A TWO-STEP APPROACH FOR THE CORRECTION OF THE SEED MATRIX IN THE DYNAMIC DEMAND ESTIMATION

Guido Cantelmo
Engineering Department
Roma Tre University
Via Vito Volterra 62, 00146, Rome, Italy
guidouniroma3@gmail.com

Francesco Viti
Faculty of Science, Technology and Communication
Research Unit in Engineering Science
University of Luxembourg
Rue Coudenhove-Kalergi 6, L1359 Luxembourg city, Luxembourg
francesco.viti@uni.lu

Chris M.J. Tampère
Department of Mechanical Engineering
Center for Industrial Management - Traffic and Infrastructure
Katholieke Universiteit Leuven
Celestijnenlaan 300A - PO Box 2422, 3001 Heverlee, Belgium
chris.tampere@cib.kuleuven.be

Ernesto Cipriani
Engineering Department
Roma Tre University
Via Vito Volterra 62, 00146, Rome, Italy
ernesto.cipriani@uniroma3.it

Marialisa Nigro
Engineering Department
Roma Tre University
Via Vito Volterra 62, 00146, Rome, Italy
marialisa.nigro@uniroma3.it

Word Count
No. of words: 5339
No. of figures: 9x*250 =2225 5 Figures, 4 tables
Total: 7500

ABSTRACT

The Dynamic Demand Estimation problem is strongly related to which data are available and where, and to the choice of the starting seed matrix.

In this work deterministic and stochastic optimization methods are tested for solving the Dynamic Demand Estimation problem. All the adopted methods demonstrate the difficulty in reproducing the correct traffic regime, especially if the seed matrix is not sufficiently close to the real one.

Therefore, in this paper a new and intuitive procedure to specify an opportune starting seed matrix is proposed: it is a two-step procedure based on the concept of dividing the problem into small-sized problems, focusing on specific OD pairs in different steps. Specifically, the first step focuses on the optimization of a subset of OD variables (the ones who generate the higher flows or the ones who generate the bottlenecks on the network). In the second step the optimization works on all the OD pairs, using as starting matrix the matrix derived from the first step.

The procedure has been tested on the real network of Antwerp, Belgium, demonstrating its efficacy in combination with different optimization methods.

INTRODUCTION AND LITERATURE REVIEW

Traffic congestion, especially in urban networks, is nowadays a relevant societal problem, and of primal interest in traffic engineering. Typically, congestion phenomena are due to bottlenecks that propagate congestion on the network, making very difficult to trace back its real causes. A correct representation of the spread of congestion, which is essential for the proper evaluation of management operations, requires tools capable of simulating and predicting time-dependent network traffic conditions. To this aim, typical tools are dynamic traffic assignment models that require input information on origins and destinations of traffic demand; such information must be consistent with the time evolution of the network conditions to be estimated. Reliable traffic demand information, usually gathered with direct surveys, is difficult to be updated because of costs and time needed. The availability of cheaper, frequently updated and temporal consistent measurements on network links makes this type of observations very attractive for deriving indirect information on traffic demand, so originating the problem often referred to as the Dynamic Origin-Destination (OD) Matrix Estimation.

The dynamic demand estimation (or the demand adjustment, if we start from a known OD matrix usually derived by a combination of surveys and mathematical models) searches for temporal OD matrices that best fit link measurements as traffic counts. The problem is well-known in both the off-line (medium-long term planning and design) and in the on-line (real-time management) context. Cascetta et al. [1] proposed to face the problem using a sequential or a simultaneous approach: the first makes the demand estimation for each single time slice, holding constant the others. In the simultaneous approach the matrices of every time slice are perturbed simultaneously to guarantee full consistency between estimation periods. This approach is virtually more correct than the sequential one, taking into account the relationship among different OD pairs. On the other hand, the computational times are higher, so generally it is preferred only for the off-line context.

Different approaches and solution algorithms have been developed in the last years for both off-line and on-line dynamic OD estimation; firstly it is possible to distinguish between formulating the estimation as a single level optimization problem [2], or as a bi-level optimization problem [3]; moreover, another classification distinguishes approaches explicitly using the assignment matrix as a link between traffic counts and demand [4], or approaches using a linear approximation of the assignment matrix [5-6], or assignment-free approaches [7].

About the solution algorithms, it is well known the effectiveness of Kalman filtering, especially for capturing day-to-day dynamics [8] or for online estimation [9]; however, also studies on the Kalman filter for the off-line context are known [10]. New stochastic solution approaches have been recently proposed by Antoniou et al. [11] and Cipriani et al. [12].

Different authors focused on the problem of increasing the amount of information required by the estimation including in the objective function of the problem adding further measures compared to the traditional traffic counts, which are not able alone to discriminate between the congested or uncongested state of the network: for example, link speed and occupancy measurements have been proposed by Balakrishna [13], probe data from vehicle equipped by AVI tags by Dixon and Rilett [14], Eisenman and List [15] Caceres et al. [16], Barcelò et al. [17], Mitsakis et al. [18], aggregate demand data such as traffic emissions and attractions by zones by Iannò and Postorino [19] and Cipriani et al. [12].

The majority of the approaches reported in literature focus on the estimation of the dynamic OD matrix from the assumption that a good starting matrix (here called seed matrix) is available. This is not always possible, while the quality of the seed matrix can deeply influence the estimation result [20-21].

Starting from these remarks, this study aims at proposing a method which, based on state-of-the-art Dynamic Demand Estimation procedures, allows to build a proper dynamic seed matrix to be used as input in the estimation problem. Therefore, firstly different deterministic and stochastic optimization methods to solve the estimation problem are tested; once verified the difficulties of these methods in obtaining a demand able to reproduce the correct traffic regime on the network, especially if the seed matrix is not sufficiently close to the real one, a two-steps procedure is proposed in order to improve the quality of the seed matrix. The two-step procedure works at the first step only on a subset of the OD variables (appropriately selected), while optimizing all the variables at the second step starting from the matrix derived from the first step.

