expected that the result of the estimation stays the same if one of the four links is not used as constraint. This does not reduce the available information since the information on the missing link (no matter which one of the four has been chosen) is a linear combination of the information on the remaining links. However, the result of the estimation differs depending on the link information that is omitted. Thus, the method requires that any information used as constraint is independent from any other used information. Van Zuylen proposed a modification of the original IM model to deal with this effect.
Wang (2008) showed that this problem can also be dealt with by using not only the $p_{aij}$ of the links with counts when calculating $g_{ij}$ in equation (3), but those of all links in the network. Considering this, the method always produces the same stable estimation result, whether all four links are used as constraints or one of the four links is omitted. This is because the exponent $p_{aij}/g_{ij}$ better reflects the real weight of a constraining count if $g_{ij}$ considers all links in the network. This approach turned out to perform slightly better than the modified IM model, especially when the $p_{aij}$ are not known exactly but only estimated, which is the case in reality.

However, completely redundant information should still be avoided. According to Wang (2008) and Friedrich/Wang (2006, 2008), completely redundant information arises for example from a link that splits up into several turning links at a node. If the information on the first link is available and used as constraint, as well as the information on several or all of the turning links, the latter are completely redundant and thus the information has a higher weight during the estimation process compared to other non-redundant information, which is not justified and leads to a reduced quality of the estimation result. However, the information on turning links is more valuable for the estimation process because it is more detailed, as has been argued before in paragraph 4.2. The approaching link only contains sort of accumulated information. Therefore, several rules have been defined by Wang (2008) and Friedrich/Wang (2006, 2008) in order to eliminate completely redundant information while maintaining the most precise information possible. These rules consider different types and combinations of nodes with a varying number and arrangement of connected links (see figure 6 with examples for the most relevant considered structures). In principle, redundant information on approaching links ($p_{aij}$ and, if available, $q_{aj}$) is eliminated by subtraction of more precise information on turning links if counts on these turning links are available. Applying these rules on the network before starting the IM model assures that no completely redundant information is used.

The node structure specific rules of Wang (2008) and Friedrich/Wang (2006, 2008) have been transformed into a general algorithm in this work. The algorithm can be applied to any network fulfilling the aforementioned requirement that nodes always have to have either a single input link and multiple output links or multiple input links and a single output link.

For each link in the network it has to be checked whether it contains accumulated information ($p_{aij}$ and, if available, $q_{aj}$) that is also contained more detailed on other links. To do so, two sets of links are used. Set $L_1$ contains all links that are single input links at their end node and set $L_2$ contains all links that are single output links at their start node. By definition, each link of the graph must be in at least one of the two sets. Starting with set $L_1$, all single input links are successively checked for redundancy in downstream direction. For illustration, figure 7a shows an example where link 7 is the link to be checked. Using this link as root, a tree is generated by applying a recursive tree spanning algorithm. The branches of the tree end at links that fulfil one or more of the following criteria:

(a) They have a detected/forecasted or derived traffic volume (bold links 1 and 2 in figure 7a).
(b) They are one of multiple input links at their end node (link 3).
(c) They end at a destination (link 6).

Figure 7: example node structure to illustrate redundancy elimination

Thus, all traffic on any link of the tree must inevitably also have passed the root link. If a detected or derived count exists on the root link, the counts on the bold links 1 and 2 are redundant, but more detailed at the same time. Consequently, the latter are kept and the count on the root link is reduced by the counts on the bold links.

The same has to be done with the \( p_{ij} \). If no count is available on the root link, only the \( p_{ij} \) are reduced by those on the bold links. This is relevant in order to calculate the correct \( g_{ij} \), which must not contain completely redundant \( p_{ij} \) either, even though a root link without count itself will not be used as a constraint for the IM model.

The remaining information on the root link after redundancy elimination can be interpreted as the information on a virtual link connected to the end links in the tree that have no counts. In the example in figure 7b the virtual link contains the accumulated information of links 3 and 6. If all branches of the spanned tree end with links with counts, the information on the root link is reduced completely and no virtual link exists.

The information on those links in the spanned tree that do not have counts (links 3 to 6 in the example) is eliminated and these links are removed from set \( L_1 \) because they do not have to be checked for redundancy again. If used as root links themselves, the links in the corresponding trees would only be a subset of the links in the current tree. If one of these links has been checked as root link before, the information will still be deleted because its \( p_{ij} \) are contained in the root link of the new, larger tree. Only the information on the links with counts and the reduced information on the new root link are kept. The links with counts remain in set \( L_1 \) because they might themselves diverge into links with more detailed counts and their information might have to be reduced as well.

