DFROUTER—Estimation of Vehicle Routes from Cross-Section Measurements

TeRon V. Nguyen, Daniel Krajzewicz, Matthew Fullerton and Eric Nicolay

Abstract This contribution evaluates and improves the open-source “DFROUTER” tool that is contained in the SUMO traffic simulation suite. DFROUTER uses vehicle counts (e.g. from inductive loops) to calculate routes of vehicles through road networks. This approach is designed for highway corridors that are covered with measurement facilities at all entry and exit points. The study analyzes DFROUTER’s current functionality and compares it with other approaches that have a similar purpose. Tests performed using different networks and sensor coverage amounts are presented. Additionally, an extension to the software is presented that completes missing flows, increasing the correctness of the tool’s results.

1 Introduction

Transport planners and traffic engineers worldwide challenge with the increase in traffic amount. A wide range of measures is implemented to tackle this problem, ranging from large-scale traffic management strategies to in-vehicle Intelligent Transport Systems (ITS). Accordingly, the deployed methods range from traffic access regulations, such as calming areas or speed limits over route guidance, to in-vehicle solutions that advice a speed to use to pass the next traffic light at green or that help in changing lanes. All these solutions target the improvement of traffic in means of safety or efficiency and a more optimal use of natural resources.

One first step to take within the development of such solutions is to model the situation on roads. Besides the representation of the road networks, a proper

T.V. Nguyen · M. Fullerton
Institute of Transportation, Technische Universität München, Arcisstraße 21, 80333 Munich, Germany

D. Krajzewicz (✉) · E. Nicolay
German Aerospace Center, Institute of Transportation Systems, Rutherfordstraße 2, 12489 Berlin, Germany
e-mail: daniel.krajzewicz@dlr.de

M. Behrisch and M. Weber (eds.), *Modeling Mobility with Open Data*, Lecture Notes in Mobility, DOI 10.1007/978-3-319-15024-6_1

representation of the traffic demand is one of the major inputs for transportation system operation, design, analysis, and planning [1]. Origin-destination (O-D) matrices are one of such representations. They contain information about the spatial and temporal distribution of activities in different traffic zones in an area. Various methods for generating origin-destination matrices have been proposed, like household surveys, roadside interviews, license plate recognition and returnable-post card interviews [2]. However they are all expensive and obtaining the data is cumbersome [3].

Meanwhile, many data about vehicles flows (numbers of vehicles) is being collected for other traffic management purposes. Often, inductive loop detectors are used that are installed under the road surface. Inductive loops usually collect information such as vehicle type and speed, traffic volume, and detector occupancy. Being continuously retrieved for other purposes, such data is usually available at a lower price than the employment of previously mentioned methods. But while induction loops are a good source of information about the number of vehicles on a street, they fail to provide information about the vehicles' further routes. Hence, methods for estimating traffic flows between origin/destination pairs have been developed. Usually, they efficiently combine traffic count based data with other available information [4].

Within this report, the application "DFROUTER" is analyzed. DFROUTER uses detector values to calculate routes for simulated vehicles through a given motorway/corridor simulation network. It is included in the microscopic road traffic simulation package SUMO [5]. Besides routes, this tool also generates the according demand for the traffic simulation, consisting of single (microscopic) vehicle insertion definitions the traffic simulation can read. This paper provides a detailed description of the tool and compares it with other similar approaches that do not necessarily place restrictions on the network type. Special attention is given to the accuracy of the tool where reproduction of the flows' probabilities is the decisive indicator for this evaluation.

This report starts with an introduction into the problem of O-D matrix estimation. Then, DFROUTER and the algorithms it uses are discussed. Afterwards, evaluations of the DFROUTER and comparisons to other approaches are given. Then, an extension to the DFROUTER is presented that allows using it on highway networks that are not completely covered with detectors. This report ends with a conclusion.

2 Theoretical Background

O-D matrices describe traffic demand by dividing a given area into so-called "traffic assignment zones" (TAZs). For every TAZ at which traffic participants start (the demand origins), the number of participants that approach a destination TAZ is given. O-D matrix estimation methods based on traffic counts have been developed over the last 30 years. There are two major types of O-D matrix estimation: the static method assumes O-D flows are constant over time for determining an average O-D demand for long-time transport planning and design purposes; whereas the dynamic method considers O-D flows with time variation for short-term strategic traffic control and management [6]. One could state that dynamic methods are an extension of static methods considering the time varying dimension. Furthermore, due to congestion effects, the O-D matrix estimation can use proportional assignment (uncongested) or equilibrium assignment (congested), resulting in four basic cases of O-D estimation [1].

Contrary to urban road networks where more than one route between one origin-destination pair exists, an O-D pair in a highway corridor consisting of an on- and an off-ramp has only one possible route. This less complicated characteristic facilitates the estimation of vehicle routes and traffic demand based on detector data. Specifically for highways, the problem of determining an O-D matrix from traffic counts can be formulated as follows:

$$\sum_i b_{ij} O_i = D_j \tag{1}$$

$$\sum_j b_{ij} = 1 \tag{2}$$

where

b_{ij} proportion of trip from i to j;
O_i on-ramp counts (origin flows);
D_j off-ramp counts (destination flows).

Considering an example highway section that illustrates the O-D matrix estimation problem, the one shown in Fig. 1 could be used.