METHODOLOGY

The Dynamic Demand Estimation problem is generally solved as an optimization problem. To formulate the problem it is necessary to choose the goal function, the optimization method and the criterion to upgrade the solution during each iteration. Concerning the goal function, the goal in the estimation problem is to find the matrix that minimizes both the distances with respect to the traffic measurements and to the seed matrix. Cascetta and Nguyen [22] formalized the problem as follows:

$$\boldsymbol{d}^{*} = argmin\left[z_1\left(\boldsymbol{x},\widehat{\boldsymbol{d}}\right) + z_2\left(\boldsymbol{v}(\boldsymbol{x}),\widehat{\boldsymbol{f}}\right)\right] \tag{1}$$

The corrected-estimated matrix is the matrix $\mathbf{d}^*$ that minimizes the distance between the seed-starting matrix $\hat{\mathbf{d}}$ and the measurements from the network $\hat{\mathbf{f}}$. The function z_1 and z_2 are estimators of the error. Generally these functions are chosen among the maximum likelihood and generalized mean square error (GLS) theory. The idea in the first case is to use a function to measure the probability to observe the $\hat{\mathbf{d}}$ and $\hat{\mathbf{f}}$ vectors if $\mathbf{d}^*$ is assigned. The goal function maximizes this probability. In the second case the function takes into account the squared difference with the $\hat{\mathbf{d}}$ and $\hat{\mathbf{f}}$ vectors if $\mathbf{d}^*$ is assigned. In this case the goal function tries to minimize the error between the vectors.

The most common traffic measurements are flows, density and speeds on the network, obtained from different sources. To obtain the measurements, it is possible to use fixed detectors, but also probe vehicles, GSM data, cameras, Bluetooth sensors, etc. It is important to use measurements as the density and the speed together with the flows in the dynamic case, to intercept the correct congestion branch on the fundamental flow/density diagram [23]. Otherwise it is possible to obtain correct flows with incorrect traffic regime on the link [24]. Moreover, being the problem underdetermined (more unknowns than observations), especially when only link measurements are available, multiple matrices could generate the correct regime on the network. In order to overcome this issue, additional a priori information on demand matrix must be added in the problem: this is usually done including measurements about the starting demand (seed term) in the goal function.. The generic goal function, using a simultaneous approach on the variables, has the following form [6]:

$$(\boldsymbol{d}_1^*,\ldots,\boldsymbol{d}_n^*) = argmin\begin{bmatrix} z_1\left(\boldsymbol{l}_1,\ldots,\boldsymbol{l}_n,\widehat{\boldsymbol{l}_1},\ldots,\widehat{\boldsymbol{l}_n}\right) + \\ +z_2(\boldsymbol{n}_1,\ldots,\boldsymbol{n}_n,\widehat{\boldsymbol{n}_1},\ldots,\widehat{\boldsymbol{n}_n}) + \\ +z_3(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n,\widehat{\boldsymbol{x}_1},\ldots,\widehat{\boldsymbol{x}_n}) + \\ +z_4(\boldsymbol{r}_1,\ldots,\boldsymbol{r}_n,\widehat{\boldsymbol{r}_1},\ldots,\widehat{\boldsymbol{r}_n}) + \end{bmatrix} \tag{2}$$

where

-]$\mathbf{l}/\hat{\mathbf{l}}$ are the measurements on the links;
- $\mathbf{n}/\hat{\mathbf{n}}$ are the measurements on the nodes;
- $\mathbf{x}/\hat{\mathbf{d}}$ are the measurements on the seed demand;
- $\mathbf{r}/\hat{\mathbf{r}}$ are the measurements on the route.
- $\boldsymbol{d}_n^*$ estimated demand matrix for time interval n;

- $\boldsymbol{z}$ is the estimator

To solve problem (2), different solution algorithms have been proposed in the past. For a detailed overview we refer to Lindveld [25] and Balakrishna [13]. Concerning the optimization method, in this study, three path-search methods are used as reference: the Finite Difference Stochastic Approximation (FDSA), the Simultaneous Perturbation Stochastic Approximation (SPSA) and the Sensitivity-Based OD Estimation (SBODE) method. These are here briefly introduced.

Finite Difference Stochastic Approximation (FDSA)

The FDSA (Finite Difference Stochastic Approximation) (Kiefer and Wolfowitz [26] is a method to obtain the descent direction perturbing every OD pair in the matrix as in equation (3):

$$\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^{i} + \alpha^{i}\boldsymbol{G}^{i} \tag{3}$$

With $\boldsymbol{\theta}$ the matrix for the iteration i, α is the step length and $\mathbf{G}_i$ is the gradient. The gradient is obtained as follows:

$$\boldsymbol{G}^{i}(\boldsymbol{\theta}^{i}) = \begin{bmatrix} \dfrac{z(\boldsymbol{\theta}^{i} + c^{i}\boldsymbol{\xi}^{1}) - z(\boldsymbol{\theta}^{i})}{c^{i}} \\ \vdots \\ \dfrac{z(\boldsymbol{\theta}^{i} + c^{i}\boldsymbol{\xi}^{r}) - z(\boldsymbol{\theta}^{i})}{c^{i}} \end{bmatrix} \tag{4}$$

where ξ is the vector with zeros, except for the OD pair to be perturbed. So the number of simulations is equal to the number of the OD pairs, because every OD pair is perturbed once, to intercept the impact on the goal function.

Simultaneous Perturbation Stochastic Approximation (SPSA)

The Simultaneous Perturbation Stochastic Approximation (SPSA, [26-28]) is a stochastic approximation of the gradient, based on the numeric perturbation of the matrix to correct. With respect to the FDSA, the gradient has a stochastic component, but the computational time to obtain the descent direction is smaller as the gradient is approximated performing evaluation of only two feasible directions, and then choosing the one that produces a descent. In the SPSA, the equation to upgrade the matrix is the standard formulation reported in (3). The gradient **G** is obtained in this model with a numeric perturbation of the matrix $\boldsymbol{\theta}$. The model obtains an average direction perturbing concurrently all the OD pairs as follow:

$$\widehat{\boldsymbol{g}}_{k}(\boldsymbol{\theta}^{i}) = \frac{z(\boldsymbol{\theta}^{i} + c^{i}\Delta^{k}) - z(\boldsymbol{\theta}^{i})}{c^{i}} \begin{bmatrix} (\Delta_{1}^{k}) \\ \vdots \\ (\Delta_{r}^{k}) \end{bmatrix} \tag{5}$$

$$\boldsymbol{G}^{i} = \overline{\boldsymbol{g}}(\boldsymbol{\theta}^{i}) = \frac{\sum_{\boldsymbol{k=1}}^{\boldsymbol{Grad\_rep}} \widehat{\boldsymbol{g}}_{k}(\boldsymbol{\theta}^{i})}{Grad\_rep} \tag{6}$$

With c_i the perturbation step. Grad_rep is the number of the gradient replications. It is possible, and recommended, to repeat this perturbation to obtain a good approximation. In the equation above, the formulation of the SPSA model is presented with the asymmetric perturbation. The model formulated in this way takes the name SPSA-AD (Asymmetric Design, [29]). The advantage to use this formulation is that, with respect to the basic SPSA with symmetric design, the number of assignment needed to compute the gradient is reduced of the 50%. Both these variants will be tested on the case study.