After all links in set \( L_1 \) have either been checked or removed, all links in set \( L_2 \) are checked or removed following the same principle, but in upstream direction. Links that are contained in both sets are checked for redundancy in both directions, which might result in two independent virtual links.

After redundancy elimination, all \( g_{ij} \) can be calculated by summing up all the remaining \( p_{ij} \) in the network for every relation \( ij \). During the application of the IM model, all links whose counts have not been reduced completely or at all are used as constraints.

4.4. Travel time estimation for traffic assignment

The IM model requires that all \( p_{ij} \) are known. This is generally not the case in practice. Therefore, Wang (2008) tried different traffic assignment techniques to estimate the \( p_{ij} \). In this work, two of these techniques have been taken up: the successive assignment and the stochastic user equilibrium using the C-Logit model (cp. Cascetta et al., 1996) in combination with the method of successive averages (cp. Sheffi, 1985). Link travel times needed during the assignment process were calculated by Wang (2008) by means of the well known BPR function described in the Highway Capacity Manual (HCM). It is a common volume delay function of the following form:

\[
t_a(q_a) = t_0 + \left( 1 + \frac{q_a}{C_a} \right)^\beta \tag{4}
\]

A reasonable estimation of link travel time with this equation depends on an adjusted choice of the free flow travel time \( t_0 \), the link capacity \( C_a \) and the parameters \( \alpha \) and \( \beta \). Recommended parameters are also given in the HCM.
In this work a second technique to estimate travel times has been tested where travel times are derived from a macroscopic simulation. A fast Java implementation of the Cell Transmission Model (CTM, see below) with some further extensions has been implemented to serve as traffic model for the signal optimization algorithm, which has been developed in a next step and is not part of this paper. This implementation turned out to be sufficiently fast to be used not only during the signal setting optimization process but also beforehand during the preceding process of traffic demand estimation. For the test network that has been presented in paragraph 2 a time period of 15 minutes can be simulated in 52 ms on average on an Intel Core i7-920 quad-core processor with 2.67 GHz using only one processor. (The implementation has not been designed for parallel computing on two or more processors.)

The CTM has been proposed by Daganzo (1994, 1995) and is described there in detail. Basically, it is a space and time discrete form of the macroscopic, hydrodynamic LWR model. For the following it is important to know that each link is divided into a finite number of cells. The length of a cell is the quotient of the free flow speed \( v_f \) [m/sec] and the duration \( t_{sim} \) [sec] of a simulation step. This guarantees that no vehicle can pass more than one cell during one simulation step. With \( v_f = 13.8 \text{ m/sec (50 km/h or 31.1 mph)} \) and \( t_{sim} = 1 \text{ sec} \), each cell has a length of 13.8 m (45.3 ft). Basically, in one simulation step \( t \) the inflow \( q_{in}(t) \) and the outflow \( q_{out}(t) \) of each cell are calculated and the number of vehicles \( n(t) \) in the cell is updated afterwards. With a duration of the simulation step of 1 second, the total delay of the vehicles in a cell during one step is:

\[
d(t) = n(t-1) - q_{out}(t)
\]

By summing up the delay \( d(t) \) and the inflows \( q_{in}(t) \) of all the time steps, the total delay and total inflow of a cell during the whole simulation run is obtained:

\[
D_{total} = \sum_t d(t)
\]

\[
Q_{in,total} = \sum_t q_{in}(t)
\]

The average delay of all vehicles having entered a cell during the simulation is thus:

\[
D_{avg} = \frac{D_{total}}{Q_{in,total}}
\]

The travel time on a link can then be estimated by taking its free flow travel time (which corresponds to its number of cells in seconds) and adding the average delays of all its cells:

\[
t_a = t_b + \sum_{cells} D_{avg}
\]

The travel time on a route is then obtained by summation of the travel times of its links.

Even though the approach to estimate travel times with the CTM is rather simple, a good performance could be achieved. This has been verified using the test network described in paragraph 2. For 108 different routes travel times generated by the CTM have been compared to average travel times calculated from 30 replications with Aimsun NG. The same traffic demand has been used in both cases. A correlation coefficient \( r_{xy} \) of 0.940, an RMSE of 22.55 sec and an RRMSE of 0.136 were achieved. The implemented CTM also includes the current signal timings at each intersection. Therefore it has been assumed that the effect of these signal timings on travel times would be taken into account in a better and more realistic way compared to static BPR functions. The final outcome will be discussed in paragraph 5.1.