As shown in Table 1, several results could satisfy the requirements due to the under-specification problem: there are fewer equations than variables. The problem does not have a unique solution.

Many O-D matrix estimation techniques exist, where Information Minimization (IM) and Entropy Maximization (EM) [2], Maximum Likelihood (ML) [4], Generalized Least Squared (GLS) [3], or Bayesian Inference approach [7] could be named as the most popular static ones. Regarding dynamic methods, one can find

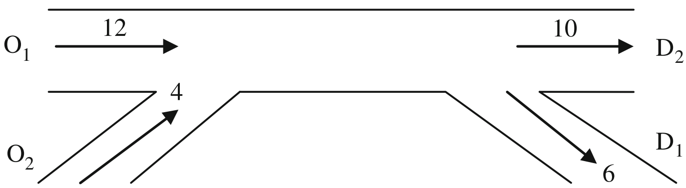

Fig. 1 Sample highway segment

Table 1 Some examples for O-D matrices that correctly represent the flows of the example from Fig. 1

	D_1	D_2	Sum		D_1	D_2	Sum		D_1	D_2	Sum
O_1	8	4	12		10	2	12		6	6	12
O_2	2	2	4		0	4	4		4	0	4
Sum	10	6	16		10	6	16		10	6	16

approaches such as Cross-correlation matrices, Constrained optimization, Recursive estimation, Kalman filtering [8], Recursive least square [9], Artificial Neural Networks [10], and Combined estimators [11].

3 DFROUTER

3.1 Development Context

DFROUTER builds upon experience gained during the set-up of a large-scale traffic observation and prediction project which results were applied during the Pope's visit in Germany in the year 2005 [12]. About 1 Mio persons were expected to participate in this event that took place on a green field near to the city of Cologne. The project's scope was to support the police and the local traffic management with on-line information about the state on the roads. The deployed system consisted of an airborne camera-based traffic surveillance system mounted under a zeppelin that sent information about recognized vehicles to a traffic management center. Together with measurements from inductive loops, this data was used to calibrate a mesoscopic traffic simulation. This simulation had the task to extrapolate the traffic counted at measurement points over the road network as well as to predict the traffic situation half an hour into the future. The so obtained states of the road network were visualized at the traffic management center of the city of Cologne (Fig. 2).

To achieve the goal of predicting traffic, an initial demand was needed, that could be calibrated using on-line measurements after deployment. The available input data included two commercial O-D matrices. The first one described a usual

Fig. 2 *Left* the zeppelin that carried the airborne traffic surveillance system; *Right* the visualization

working day. The second one was a prediction of the road traffic during the Pope's visit, but covered only one half of the area that was defined to be simulated. Both matrices were static, containing the demand of a complete day. Additionally available was a microscopic demand from the TAPAS project [13], which was based on a synthetic population. On-line data were supported by the highway administration of North Rhine-Westphalia and by the traffic management center of the City of Cologne.

Besides the amounts of passing vehicles, on-line calibration requires information about the routes which newly inserted vehicles shall use for continuing their journey. For the reasons outlined in the following, it was decided to use the inductive loop measurements not only for adapting the simulated traffic flow volumes to the measured vehicle numbers, but also as the ground truth for computing routes across the highway part of the simulated area.

The first reason to name is the uncertainty whether the available demand descriptions were applicable. To obtain routes running over the measurement points, a traffic assignment [14] would have to be performed, first. But this process is very sensitive to both, the road network representation as well as to the used demand. As both were not completely revalidated at this decision step, it was assumed that the resulting routes distribution would be erroneous. In addition, the given matrices resembled different traffic conditions (usual day vs. visitors' traffic) and had different granularities (microscopic from a synthetic population demand vs. static O-D matrices). Attempts to combine these matrices were dismissed.

Moreover, first system runs have shown that using the given O-D matrices as source of routes distributions yields in a too high memory consumption: at each measurement point, a distribution of routes to use has to be given. The simulated network was very large and routes are defined as a list of all road network edges the simulated vehicle shall pass within the used simulation SUMO. To decrease both, their number as well as the sizes of the routes stored for each measurement point, the demand was split into highway and non-highway-parts. Every vehicle that entered the highway was given a new route. When leaving the highway, the vehicle obtained a new route again. This kept the routes relatively small. This was only possible, because the area around the city of Cologne is well-covered with sensors, including all highway entries and exits.

This kind of modelling breaks all previously existent O-D relationships of single vehicles as their routes are constructed from different route distributions. This was acceptable, because the project's target was to resemble the flows on the simulated road network, not the mobility of single participants. What was realized as a tool for the Pope's visit was transformed into DFROUTER in subsequent projects.

3.2 Algorithm

DFROUTER performs several steps to obtain routes and the vehicle insertion definitions, being mainly:

1. Reading the road network to route on, the detector positions, and their measurements,
2. Detector Classification,
3. Routes computation,
4. Flows generation,
5. Writing the results.

The major algorithms and overall features of the DFROUTER are explained more detailed in the following subsections.

3.2.1 Detector Classification

The needed functionality included a classification of the detectors into the following types:

- "pure sources": starting points of routes—vehicles that enter the highway get a new route assigned;
- "in between": the simulated vehicle numbers are adapted to the measurements at these positions; only vehicles that are added obtain a new route;
- "pure sinks": ending point of routes—vehicles get a new route assigned that is based on given data from the available demand descriptions.