<u>Sensitivity-Based OD Estimation (SBODE)</u>

The last method considered in this study is the Simulation-Based OD Estimation model (SBODE, [30]). The SBODE model is based on the idea of perturbing every OD pair like for the FDSA method. The formulation is very similar to the Gauss-Newton method, with the difference that it is applicable not only to quadratic problems. The model does not use the standard formulation to upgrade the solution at the ***i***-th iteration because the gradient and the step are chosen concurrently:

$$\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^{i} + \boldsymbol{p}_i \tag{7}$$

$$\boldsymbol{p}_i = -(\boldsymbol{J}^T\boldsymbol{J})^{-\mathbf{1}}(\boldsymbol{J}^T\boldsymbol{F}(\boldsymbol{x}_{i-1})) \tag{8}$$

Where J is the Jacobian and $\mathbf{F}(\mathbf{x}_{i-1}.)$ is the vector of the deviation between the measured and the simulated link flows acquired by assigning $\mathbf{x}_{i-1}$. So the SBODE model uses the Gauss-Newton only to obtain the direction. In this model is possible to include also the deviation from the a priori matrix as regularization term:

$$\boldsymbol{p}_i = -(\boldsymbol{J}^T\boldsymbol{J} + \varepsilon\boldsymbol{I})^{-\mathbf{1}}(\boldsymbol{J}^T\boldsymbol{F}(\boldsymbol{x}_{i-1}) - \varepsilon(\boldsymbol{x}_{i-1} - \widetilde{\boldsymbol{x}})) \tag{9}$$

with ε the weight of the regularization term.

The first step, after the initialization of the variables, is the simulation of the starting matrix, to obtain the goal function value and the link flows on the network. Then, the Jacobian is obtained from the starting matrix, perturbing every OD pair. In this case, the higher the dimension of the OD matrix is, the higher will be the computational time [The algorithm requires one simulation for every OD pair perturbed]. The Hessian and the gradient are obtained as follows:

$$\boldsymbol{f}_i = (\boldsymbol{J}^T\boldsymbol{F}(\boldsymbol{x}_{i-1}) - \varepsilon(\boldsymbol{x}_{i-1} - \widetilde{\boldsymbol{x}})) \tag{10}$$

Where J is the Jacobian and $\mathbf{F}(\mathbf{x}_{i-1}.)$ is the vector of the deviation between the measured and the simulated link flows acquired by assigning $\mathbf{x}_{i-1}$. In this model, it is possible to include also the deviation from the a-priori matrix as regularization term:

$$\boldsymbol{H} = -(\boldsymbol{J}^T\boldsymbol{J} + \varepsilon\boldsymbol{I})^{-\mathbf{1}} \tag{11}$$

So the following quadratic-programming problem is solved:

$$\min\ (\tfrac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{f}^T\boldsymbol{x}) \tag{12}$$

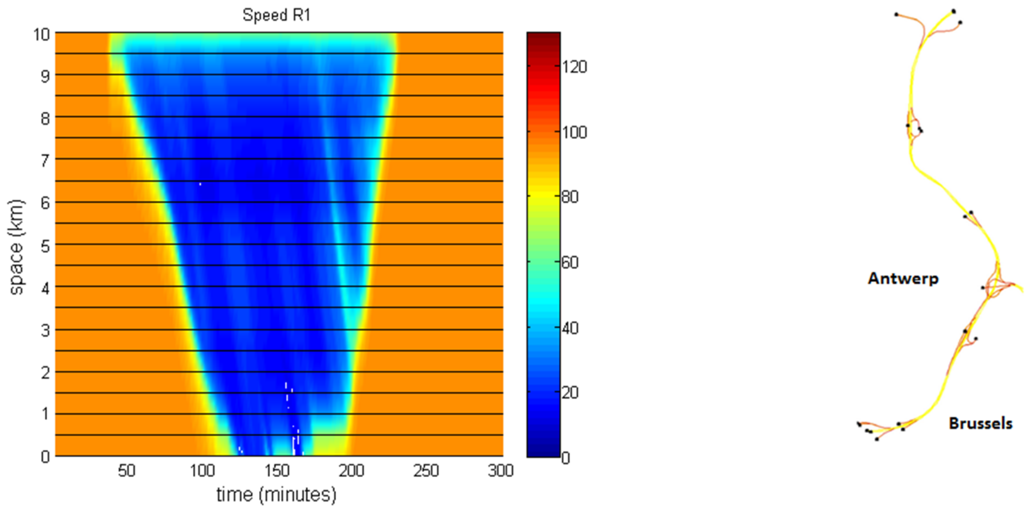
The point $\mathbf{x}^*$ is the solution of the quadratic problem. So the Gauss-Newton solution for the i- iteration is $\mathbf{x}^*$. In this study, this method is used combined with a Line Search to find the optimal step. The equation to upgrade the solution is again equation (3) . Vector $\mathbf{x}_i - \mathbf{x}^*$ is taken as descent direction, so a Line Search (LS) along this direction is done, approximating the goal function trend between $\mathbf{x}_i - \mathbf{x}^*$ to a parabolic trend. A different goal function can be used during line search, with a Boolean term to check that the new solution is still in the correct regime.

$$\boldsymbol{G}^k(\boldsymbol{x}) = \frac{\|\widetilde{\boldsymbol{y}}-\boldsymbol{y}(\boldsymbol{x}_k)\|_2^2}{\|\widetilde{\boldsymbol{y}}-\boldsymbol{y}(\boldsymbol{x}_{k-1})\|_2^2} + \frac{A}{k}\frac{\|\boldsymbol{r}-\boldsymbol{r}(\boldsymbol{x}_k)\|_2^2}{\boldsymbol{I}} \tag{13}$$

Here $\mathbf{r}$ and $\mathbf{r}(\mathbf{x})$ are vectors of binary variables indicating whether a link flow is on the corrected branch of the fundamental diagram or not.