4.5. Complete process

The IM model and its modifications have been integrated into an iterative procedure as proposed by Wang (2008). Starting with a unit matrix or a historic matrix in the first step a repeated traffic assignment and matrix estimation is executed until the estimated matrix converges against a stable solution. The iteration comprises the following steps (cp. figure 8):

\[
d(t) = n(t-1) - q_{out}(t)
\]
Step 1: The available detector counts $q_s$ serving as constraints are assigned to the corresponding links of the graph.

Step 2: As many additional counts as possible for links without counts are derived as described in paragraph 4.2.

Step 3: All detected and derived counts are balanced for consistency according to van Zuylen/Branston (1982).

Step 4: In order to have a first estimate of the $p_{aij}$, a unit matrix or the estimated matrix of the last time interval is assigned to the different routes using successive assignment or stochastic user equilibrium. For the necessary estimation of travel times either the BPR functions or the CTM based approach described in paragraph 4.4 is used.

Step 5: All redundant information is eliminated and all $g_{ij}$ are calculated. (It is sufficient to execute the algorithm identifying all redundant dependencies between links as described in paragraph 4.3 only once at the beginning. The dependencies are then stored and can be used for redundancy elimination during every iteration step.)

Step 6: A new updated matrix is estimated by the IM model using the previous matrix as historic matrix.

Step 8: If the average changes between the previous and the updated matrix are small enough, the matrix is stable and taken as solution. If the variation is still too big, the updated matrix is assigned to the routes, resulting in new $p_{aij}$, and the algorithm returns to step 5.

Step 9: The final matrix and the assignment results of the last iteration step are used to determine the estimated flows on each route and on each link in the network.

Figure 8: flowchart of the complete process of traffic demand estimation

4.6. Performance of the module

In order to test the plain demand estimation for single time intervals the demands of the morning and afternoon peak intervals have been used. The constraining average detector counts for these intervals have been extracted from the artificially created data (cp. paragraph 2). The OD matrices that produced these counts are known exactly because a defined demand has been fed to the system during the simulation. Therefore, the OD flows $f_{ij}$ that are estimated based on the counts can be compared to the “real” OD flows.

Figure 9 shows a comparison of the real $f_{ij}$ and the estimated $f_{ij}$ for the morning peak interval. The two cases of detectors being placed at all turning links or only at real detector locations have been considered (cp. figure 1). The latter is the more realistic scenario. Stochastic user equilibrium has been used as assignment technique. Table 1 shows the achieved quality of the estimation for both morning and afternoon peak interval. Note that the unit of the $f_{ij}$ is veh/h even though the duration of the time interval is 15 minutes.
Table 2: achieved quality of OD matrix estimation for morning and afternoon peak interval

As expected the quality of the estimation decreases when counts are not available for all turning links. Even though many missing counts can be derived when real detector locations are used the final information is still less precise. It can also be seen that not all $f_{ij}$ are estimated precisely even for the case that all turning links can be used as constraints. However, the performance of the estimation is of acceptable quality.

5. Performance of continuous demand estimation

5.1. Demand estimation without forecast

The method described in this paper has been designed for a continuous demand estimation that allows optimizing signal settings every quarter of an hour. Therefore, it is crucial to know how well the method can repeatedly estimate the traffic demands of consecutive 15-minute time intervals. In a first step, the case without forecasting has been considered, i.e. detector counts of the currently ended interval have been used to estimate the traffic demand of this very interval. The logged data of the simulation described in paragraph 2 has been used for extensive testing of the demand estimation. Since the simulation comprised 56 consecutive 15-minute time intervals that have been simulated in 30 different replications, a total of 1,680 different traffic demands and corresponding sets of detector counts (30 replications times 56 time intervals) could be used for testing.

Figure 10 shows a comparison between estimated traffic volumes on routes and links and the respective “real” traffic volumes observed during the simulation with Aimsun. The diagrams show the data of all 56 time intervals of an arbitrarily selected replication run. The estimation was based on the logged detector counts of the same replication run. Stochastic user equilibrium has been used as assignment technique. A unit matrix has been used as initial historic matrix in every time interval.
It can be seen that the estimated traffic volumes on routes \( r_{xy} = 0.889 \) are not outstanding. However, they can still be used to identify the routes that tend to be among the currently most heavily loaded routes. This information is valuable for strategies such as online coordination of signalized junctions. The second diagram in figure 10 shows that the estimation of traffic volumes on links, which result directly from the route volumes, is far more satisfying \( r_{xy} = 0.991 \). Based on these link volumes the traffic demand for the online optimization of traffic signal settings in the form of source inflows and turning rates is generated. Thus, a good reproduction of the current traffic situation in the traffic model used for optimization can be expected.