A detector is classified as a "pure source" if the following constraints are valid:

- there is no other detector on the same street in front of it,
- there is no detector on any foregoing street.

Analogous, a detector is classified as a "pure sink" if

- there is no other detector on the same street behind,
- there is no detector on any following street.

3.2.2 Routes Computation

The main steps of the algorithm that computes the route usage probabilities are as following. Please note that usually measurements are given per-lane and need to be summarized for each cross section.

- Step 1: Determine downstream detectors (taking into account downstream road junctions) for all source and in-between detectors.
- Step 2: Calculate the proportion of flow for each junction using detector data; junction directions not equipped with detectors get a probability of 1.0 as default (what is a fallback to work with real-life networks).
- Step 3: Calculate destination distributions for all source detectors by multiplying all flow probabilities on all edges constructing that route.

Simply spoken, the algorithm computes routes by taking the destination proportion as route probability at every junction. If all sink detectors are supplied, the flows should be replicated correctly. But a single solution to the O-D guessing problem can be only obtained if there is only one origin and the network is fully covered by detectors. This is rather not the case for real-world networks.

This simple algorithm fails in the case of missing detectors, especially detector data on split edges (in-between or sink detectors) as it is not able to guess the missing data and thereby cannot compute the probability to choose one of the subsequent roads. As a default, the probability to use the non-observed road is set to 100 %, overestimating it. This default is rather arbitrary chosen—any other used value would be incorrect as well.

3.2.3 Output Generation

Vehicles are inserted at source detector positions. For every detector recognized as being a "pure source", DFROUTER generates the routes distributions and a list of vehicles that shall be inserted into the simulation network at this position. A route distribution is defined as a route and a probability to choose it, where a route is defined as a list of edges to pass.

As during computation, a routes distribution was obtained for every cross section, DFROUTER can write inputs to in-simulation flow "calibrators" for each in-between detector. These calibrators may be loaded into a simulation scenario. There, they adapt the number of passing vehicles to the read values by adding/removing vehicles into/from the simulation.

A further output consists of "variable speed sign" definitions for sink detectors, as well readable by the simulation. These simulation instances read a time line of speeds and apply them to a defined lane. This feature is mainly used in jam formation analysis to model boundary conditions properly. The speed is read from the detector measurements while reading the flow amounts. Besides "variable speed signs", so-called "rerouters" may be additionally written for sink detectors. Equipped with—externally generated—route distributions, these in-simulation instances assign a new route to passing vehicles. Within the Pope's visit, they were used to assign new routes to vehicles that leave the highway.

4 Evaluations

In the following, different evaluations of DFROUTER are presented. At first, synthetic scenarios are given to DFROUTER to determine how well it can reproduce an originally completely known flow. In a second step, a single scenario is used to compare DFROUTER to some selected O-D estimation algorithms.

4.1 Replication of Synthetic Scenarios

In order to analyze the algorithm, several abstract highway networks and demands, ranging from simple to complex, were applied. The four factors to be considered are the network type, the number of detectors, vehicle flows, and routes. These elements were altered to test the generated results. It was expected that the algorithm works well in simple cases but may fail when being confronted to more complex ones.

Beginning with two on- and off-ramps, the initial network was incrementally extended to more complicated scenarios with extra ramps, lanes, entrances, and exits (origins and destinations). Basically, there is one main highway line connected to several on- and off-ramps equipped with detectors.

The evaluation is performed by generating virtual detector data using the simulation SUMO. The resulting measurements from simulated inductive loops are then given to DFROUTER for generating routes and demand definitions. Routes and vehicle flows are the main indicators for this evaluation. In general, the flows/routes/detectors generated by DFROUTER should be identical to the initial input for SUMO simulation. The general work flow of this analysis is shown in Fig. 3.

The used synthetic scenarios are shown in the following figures (Figs. 4, 5 and 6).

A comparison of the output generated by DFROUTER against the initial input for the three cases shows that:

- The algorithm works well if the network is fully covered with detectors and generates routes comprising all O-D pairs. The algorithm could not detect that some routes were absent; e.g. in one scenario of CASE 2 there were only 4 routes but DFROUTER created 6 routes, which consist of all possible connections.