CASE STUDY

The case study is related to the inner ring-way around Antwerp, Belgium. The network includes 56 links, 39 nodes, with 46 OD pairs. The morning peak period is considered, between 05:30 and 10:30. The data – speeds and flows – are available every 5 minutes. The detectors are located at the on- and off-ramps and on some intermediate sections. The dynamic OD flows are estimated with 15-minutes departure intervals. So the dynamic matrix contains 966 OD pairs, and the total starting demand is equal to 202.200 trips. The initial OD matrix is derived from an existing static OD matrix by superimposing a time profile. A selection of OD flows was increased to obtain a congestion pattern similar to reality. So the starting matrix presents the correct traffic regime on the detectors.



(a) (b)

Figure 1: (a) x,t plot of the measured speeds on the network, (b) Ring of antwerp

For that reason, only the flows are used inside the GLS goal function to calibrate the OD matrix. The speed measurements are used only for validation. Starting every link in the correct traffic regime, the expectation is that the new matrix preserves the correct traffic regime reducing the errors on the link flows. RMSE/RMSN are used to quantify the distance between measured and simulated speeds and flows. Also, the distance between the estimated matrix and the seed matrix is used to evaluate the different solutions. The goal function is the following:

$$\min f\left(\boldsymbol{\theta}^{i}\right) = \left[h(\boldsymbol{y}_{s} - \boldsymbol{y}_{r})\right] \tag{14}$$

With $\mathbf{y}_s$ and $\mathbf{y}_r$ the simulated and measured flows on each link. In the application of the SPSA, differently from the basic version explained in the previous pages, the step c_k is a percentage of the OD pair itself. In this way it is possible to obtain a more representative value of c_k, taking into account the different dimension of the OD pairs. Moreover, the basic gradient is multiplied for the OD pair itself, to obtain bigger steps for the bigger OD pairs. With this feature, tested by Frederix [13], the SPSA-AD obtains greater reduction of the goal function respect to the same method without this weight.

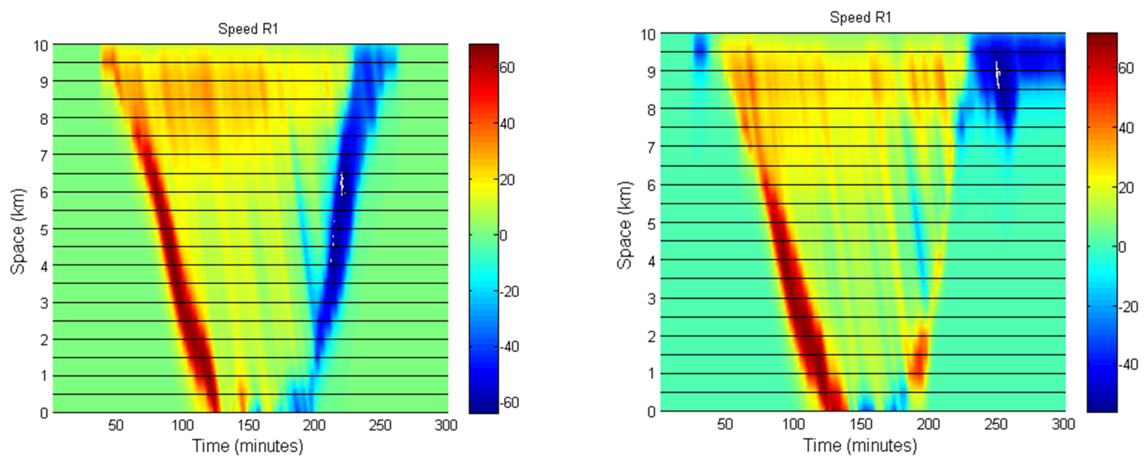
Table 1 shows the results found from the different methods. Regarding the computational time, these tests are obtained working on computer that, for every iteration, saved the results sending the data to a server. This caused the very large computational times reported. Computational speed is however not a main concern of this study. Furthermore, it is possible to find more accurate information about the computational efficiency of this method on others' work (by Frederix et.al. [5,24,30], and by Cipriani et.al [3,12,21,29]). In this study, the computational time is used only to compare the performances of the different solutions, so it is regarded as only a metric.

Table 1 Results of each method

	final deviation	FO Improvement [%]	RMSE linkflows	RMSN linkflows [%]	RMSE speeds	RMSN speeds [%]
SBODE	6.10E+07	97.08	237.73	7.35	18.64	28.93
SPSA (Ck=0.01 , Grad_rep=50)	4.29E+08	79.50	628.29	19.44	17.93	27.83
SPSA (C=0.01 , Grad_rep=1)	6.55E+08	68.66	795.44	24.61	27.72	38.94
SPSA AD(Ck=0.01 , Grad_reo=50)	3.28E+08	84.29	552.03	17.08	20.35	31.58
SPSA (Ck=0.01 , Grad_rep=1)	3.52E+08	83.17	570.67	17.65	20.61	31.98

	N.Iterations	Comp. Time for one iteration[min]	Total comp. time [h]	Total comp. time [days]
SBODE	40	420	280.00	11.67
SPSA (C=0.01 , n.dir=50)	93	41	63.55	2.65
SPSA (C=0.01 , n.dir=1)	1000	1	16.67	0.69
SPSA AD(C=0.01 , n.dir=50)	273	21	95.55	3.98
SPSA (C=0.01 , n.dir=1)	929.00	0.5	7.74	0.32
Starting deviation	2.09E+09			

It is possible to observe that the SBODE model obtains the best improvement in the goal function, but at the same time has the greatest computational time. The SPSA-AD has a greater improvement with respect to the basic model. In the following tests the version of the SPSA-AD with c_k equal to 1% and Grad_rep=50/360 is used. For a statistical analysis, eight different optimizations with these parameters were done. The average final deviation is 3.96E+08, the highest value is 5.00E+08, and the best is 3.26E+08.



(a) (b)

Figure 2: (a) Δ x,t plot of the measured speeds on the network for the solution of the SPSA, (b) Δ x,t plot of the measured speeds on the network for the solution of the SBODE

Concerning the results, it is important to highlight how, for all the methods tested, a congestion pattern very close to the real one was obtained, represented in Figure 1a. At the same time, all the methods present an offset in the congestion pattern: the congestion period begins and finishes later with respect to the real one.