Figure 11 and 12 illustrate the quality of route volume estimation for all time intervals and all replications. The average \( r_{xy} \) (bold black line) over all replications plus/minus standard deviation (fine black lines) is shown in figure 11. Additionally, the results for every single replication run are plotted as faint gray lines. Figure 12 shows the same for the RMSE.

The average \( r_{xy} \) has a stable value over time of about 0.88 whereas the average RMSE is clearly proportional to the demand profile. It varies between 6.95 and 20.39 veh/h. The average RRMSE is stable again with a value of about 0.59 (not illustrated). The same evaluation for the estimation of link volumes reveals a much better stable average \( r_{xy} \) of 0.99 and a slightly worse RMSE between 10.03 and 24.57 veh/h. However, link volumes are much higher than route volumes and thus the RRMSE has a good value of about 0.11.
Further settings have been analysed (cp. table 3). Instead of using a unit matrix as start matrix in every time interval the estimated matrix of the previous time interval has been used as historic matrix. No significant differences could be observed. The iterative algorithm of repeated traffic assignment and matrix estimation is obviously able to adapt any historic matrix to a rather similar estimated matrix.

Using successive assignment instead of stochastic user equilibrium turned out to be less accurate. The traffic demand used for evaluation includes no oversaturated periods and thus travel times on the different routes depend much more on the signal settings than on the traffic volumes on the links. Because of this, the successive assignment tends to assign all vehicles of an OD pair to the shortest route only. This results in a worse estimation of the \( p_{ij} \) and reduces the quality of estimation.

Surprisingly, using the traditional BPR-function for travel time estimation instead of the CTM-based travel times did not have an as big negative impact as expected, even though the quality of the estimation is slightly reduced, at least concerning the estimation of route volumes. Obviously, the estimated \( p_{ij} \) are still good enough and the algorithm is not highly susceptible to incorrect \( p_{ij} \) as long as the variations are still in a tolerable range.

<table>
<thead>
<tr>
<th></th>
<th>route volumes</th>
<th>link volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard settings</td>
<td>( r_{xy} [-] ) 0.88</td>
<td>( r_{xy} [-] ) 0.99</td>
</tr>
<tr>
<td></td>
<td>RMSE [veh/h] 6.95 to 20.39</td>
<td>RMSE [veh/h] 10.03 to 24.57</td>
</tr>
<tr>
<td></td>
<td>RRMSE [-] 0.59</td>
<td>RRMSE [-] 0.11</td>
</tr>
<tr>
<td>previous matrix as historic matrix</td>
<td>( r_{xy} [-] ) 0.88</td>
<td>( r_{xy} [-] ) 0.99</td>
</tr>
<tr>
<td></td>
<td>RMSE [veh/h] 6.95 to 21.12</td>
<td>RMSE [veh/h] 9.89 to 23.54</td>
</tr>
<tr>
<td></td>
<td>RRMSE [-] 0.64</td>
<td>RRMSE [-] 0.11</td>
</tr>
<tr>
<td>successive assignment</td>
<td>( r_{xy} [-] ) 0.76</td>
<td>( r_{xy} [-] ) 0.97</td>
</tr>
<tr>
<td></td>
<td>RMSE [veh/h] 9.57 to 31.60</td>
<td>RMSE [veh/h] 16.12 to 46.28</td>
</tr>
<tr>
<td></td>
<td>RRMSE [-] 1.09</td>
<td>RRMSE [-] 0.19</td>
</tr>
<tr>
<td>BPR-functions</td>
<td>( r_{xy} [-] ) 0.82</td>
<td>( r_{xy} [-] ) 0.99</td>
</tr>
<tr>
<td></td>
<td>RMSE [veh/h] 7.84 to 24.40</td>
<td>RMSE [veh/h] 10.49 to 25.45</td>
</tr>
<tr>
<td></td>
<td>RRMSE [-] 0.64</td>
<td>RRMSE [-] 0.12</td>
</tr>
</tbody>
</table>