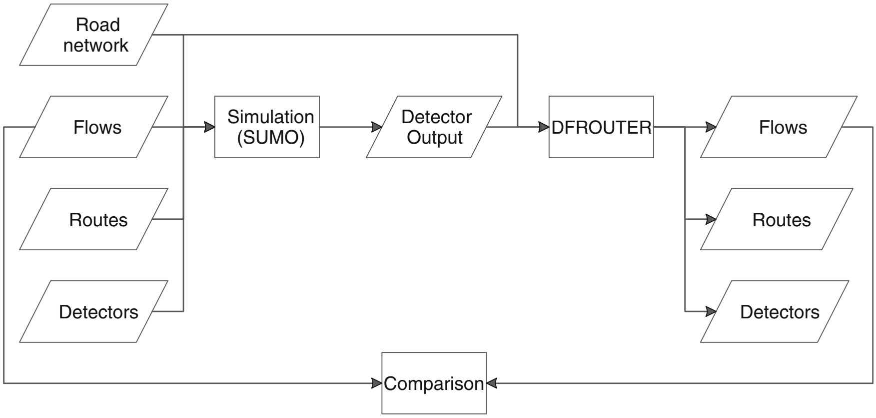

Fig. 3 The work flow of the evaluation process

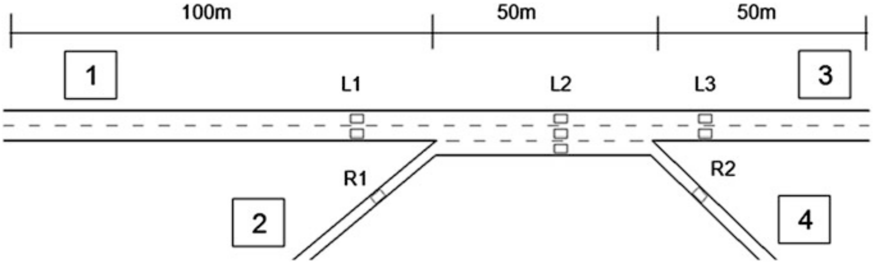

Fig. 4 CASE 1—2 origins, 2 destinations

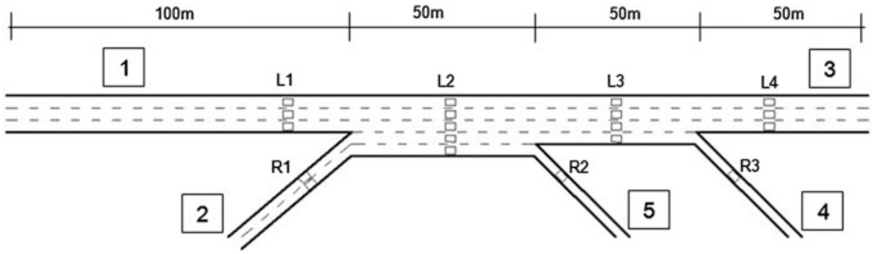

Fig. 5 CASE 2—2 origins, 3 destinations

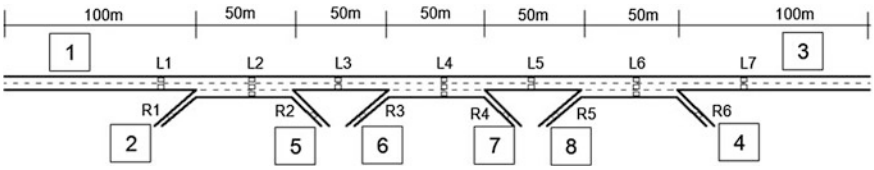

Fig. 6 CASE 3—3 origins, 3 destinations

- Missing in-between detectors (in three cases) do not cause a big estimation problem as long as the source and sink detectors are present. This shows that the in-between detectors do not play an important role in the probability estimation procedure.
- Basically the estimated probabilities are identical to flow proportions at destinations, therefore sink detectors are the decisive elements in flow computation.
- This simple algorithm does not work successfully in the case of missing detectors, especially detector data on split edges (in-between or sink detector) as it is not able to guess the missing data.

4.2 Comparison with Other Approaches

In the following, some O-D matrix estimation approaches are described and compared with DFROUTER's algorithm. Some of these approached do not necessarily place restrictions to the network type. For a fair comparison, the same scenario is given to the compared algorithms. The main used performance indicator is the route probability.

DFROUTER generates route/demand data based merely on proportions of flows on split edges. The destination distribution is an average result of different calculations performed on time slices with a duration of 60 s, as default. Congestion effects and the travel time between the origin and the destination are not considered. This method is most likely to work for the static O-D estimation method mentioned above (a workaround would be to run DFROUTER multiple times with data split into intervals for which routes are desired, e.g. 15 or 60 min). However the algorithm considers only constraints between link flows (sum of all link proportions equal to 1.0 in case of full detector coverage) but no optimization function (e.g. minimization differences between estimated and observed link flows).

The used scenario, summarized in Fig. 7, comprises detector data as shown in Table 2 and the highway network as used for the initially described CASE 2 (Fig. 5).

The DFROUTER algorithm calculates flow probabilities for each of the split edges by examining the outflows of each junction considering off-ramp counts and

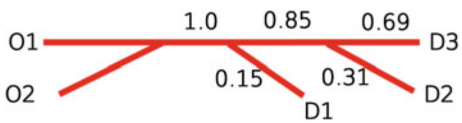

Fig. 7 Comparison test case configuration

Table 2 Network settings and detector data used for comparison

Item	Value
Section length	100, 50, 50, 50
On-ramp counts	280, 180
Off-ramp counts	70, 120, 270

Table 3 The DFROUTER O-D matrix

O\D	D_1	D_2	D_3
O_1	= 1.0 * 0.15	= 1.0 * 0.85 * 0.31	= 1.0 * 0.85 * 0.69
	= 0.15	= 0.26	= 0.59
O_2	= 1.0 * 0.15	= 1.0 * 0.85 * 0.31	= 1.0 * 0.85 * 0.69
	= 0.15	= 0.26	= 0.59

Table 4 The equally split O-D matrix

	D_1	D_2	D_3
O_1	1/3	1/3	1/3
O_2	1/3	1/3	1/3

mainline counts, e.g. 70/460, 390/460, 120/390, 270/390 (equal to 0.15, 0.85, 0.31 and 0.69 respectively).