This offset is evidenced in Figure 2. In this figure the time-space plots of the vector Δ, equal to the difference between simulated and measured speeds are presented. The red zone represents an overestimation of the speeds, so congestion is estimated to begin earlier in time with respect to the actual congestion pattern. On the other hand, the blue zone represents a significant underestimation of the speeds, so the estimated matrix is still congested, differently from the real one, which recovers in shorter time. This error is present in both the models, deterministic SBODE and stochastic SPSA, although they differ from each other significantly, especially at the congestion recovery part. If the offset is clearly defined in the SPSA, this difference is less evident in the SBODE.

This suggests two different results. The first one is that, as predictable, the error in the congestion pattern, presents in all optimizations performed, is higher for the stochastic methods. For the deterministic method, there is not an offset, but there is a deformation of the congestion patter. If the error is greater with the stochastic approaches, the structure of the error is the same in both the situations, with a well definite delay in the beginning of the congestion. The second consideration is that the error is not related to the optimization method, but is related to the specific case study.

To solve this problem it is necessary to change the features of the problem. One possibility is correct the starting matrix to obtain different conditions and to reduce the error in the congestion period. It is necessary to highlight how, anyway, is very difficult for the model to understand exactly the moment of the beginning of the congestion, which is normally somewhere in between the available measurement locations, and this is demonstrated by the results of the two methods.

THE TWO-STEP APPROACH

To improve the solutions obtained on the network, a new approach to the problem is formulated in this part of the study. This approach – called "Two-Step Approach"- aims to be a generic procedure applicable to different methods to improve their results, by improving the quality of the seed matrix, prior to carry on the optimization process. The basic idea is to divide the problem in two small-sized problems, and solve them separately. Using this method on both the SPSA and the SBODE method, it is possible to obtain general conclusions about the properties obtained by the application of the approach. The approach works as follows:

- <u>FIRST STEP:</u> The first step is focused on the optimization of a subset *N* of the OD pairs. The goal of this step is to correct only a part of the seed matrix, to obtain, starting from the original seed matrix – in the rest of the article called "wrong seed matrix" - a more correct dynamic seed matrix by concentrating on the OD flows that contribute the most to the congested area, i.e. those OD pairs passing onto the bottleneck link. The expectation is to delete the systematic error observed in Figures 2. Furthermore the SPSA generates an approximate gradient, with the maximum error for the bigger and the smaller flows. Starting from a matrix closer to the real one the error generated from the approximate gradient could be reduced.
- <u>SECOND STEP:</u> In the second step, the correction of the OD matrix is done, starting from the new dynamic seed matrix obtained from the first step. Therefore the procedure is the same used to obtain the results presented in Table 1, but the starting point of the optimization problem is the matrix obtained in the first step.

Before carrying on the experiments, it is necessary to understand how to define the subset *N* of variables. Two ways are explored in this work, one more analytical and another one more generic.

<u>Approach 1</u>

In this approach the subset *N* is defined as the subset of OD pairs that generate the greater flows on the network. The flows are the unique measures inside the goal function, so the subset *N* so defined is the subset of the most important descent directions for the starting seed matrix. The result of this optimization is the minimum of the function for the most important descent directions. By doing so, we focus on the part of the goal function that contributes to the largest gain. An analogous approach was recently proposed by Djukic et al. [31], in which the idea to reduce the OD demand using Principal Component Analysis is introduced.

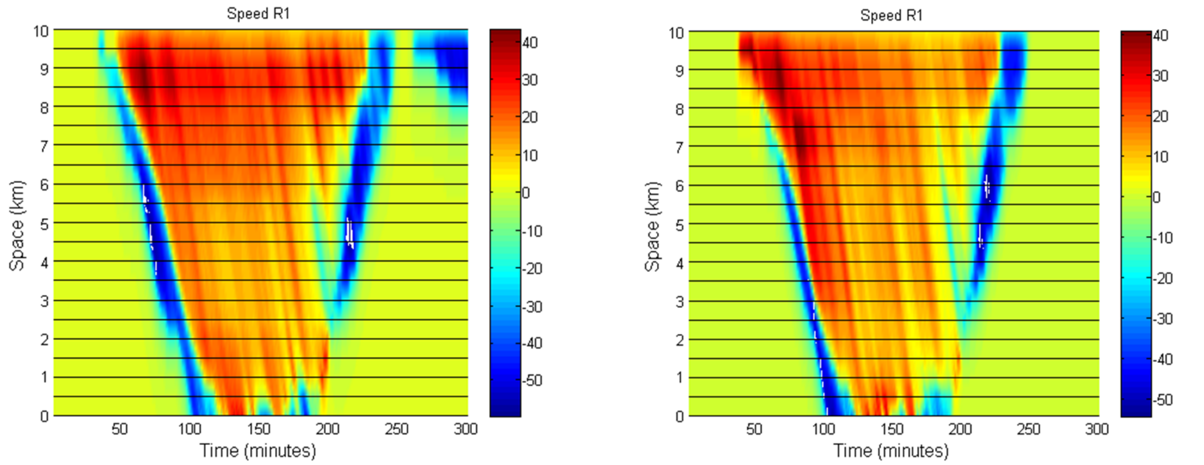
In the first step, 126 OD pairs out of 966 are selected to be included in the optimization method. Taking into account the smaller number of variables, and the will to obtain a good gradient to correct the wrong seed matrix along the main descent direction, the method chosen for the optimization is the FDSA. Starting from the results of the FDSA, both the SPSA and the SBODE are then applied. Concerning the SPSA, the result of the first step is an exact gradient that works on the greater flows. The SPSA is an average stochastic gradient, so the greater errors are generated for the greater and the smaller link flows, while there is a good representation for the average link flows. So in this example the SPSA works only on the OD pairs that were not included in the first level. The results are presented in Table 2:

Table 2 Numerical results for each method

Optimization Method	final deviation	FO Improvement [%]	RMSE link flows	RMSN link flows [%]	RMSE speeds	RMSN speeds [%]
Step 1 (FDSA)	5.80E+08	72.26	730.385	22.6	18.47	28.67
Step 2 (SPSA AD)	4.49E+08	78.52	644.82	19.95	13.67	21.16

It can be observed that, despite perturbing only 126 OD pairs out of 966, the reduction of the goal function is very high. It is important to stress out that the gradient is deterministic in the FDSA. In the second level, it is possible to do other observations. The first is that the final deviation is greater with respect to the basic SPSA-AD. On the other hand, three features of this solution result particularly interesting:

- The best congestion pattern until now is obtained in this solution. The RMSE is equal to 13.67, which is lower than the basic SBODE and all the others models.
- The absolute distance between the final matrix and the seed matrix is equal to 6.29E+04; The distance in the first level was equal to 5.10E+04, and in the basic SPSA-AD was equal to 9.18E+06, so in the second level the algorithm is closer to the seed matrix.
- The congestion pattern has a longer duration than the real one, but the offset is completely disappeared, as shown in Figure 3.