Table 3: comparison of average estimation results for different settings

5.2. Demand estimation with forecast

For signal setting optimization it is preferable to have an estimate of the forthcoming traffic demand in the next interval or even the one after next. Therefore a combination of detector count forecasting and demand estimation has been assessed as well. Detector counts forecasted according to paragraph 3.1 have been used as constraints in order to estimate the traffic demand of the next optimization interval. The standard settings (unit matrix used as historic matrix, stochastic user equilibrium, CTM-based travel times) have been used. The results are summarized in table 4.
<table>
<thead>
<tr>
<th>forecast</th>
<th>reference</th>
<th>quality criterion</th>
<th>route volumes</th>
<th>link volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>none (cp. table 2, standard settings)</td>
<td>real volumes</td>
<td>( r_{xy} [-] )</td>
<td>0.88</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE [veh/h]</td>
<td>6.95 to 20.39</td>
<td>10.03 to 24.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RRMSE [-]</td>
<td>0.59</td>
<td>0.11</td>
</tr>
<tr>
<td>1 interval ahead</td>
<td>real volumes</td>
<td>( r_{xy} [-] )</td>
<td>0.88</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE [veh/h]</td>
<td>10.84 to 20.16</td>
<td>22.99 to 36.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RRMSE [-]</td>
<td>0.55</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>average volumes</td>
<td>( r_{xy} [-] )</td>
<td>0.93</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE [veh/h]</td>
<td>8.86 to 16.37</td>
<td>8.79 to 20.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RRMSE [-]</td>
<td>0.42</td>
<td>0.08</td>
</tr>
<tr>
<td>2 intervals ahead</td>
<td>real volumes</td>
<td>( r_{xy} [-] )</td>
<td>0.88</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE [veh/h]</td>
<td>12.08 to 20.11</td>
<td>22.96 to 37.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RRMSE [-]</td>
<td>0.55</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>average volumes</td>
<td>( r_{xy} [-] )</td>
<td>0.93</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE [veh/h]</td>
<td>8.39 to 16.72</td>
<td>8.71 to 22.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RRMSE [-]</td>
<td>0.42</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 4: results of route and link volume estimation for different scenarios

The first row of table 4 repeats the results for the case without forecasting presented in paragraph 5.1. The lower part of the table compares these results to the case that forecasted detector counts with less precision than the real counts observed for a specific time interval are used. If the quality criteria are determined by comparing the estimated route and link volumes to the real volumes of the respective optimization interval a decrease of the quality of estimation can be observed which is not surprising. RMSE for both route and link volume estimation and RRMSE for link volume estimation are a bit higher. Only RRMSE for route volume estimation is slightly decreased. No relevant difference can be observed between the two cases of forecasting one or two intervals ahead.

It has been highlighted in paragraph 3.2 that the forecasting method tends to produce average detector counts as forecasted counts. Therefore, the estimated route and link volumes have been compared not only to the real volumes of every interval of the respective replication but also to the average volumes of each interval over all replications. This analysis reveals a much higher quality of estimation with an average RRMSE for link volume estimation of only 0.08 for both cases of forecasting one or two intervals ahead. Apparently, the forecasted counts being close to the average counts of a specific interval are a good basis for estimating average route and link volumes for the respective interval.

As has been stated in paragraph 3.2, systematic over- or underestimation of detector counts cannot be excluded completely. The sometimes reduced quality of forecasted traffic counts directly influences the quality of the estimated traffic demand, resulting in a few time intervals with a rather bad quality of route and link volume estimation. This is not reflected in the average values in table 4. However, for the majority of cases the method produces good results.

### 6. Conclusions

The achieved results show that the described method is appropriate for forecasting traffic counts and estimating the expected average traffic demand of the next upcoming one or two time intervals, as long as the demand profile over time follows a typical reference pattern. The performance of the method is mostly good and the results are, though not perfect, at least satisfying. Especially the estimation of current link volumes is of notable quality and can most likely be used as input data for online control strategies. Future research has to show how well traffic signal settings can be optimized based on the estimated traffic demand that does not perfectly reflect the real traffic demand to come.

By its nature the forecasting technique used in this work cannot cope with oversaturation. In this case detectors measure capacity rather than traffic demand and the resulting reference patterns or the current pattern will not reflect the real traffic demand. Furthermore, the method will fail when lanes with detectors are closed, e.g. because of road works. This will change the detector count pattern significantly, and the reference patterns are no longer valid during these periods. Both issues should also be addressed in the future. For now it has to be stated that the method in its present state can only be used for undersaturated conditions.
Acknowledgements

The authors would like to thank Deutsche Forschungsgemeinschaft (DFG) for granting and supporting the research project with grant number FR 1670/4-1. The content presented in this paper is a direct result of a work package of this project.

References