The destination distribution can be obtained by multiplying the available probabilities on each route departing from a source detector as shown in Table 3.

This O-D matrix estimation method can be compared to similar approaches, which generate traffic demand without taking into account an optimization function, such as the equally split O-D matrix, the proportional O-D matrix, iterative methods, the gravity model, and turning percentages.

4.2.1 The Equally Split O-D Matrix

This is the simplest method for seed generation. As the name suggests, an equal proportion is assigned to all destinations. In the test case (Fig. 7) with three destinations, the method concludes that D_1, D_2 and D_3 are equally likely for trips from origin O_1 and O_2, so the proportion will be 1/3 (33.3 %) (Table 4).

4.2.2 Proportional O-D Matrix

This is one of the most common and oldest methods to estimate an O-D matrix [15]. It is based on the concept that the attraction of any destination is the function of the number of trips that end at that destination. In other words, higher attraction yields in a higher flow proportion. The origin flow will hence be distributed according to destination flows.

Considering the test case (Fig. 7) where destination flows collected at D_1, D_2, D_3 are 70, 120, 270 vehicles respectively, the proportional O-D matrix can be computed manually as follows, which is identical to DFROUTER's calculation (Table 5).

Table 5 The proportional O-D matrix

	D_1	D_2	D_3
O_1	= 70/(270 + 120 + 70)	= 120/(270 + 120 + 70)	= 270/(270 + 120 + 70)
	= 0.15	= 0.26	= 0.59
O_2	= 70/(270 + 120 + 70)	= 120/(270 + 120 + 70)	= 270/(270 + 120 + 70)
	= 0.15	= 0.26	= 0.59

4.2.3 Iterative Method

This is considered as a hybrid proportional assignment technique that balances both inflows and outflows [15], adopted from Wills and May (1981) based on an iterative fitting algorithm. The algorithm computes each O-D cell iteratively until a convergence is reached. The algorithm steps are given below.

Step 0

$$\text{set } \mathrm{k} = 0$$
$$\text{set } T_{ij}^{(0)} = \begin{cases} 1 & \text{for all possible interchanges} \\ 0 & \text{for all impossible interchanges} \end{cases}$$

Step 1

$$\text{set } T_{ij}^{(2k+1)} = \frac{O'_i}{\sum_j T_{ij}^{(2k)}} T_{ij}^{(2k)} \quad \text{for all } i, j \tag{3}$$

where $\mathrm{O'_i}$ is the observed volume at point *i* adjusted for all known demands from i.
Step 2

$$\text{set } T_{ij}^{(2k+2)} = \frac{D'_j}{\sum_j T_{ij}^{(2k)}} T_{ij}^{(2k)} \text{ for all } i, j \tag{4}$$

where $\mathrm{D'_j}$ is the observed exit volume at point *j* adjusted for all known trips that end at j.
Step 3

$$\text{if } \mathrm{T_{ij}^{(2k+2)}} - \mathrm{T_{ij}^{(2k)}} < \delta \quad \text{for all i, j then STOP}$$
$$\text{else set k} = \mathrm{k} + 1 \text{ and go to Step 1}$$

Using this algorithm to compute the O-D matrix for the test case (Fig. 7) yields in the results shown in Table 6. The algorithm produced a converged output after two iterations.

The final iterative O-D matrix estimation, however, contains the same values as that of the proportional O-D matrix estimation. This may be because the method computes O-D elements iteratively but does not consider any constraint such as

Table 6 The iterative O-D matrix estimate

	D_1	D_2	D_3
O_1	0.15	0.26	0.59
O_2	0.15	0.26	0.59

distance or travel time as inputs to a deterrence function. The following gravity model will take these parameters into account.

4.2.4 The Gravity Model

The gravity model is one of the oldest trip distribution methods and is widely used in macroscopic modelling. An extension proposed by Nancy Nihan [16] uses the impedance function to estimate the trip proportion between ramps. It is related to the concept that the probability of very long and very short trips is low on the freeway. The model is based on the Gamma distribution as follows:

$$F_{ij} = \frac{\beta^{\alpha}}{\Gamma(\alpha)} d_{\mathrm{ij}}^{(\alpha-1)} e^{-\beta d_{\mathrm{ij}}} \tag{5}$$

where

F_{ij}	is the travel propensity factor between ramp *i* and *j*;
α	shape factor ≅ 3.0 for the highway;
β	size parameter = α/(average trip length);
d_{ij}	distance between pair (*i*, *j*);
average trip length	(1/T) * Σ(Link length) * (Link volume);
T	sum of all trips generated.

The cell entries in the O-D matrix are defined as:

$$T_{ij} = \frac{b_j F_{ij}}{\sum_j b_j F_{ij}} O_i \tag{6}$$

where

T_{ij}	trip interchange between pair (i, j);
b_j	balance factor from iterations;
O_i	production at *i*;
D_j	attraction at *j*.