(a) (b)

Figure 3: (a) Δ x,t plot of the measured speeds on the network for the solution of the SBODE and the two-step approach,(b) Δ x,t plot of the measured speeds on the network for the solution of the SPSA and the two-step approach

The same results are obtained using the SBODE model in the second level. Also in this case the final deviation of the goal function is greater respect to the basic version.

Table 3 Numerical results for each method

Optimization Method	final deviation	FO Improvement [%]	RMSE link flows	RMSN link flows [%]	RMSE speeds	RMSN speeds [%]	N. Iterations
Step 1 (FDSA)	5.80E+08	72.26	730.385	22.6	18.47	28.67	53
Step 2 (SBODE)	7.85E+07	96.24	269.61	8.34	16.76	26	7

The model presents the same three features observed using the SPSA in the second step of the model:

- The congestion pattern is better than the basic SBODE, with the RMSE is equal to 16.76.
- The final matrix is closer to the seed matrix. The absolute distance between the final matrix and the seed matrix is equal to 1.75E+05 travelers; The distance in the first level was equal to 5.10E+04, and in the basic Gauss Newton was equal to 1.89E+05, so in the second level the algorithm is closer to the seed matrix.
- The congestion pattern has a longer duration than the real one, but the offset is disappeared (Fig.3a).

In both situations, using the SBODE or the SPSA in the second step, the offset is disappeared, but the error on the congestion patter is again on the boundary of the congestion period. In this case the congestion is slightly longer respect to the real one. Anyway the error is smaller respect to the basic approach, as demonstrated by the RMSE/RMSN of the speeds.

An important consideration is the computational time. Using the Gauss Newton in the second step, the computational time is lower than the computational time for the basic Gauss Newton. If the goal function improvement is lower - 96.24% against 97.08% of the basic Gauss Newton - adopting the two-step-model the computational time is decreased from 11.67 to 3.88 days.

<u>Approach 2</u>

In the previous example, the model is developed with the basic assumption to correct the estimation of the bigger flows in the first level, assuming that the selected OD pairs are associable to the main descent direction.

If this criterion could converge faster than the basic method, it is not however verified that the main descent direction arrives closer to real matrix. One way to obtain a more generic criterion is to work on another subset N of OD pairs. The idea is to obtain the correct regime on the bottleneck in the first step, and to use the second step to obtain the global estimation. So in the first step, the estimation problem works only on the OD pairs that have a greater influence on the bottlenecks. In the second level, as in the previous case, the global optimization is developed. So in this case 630 of 966 OD pairs are perturbed with the FDSA in the first level while in the second level all the 966 OD pairs are included in the optimization, in both the SPSA and the Gauss Newton.

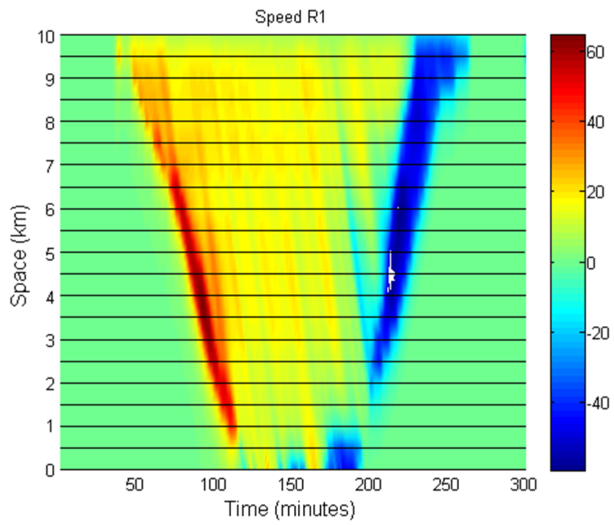


Figure 4 Δ x,t plot of the measured speeds on the network for the solution of the FDSA in the first step

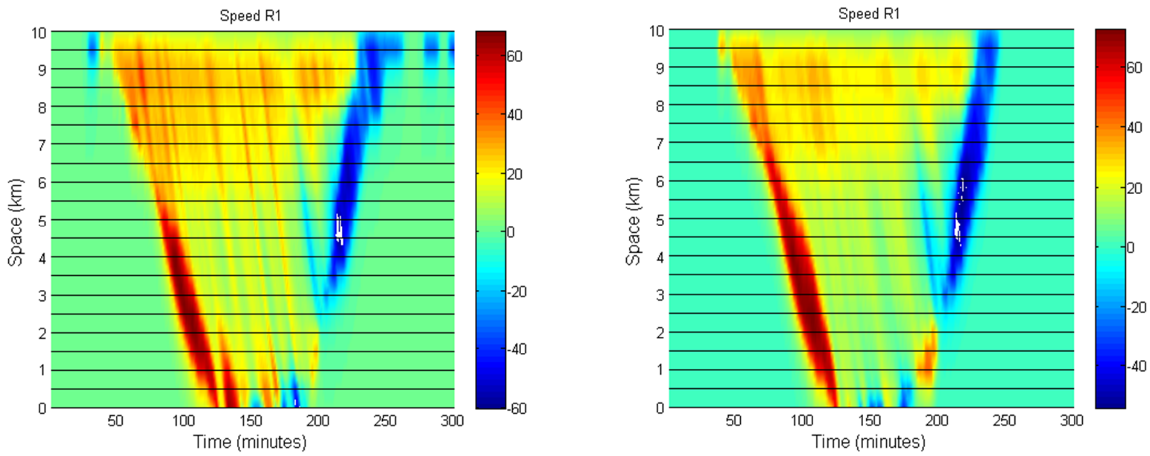
Figure 4 shows the Δ x-t plots of the speeds obtained from the assignment of the matrix output of the first-step optimization. The offset in the congestion pattern is still observed, but the error is smaller than the error obtained using the basic SPSA or Gauss Newton methods in the optimization. The main problem of this optimization is the FDSA itself. The number of variables to optimize is very high – 630 out of 966 – and, differently from the Gauss Newton, the model uses a constant step. For this reason the computational time is very high and equals 9 days. On the other hand, the value of the goal function is similar to the value obtained from the basic SPSA, while the error on the speeds is smaller. The only optimization with a better RMSE/RMSN of the speeds is the optimization obtained in the previous example using the SPSA in the second level (Table 2). In this case the absolute distance of the matrix from the seed-matrix is to 6.24E+04, so it is smaller than the distance obtained with both the basics models. Starting from this result, the second-level optimization is obtained with both the SPSA and the SBODE.

Table. 4 numerical result

Optimization Method	final deviation	FO Improvement [%]	RMSE linkflows	RMSN linkflows [%]	RMSE speeds	RMSN speeds [%]
Step 1 (FDSA)	3.72E+08	72.26	587	18.61	15.6	24.21
Step 2 (SPSA-AD)	1.46E+08	78.52	368.26	11.39	18.05	28.01
Lv 2 (SBODE)	4.07E+07	98.05	194.116	6	17.31	26.86

Figure 5 shows the x-t plots of the speeds for the solution of the second level, obtained using the SPSA as optimization method. The error of the speed is very high and the offset in the congestion pattern is greater than in the solution of the FDSA. Table 4 shows the results for the first and second step.

With respect to the basic SPSA the final deviation is smaller. The distance between the seed and the solution is the highest, respect both the solution of the SPSA implemented in the previous case and in the basic SPSA, and equal to 1.15E+05. In the end, as observable in Table 4, the error in the speeds and the offset in the congestion pattern is greater than in the previous case.



(a) (b)

Figure 5: (a) Δ x,t plot of the measured speeds on the network for the solution of the SBODE in the second step, (b) Fig.9 Δ x,t plot of the measured speeds on the network for the solution of the SPSA in the second step

As for the SPSA, also the SBODE obtained a better value of the goal function, but the computational effort is quite high. The total computational time using the SBODE model is equal to 14 days, while it is equal to 24 days using the SPSA. The error in the speeds and the error respect to the seed, is equal to 1.86E+05, are inferior than the basic method, but higher respect to both the solution of the first level and the global solution obtained in the previous case, and presented in Table 3, as shown in Table 4.

The conclusions about the "two-step" approach are different. Referring to the last experiment, it is possible to correct the starting matrix working only on the OD pairs that have a greater influence on the bottleneck, obtaining a very good matrix, close to the real one, as show from result of the first level. On the other hand, it is possible to work on this matrix to optimize also the other OD pairs, but the improvement is low and the solution is not satisfactory taking into account the total computational time. On the other hand the first approach shows as it is possible, working only with the most influencing OD pairs, to obtain a result equal to the results obtained with the basic model, but with an inferior error about the seed matrix and the speeds. So, in this approach, two fundamental indicators, the speed and the seed matrix, not directly considered in the goal function, are improved thanks to the use of the "two-step" approach. This is the demonstration that it is possible to work in a first step on the correction of the seed matrix, and only in a second time on the estimation problem. Taking into account the results of the second experiment, and especially for big-sized networks with an large number of OD pairs, it is possible to select a subset of them where to perform OD estimation. Another possibility, as shown in the first implementation of the two-step model, is to work only on the main descent direction to obtain a faster convergence to the solution. It is also evident the requirement to change the optimization method used in the first step. The FDSA obtains a good descent direction, but it is a very slow procedure. One solution is to use the Gauss Newton itself in the first level of the model.

CONCLUSIONS AND FUTURE RESEARCH

The main goal of the present paper is to propose a method for determining a starting demand that, when utilized in the dynamic demand estimation problem, improves the accuracy of the estimated matrix in reproducing the correct traffic regime on the network.

In this paper, different deterministic and stochastic solution procedures commonly adopted in literature are firstly presented and tested for the off-line dynamic demand estimation on the real case study of the inner ring of Antwerp in Belgium.

Both the deterministic and the stochastic procedures underline the same problem at the end of the estimation: an offset in the representation of the congestion pattern, with high differences in the congestion recovery part. This result leads to think that the final error is not related to the model adopted, but to the specific case study and in particular of the specific seed matrix adopted so highlighting the importance of a proper starting point.

Following a two-steps procedure, the seed matrix was modified to obtain a new dynamic starting matrix for the estimation problem. Specifically, the first step of the procedure focused on the optimization of a subset of OD variables, adopting two different approaches: the first approach considered as variables only those relative to ODs that generate the higher flows; while in the second approach only ODs generating bottlenecks on the network are taken into account. In the second step, the optimization works on all the OD pairs, using as starting matrix the matrix derived from the first step. Using the new starting matrix from step 1 implies results that differ according to the adopted approach: specifically, using the deterministic method in the two-steps procedure it was possible to obtain better solutions with the same of the goal function and with a reduction of the computational time. It was also possible to obtain a better result, with an higher improvement of the goal function, but with an increment in the computational time.

The most important result, however, is that it is possible to improve the quality of the estimated matrix without introducing new measurements or developing new models, but only working in different ways on the different OD pairs. In conclusion, it is necessary to highlight that using a two-step method it is possible to combine different kind of models, using not only path-search methods, but combining also random search and pattern search methods, based on the specific configuration of the network and of the problem.

Future developments will deal with more complex networks, because the case study focuses on highways, so the problem results quite simple respect to e.g. an urban network. With such a type of simplified network, the actual results are not sufficient to understand if this method is applicable to every other type of network. Moreover the goal function takes into account only the link flows, so it is necessary to understand if the method confirms the same features also if other measurements, more representative of the congestion state, are considered inside the goal function. Finally it is important not only to understand on which OD pairs is preferable to work, but also to develop a proper goal function that could take into account other information on the real matrix as a good OD trips distribution.

ACKNOWLEDGEMENTS

This study is partially funded by COST] Action TU1004.

REFERENCES

[1] Cascetta E., Inaudi D. and G. Marquis (1993). Dynamic Estimators of Origin-Destination Matrices using Traffic Counts. *Transportation Science 27*, 363-373.

[2] Zhou, X., Lu, C., Zhang, K. (2012). Dynamic Origin-Destination Demand Flow Estimation Utilizing Heterogeneous data sources under Congested Traffic Conditions, Available online at: http://onlinepubs.trb.org/onlinepubs/conferences/2012/4thITM/Papers-A/0117-000097.pdf. Accessed January 2013.