In addition, the following constraint has to be fulfilled:

$$\sum_i T_{ij} = D_j \tag{7}$$

In the implementation of the algorithm, the balancing factor was ignored for the first iteration. The average trip length from the geometry is:

$$(100 * 280 + 50 * 460 + 50 * 390 + 50 * 270)/(280 + 180) = 183.$$

Table 7 The Gravity model O-D matrix

	D_1	D_2	D_3
O_1	0.4312	0.3371	0.2317
O_2	0.2222	0.3909	0.3869

Table 8 Turning percentage O-D matrix

	D_1	D_2	D_3
O_1	0.15	= 0.31*(1 − 0.15)	= 1 − 0.15 − 0.26
		= 0.26	= 0.59
O_2	0.15	= 0.31*(1 − 0.15)	= 1 − 0.15 − 0.26
		= 0.26	= 0.59

Therefore parameter $\beta = 3/183 = 0.016$. Using these parameters and the distance matrix, the O-D matrix results are calculated accordingly (Table 7).

4.2.5 The Turning Percentage

This is the most intuitive method of estimating an O-D matrix for a freeway section. Similar to the equally split and proportional O-D estimate methods, it assumes that turning percentages at any given off-ramp are independent of the trip origin [15]. Therefore the O-D matrix is derived by tracking the turning percentages in each section. Using the test case (Fig. 7), there are four sections, each between on-ramp and off-ramp, with turning percentages as follows: 0, 15.2, 30.8 and 100 (0, 70/460, 120/390, 270/270, respectively). The resulting O-D matrix is shown in Table 8.

4.2.6 Discussion

The equally split O-D matrix method did not generate a plausible result. Due to the missing value for the balance factor b_j, the gravity model has not been examined thoroughly and therefore produced rather incomplete output in the first calculation iteration. Similar O-D matrices were achieved from various approaches: DFROUTER, the proportional O-D matrix, the used iterative method, and the turning percentage. The comparison results also indicate that DFROUTER is working most similarly to the turning percentage approach as it takes each flow proportion at each split edge into consideration. In contrary to the iterative method, it does not take distance, time, or any deterrence parameter into account, but performs its computations based on the number of origin and destination counts only. The results therefore are proportional to these counts.

Furthermore, DFROUTER and the proportional O-D matrix also have similar working mechanisms. Considering a tree graph as follows including one origin and seven destinations where *a*, *b*, *c*, *d*, *e*, *f* are the respective detector data on edges (Fig. 8).

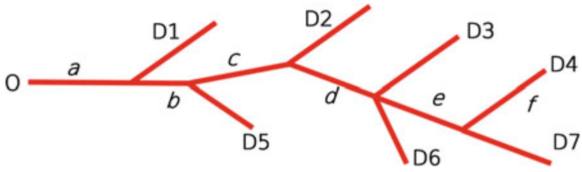

Fig. 8 A tree graph

Then the flow probability at each destination, e.g. D_4, is computed as:

$$\text{DFROUTER: } pro = \frac{b}{a} \times \frac{c}{b} \times \frac{d}{c} \times \frac{e}{d} \times \frac{f}{e} = \frac{f}{a}$$

$$\text{Proportional O-D matrix: } pro = \frac{f}{\sum_{j=1}^{n} D_j} = \frac{f}{\sum_{j=1}^{n} O_i} = \frac{f}{a}$$

From above, it could be said that for the case of one origin, DFROUTER and the proportional O-D matrix use a basically same approach. The proportional O-D matrix works more simple than DFROUTER as it does not take into account in-between detectors or split edges; only the data at sink detectors are used for calculation. A different approach named SYNOD has been developed to synthesize the required O-D matrix based on proportional O-D matrix approach. This simple proportionality scheme is on the other hand considered as a crude approximation that has the problem of over-predicting the number of very short and very long trips with 20–30 % level of error as described in [16].

Due to the drawbacks of these methods, they are often used to generate a starting solution (seed or target, a priori matrix) for the O-D estimation problem to solve the minimization function of difference between estimated and observed link flows or O-D matrix [15].

5 Extension for Completing Missing Measurements

From the analysis of DFROUTER and other, already known issues, several improvements to the algorithm could be considered:

- Guessing missing data based on existing detector flows. This could be done by considering the relationship between all inflows and outflows at a certain junction.
- Regarding the travel times when computing route usage probabilities; currently the probabilities are only computed regarding the same time slice of detector measurements.
- Computing route probabilities individually for passenger and heavy duty vehicles; albeit both types are usually explicitly counted and given in according

measurements, DFROUTER computes only route probabilities for the overall vehicle amount.
- Improving DFROUTER's operation for the case of highway rings or a fully covered urban intersection.

The most promising improvement is to guess the missing data on one of two (or several) split edges. By doing this, DFROUTER could perform well even in case of not all "pure sinks" being covered with detectors. The overestimation problem of the current DFROUTER that assigns probability = 1.0 as default for missing detector data could be eliminated. The following subsections describe this extension. At first, the algorithm to compute the missing data is given, followed by an evaluation of its function in an abstract road network. This section closes with a report on the application of the improved DFROUTER for a complex, real-world network.

5.1 Calculating Missing Data

The initial algorithm takes only those split edges that have a detector on them into account and omits those without a detector. This problem could be solved by the algorithm proposed in the following:

- Step 1: Calculate the flow value on each edge of the highway network using backward or forward recursion.
- Step 2: Determine split edges after a junction for all routes starting from source or between detectors to sink detectors.
- Step 3: Calculating flow proportion of split edges based on computed flow so that each split edge contains a different probability.
- Step 4: Calculate destination distribution by multiplying all flow probabilities on all edges constructing that route for routes starting from source detectors only.