[3] Cipriani E., Gemma A., Nigro M. (2013). A bi-level gradient approximation method for dynamic traffic demand estimation: sensitivity analysis and adaptive approach. Proceedings of the IEEE Conference on Intelligent Transportation Systems, 16th IEEE ITSC, 2013

[4] Cascetta, E., Papola, A., Marzano, V., Simonelli, F. and I. Vitiello (2013). Quasi-dynamic estimation of o-d flows from traffic counts: formulation, statistical validation and performance analysis on real data. *Transportation Research Part B*, doi 10.1016/j.trb.2013.06.007.

[5] Frederix, R., Viti, F., Corthout, R., Tampère, C. M. J. (2011). New Gradient Approximation Method for Dynamic Origin-Destination Matrix Estimation on Congested Networks. *Transportation Research Record, 2263*:19-25.

[6] Toledo, T., Kolechkina, T. (2012). Estimation of Dynamic Origin–Destination Matrices Using Linear Assignment Matrix Approximations. IEEE Transactions On Intelligent Transportation Systems. Digital Object Identifier 10.1109/TITS.2012.2226211.

[7] Cremer, M. and H. Keller (1984). A systems dynamics approach to the estimation of entry and exit o-d flows. Proceedings of 9th ISTTT.

[8] Zhou X and H. Mahmassani (2007). A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework. Transportation Research B, 41, 823-840

[9] Ashok, K. and M.Ben-Akiva (2000). Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows. *Transportation Science 34*, 21-36.

[10] Balakrishna, R., Koutsopoulos, H. N. and M. Ben-Akiva (2005). Calibration and validation of dynamic traffic assignment systems. *Proceedings of 16th ISTTT*, 407-426.

[11] Antoniou C., Balakrishna R., Koutsopoulos H.N. and M. Ben-Akiva (2009). Off-Line and On-Line Calibration of Dynamic Traffic Assignment Systems. Presented at the 12th IFAC Symposium on Control in Transportation Systems.

[12] Cipriani E., Florian M., Mahut M. And M. Nigro (2011). A gradient approximation approach for adjusting temporal origin–destination matrices. *Transportation Research C*, 19(3), 270-282

[13] Balakrishna, R. (2006). Off-line calibration of dynamic traffic assignment models. PHD thesis. Massachusetts Institute of Technology.

[14] Dixon, M. and L.R. Rilett (2002). Real-Time OD Estimation Using Automatic Vehicle Identification and Traffic Count Data. *Computer-Aided Civil and Infrastructure Engineering 17 (2002),* 7–21

[15] Eisenman, S.M., List, G.F. (2004). Using probe data to estimate OD Matrices. *Intelligent Transportation Systems Conference. Washington DC October 3-6*: 291-296.

[16] Caceres, N., Wideberg, J.P.and F.G. Benitez (2007). Deriving origin–destination data from a mobile phone network. *IET Intell Transp. Syst., 2007, 1, (1)*, pp. 15–26.

[17] Barceló, J., Montero, L., Bullejos, M., Serch, O., Carmona, C. (2012). Dynamic OD Matrix Estimation Exploiting Bluetooth Data in Urban Networks, Recent Researches in Automatic Control and Electronics, *ISBN*: 978-1-61804-080-0.

[18] Mitsakis, E., Salanova, J.M., Chrysohoou, E., Aifadopoulou, G.(2013). A robust method for real-time estimation of travel times for dense urban road networks using point-to-point detectors. Proceedings of the 92nd Annual Meeting in Transportation Research Board, TRB 2013.

[19] Iannò, D., Postorino, M.N. (2002). A generation constrained approach for the estimation of O/D trip matrices from traffic counts.

[20] Bierlaire M., F. Crittin (2004). An Efficient Algorithm for Real-Time Estimation and Prediction of Dynamic OD Tables. *Operations research Vol. 52*, No. 1, pp. 116–127.

[21] Cipriani, E., Nigro, M., Fusco, G., Colombaroni, C. (2013). Effectiveness Of Link And Path Information On Simultaneous Adjustment Of Dynamic O-D Demand Matrix. *European Transport Research Review.* DOI: 10.1007/s1254

[22] Cascetta, E., Nguyen, S., 1988. "A unified framework for estimating or updating origin/destination matrices from traffic counts," *Transportation Research Part B: Methodological, Elsevier*, vol. 22(6), pages 437-455, December.4-013-0115-z

[23] Tavana, H. (2001) Internally-Consistent Estimation of Dynamic Network OriginDestination Flows from Intelligent Transportation Systems Data Using Bi-Level Optimization. Ph.D. Dissertation, The University of Texas at Austin.

[24] Frederix, R. Tampère, C. Viti, F. (2011). Dynamic Origin-Destination Matrix Estimation in congested networks: Theoretical findings and implication in practice. *Transportmetrica*.

[25] Lindveld, K., (2003). Dynamic O-D Matrix estimation: a behavioral approach. Ph.D. thesis, Delft University of Technology.

[26] Spall, J. (2012). Stochastic Approximation. Handbook of Computational Statistic:Concepts and Methods (2nded.) (J.Gentle, W. Hardle, Y. Mori, eds.), Springer-Verlang, Heidenberg, Chapter 7, pp. 173-201.

[27] Spall, J. (2000). Adaptive Stochastic Approximation by the Simultaneous Perturbation Method. *IEEE transactions on automatic control*, Vol. 45, No. 10

[28] Spall, J. (1998). An Overview of the Simultaneous Perturbation Method for Efficient Optimization. *Johns Hopkins Apl Technical Digest*, Volume 19, Number 4, pp 482-492.

[29] Nigro, M.(2009). Correzione della domanda di trasporto in dinamica intraperiodale con l'ausilio di differenti fonti di dati. *PHD thesis*. Università degli studi Roma Tre.

[30] Frederix, R. (2012). Dynamic Origin-Destination Matrix Estimation in Large-Scale congested networks. PHD thesis. Katholieke Universiteit Leuven

[31] Djukic, T., Flötteröd, G., Hoogendoorn, S.P., van Lint, J. W. C., (2013). Efficient real time OD matrix estimation based on Principal Component Analysis. *Intelligent Transportation Systems Conference (ITSC) 2012*