Of all steps above, the first is the most challenging one as there are many dependencies to consider. Consider an edge e for which detector values are missing. Forward recursion will be performed when there is no detector before *e* and backward recursion works in the opposite way.

If the algorithm could not figure out the value after a certain number of recursions, its probability will be re-set to 1.0.

5.2 Application in an Abstract Network

A hypothetical highway network was developed to test the improved DFROUTER. It was designed to contain all cases listed in Table 9. There are only seven detectors at the location of L1, L9, R1, R2, R5, R66, R7, the remaining on- and off-ramps are

Table 9 Cases to consider in the recursion algorithm

	Recursion forward	Recursion backward
1	e afterE e = afterE	beforeE e e = beforeE
2	e afterE x e = afterE - x	beforeE e x e = beforeE - x
3	e afterE e = ∑afterE	beforeE e e = ∑beforeE

missing, including some in-between and sink detectors. The input probabilities will be compared with DFROUTER's output (Fig. 9).

In order to calculate flow at a certain edge, the recursion function will be used, whatever it is forward or backward. For instance, R3 will be computed as follows (Fig. 10):

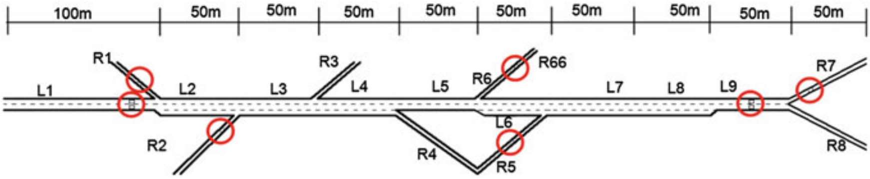

Fig. 9 Abstract network with missing detectors

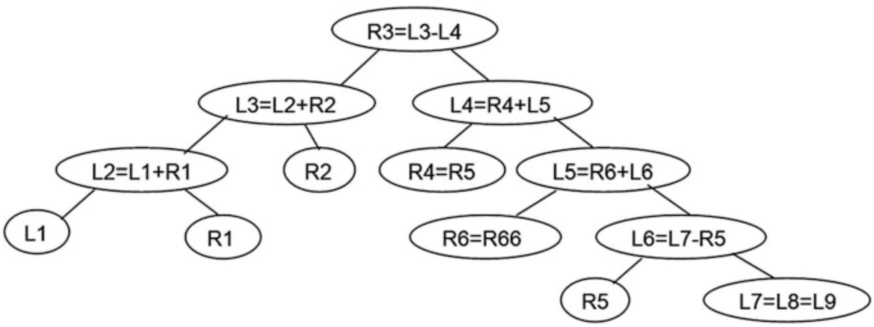

Fig. 10 Example of calculating R3

Table 10 Comparison of the destination probabilities of the original and the improved algorithm

Trip	Des-counts	Des-pro	Probability		Relative error	
			DFROUTER	Improved DFROUTER	DFROUTER	Improved DFROUTER
From L1/R1/R2 to R3	900	0.24	1	0.23	3.22	−0.03
From L1/R1/R2 to R66	500	0.13	0.14	0.13	0.06	−0.01
From L1/R1/R2 to R7	1,100	0.29	0.69	0.28	1.38	−0.03
From L1/R1/R2 to R8	700	0.18	0.69	0.20	2.75	0.09
From L1/R1/R2 to R7_1	300	0.08	0.69	0.09	7.74	0.14
From L1/R1/R2 to R8_1	300	0.08	0.69	0.06	7.74	−0.24

Calculated results and their comparisons are shown below regarding both probabilities at destinations and the original input as well. The differences are evident and significant as the original DFROUTER does not consider missing data at destinations. The probabilities generated by the improved DFROUTER are approximate to the destination probabilities and are more accurate compared to the original DFROUTER (Table 10).

5.3 Application in a Larger Network

To evaluate the algorithm on a more complex, real scenario, a larger network containing three main interchanges in Nuremberg was converted from OpenStreetMap data. Each interchange is equipped with different numbers of detectors (see Fig. 11).

- Interchange 1: fully covered with detectors and there are five routes as an input to SUMO
- Interchange 2: only detectors in main corridor; only one route toward interchange 1
- Interchange 3: only detectors in main corridor; only one route toward interchange 1

The flow probabilities produced by the improved DFROUTER are different from those computed by the original DFROUTER as shown in the Table 11. As expected, the results from the improved DFROUTER are more accurate.

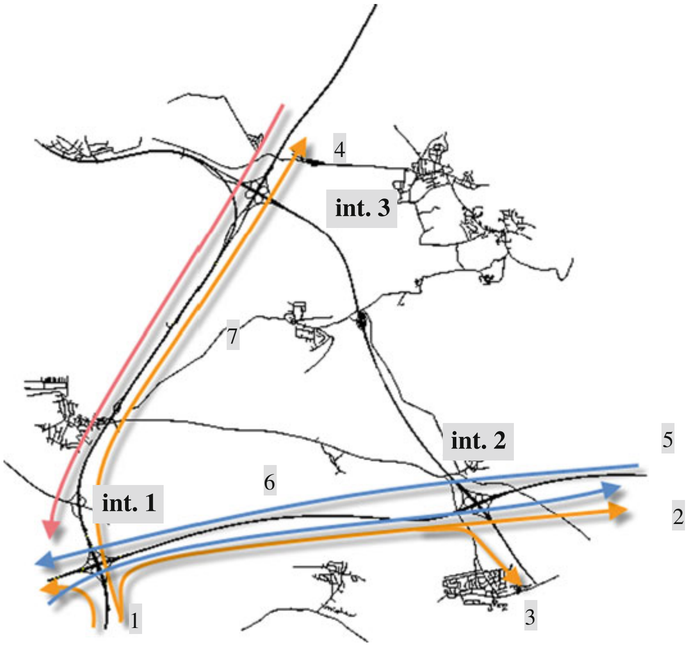

Fig. 11 Nuremberg highway network

Table 11 Comparison of DFROUTER and improved DFROUTER results with input probabilities

No	Trip	Input probability	Probability		Relative error	
			DFROUTER	Improved DFROUTER	DFROUTER	Improved DFROUTER
1	From 1 to 1 left	0.16	0.16	0.16	0.00	0.00
2	From 1 to 2 straight 1	0.13	0.06	0.18	−0.54	0.38
3	From 1 to 2 right	0.09	0.22	0.04	1.44	−0.56
4	From 1 to 3	0.63	0.24	0.62	−0.62	−0.02
5	From 1 to 2 straight 2	1	0.28	0.82	−0.72	−0.18
6	From 2 to 1	1	1	1	0.00	0.00
7	From 3 to 1	1	0.4	0.86	−0.60	−0.14

6 Conclusion

The study has been conducted to analyze SUMO's DFROUTER tool. It sought to answer the following questions:

1. How can DFROUTER be formally described?
2. What are the differences to other approaches?
3. How could the algorithm be improved in order to estimate routes/demand more accurately?

DFROUTER's results for several typical highway corridors were examined, first. Additionally, the algorithm has been compared with some O-D matrix estimation approaches based on the same abstract highway corridor. The literature review has indicated two main groups of O-D estimation: static and dynamic, which have been developed over the last 30 years. DFROUTER's approach of dividing incoming flow proportionally to off-ramp counts makes it simple and fast in calculating respective flows. In parallel, it computes results that are similar to those obtained from other algorithms.

An algorithm improvement has been proposed and applied successfully to a large highway network. It produced reliable results using recursion to guess missing data, assuring that each edge after a junction will contain a certain traffic count and relative probability. The method of multiplying individual probabilities is left unchanged. The sometimes present problem of missing detectors at destinations is thereby partially solved. The extension will be included in SUMO's standard release. The improved algorithm, however, is applicable to highway corridors (one way street) only. Future research and extension possibilities have been outlined and may be performed in the future.

References

1. Bert E (2009) Dynamic urban origin-destination matrix estimation methodology. EPFL
2. Van Zuylen HJ, Willumsen LG (1980) The most likely trip matrix estimated from traffic counts. Transp Res Part B: Methodol 14(3):281–293
3. Cascetta E (1984) Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. Transp Res Part B: Methodol 18(4–5):289–299
4. Cascetta E, Nguyen S (1988) A unified framework for estimating or updating origin/destination matrices from traffic counts. Transp Res Part B: Methodol 22(6):437–455
5. Krajzewicz D et al (2012) Recent development and applications of SUMO—Simulation of Urban MObility. Int J Adv Syst Meas 5(3 and 4):128–138
6. Kattan L, Abdulhai B (2011) Traffic origin-destination estimation in handbook of transportation engineering, Volume II: applications and technologies, 2nd edn. McGraw Hill Professional, Access Engineering, New York
7. Maher MJ (1983) Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. Transp Res Part B: Methodol 17(6):435–447
8. Cremer M, Keller H (1987) A new class of dynamic methods for the identification of origin-destination flows. Transp Res Part B: Methodol 21(2):117–132

9. Nihan NL, Davis GA (1987) Recursive estimation of origin-destination matrices from input/output counts. Transp Res Part B: Methodol 21(2):149–163
10. Yang H et al (1992) Estimation of origin-destination matrices from link traffic counts on congested networks. Transp Res Part B: Methodol 26(6):417–434
11. Bell MGH (1991) The real time estimation of origin-destination flows in the presence of platoon dispersion. Transp Res Part B: Methodol 25(2–3):115–125
12. Niebel W et al (2008) Traffic surveillance and forecast for large-scale events, monitoring and simulating the world youth day 2005 and the soccer world cup 2006. PROM: list studenata Fakulteta prometnih znanosti (21):64–66. ISSN:1332-2613
13. Justen A, Cyganski R (2008) Decision-making by microscopic demand modeling: a case study. In: Transportation decision making: issues, tools, models and case studies. ISBN:9-78-88-96049-06-8
14. Dafermos SC, Sparrow FT (1969) The traffic assignment problem for a general network. J Res Natl Bur Stan 73B:91–118
15. Muthuswamy S et al (2005) Improving the estimation of travel demand for traffic simulation: Part I, in final report 2005, Department of Civil Engineering, University of Minnesota
16. Nihan NL (1979) Use of volume data to reproduce ramp-to-ramp freeway trip patterns—a pilot study. Technical report standard, Washington State Department of Transportation, WSDOT-35.1