Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 6 (2015) 144 – 157

Transportation Research Procedia

www.elsevier.com/locate/procedia

4th International Symposium of Transport Simulation-ISTS'14, 1-4 June 2014, Corsica, France

Macroscopic traffic flow model calibration using different optimization algorithms

Anastasia Spiliopoulou[a] *, Ioannis Papamichail[a], Markos Papageorgiou[a], Ioannis Tyrinopoulos[b], John Chrysoulakis[b]

[a]Dynamic Systems and Simulation Laboratory, School of Production Engineering and Management, Technical University of Crete, 73100, Chania, Greece

[b]Department of Civil and Infrastructure Engineering, Technological Educational Institute of Athens, 12210, Egaleo, Athens, Greece

Abstract

This study tests and compares different optimization algorithms employed for the calibration of a macroscopic traffic flow model. In particular, the deterministic Nelder-Mead algorithm, a stochastic genetic algorithm and the stochastic cross-entropy method are utilized to estimate the parameter values of the METANET model for a particular freeway site, using real traffic data. The resulting models are validated using various traffic data sets and the optimization algorithms are evaluated and compared with respect to the accuracy of the produced models as well as the convergence speed and the required computation time.

Selection and/or peer-review under responsibility of the Organizing Committee of ISTS'14

Keywords: macroscopic traffic flow models; model calibration; comparison of optimization algorithms.

1. Introduction

During the last decades, several mathematical models for the road traffic flow have been proposed. Depending on the level of detail they use, the models are classified as macroscopic or microscopic (see Hoogendoorn and Bovy (2001) for an overview on traffic flow models). Traffic flow models may be employed for the planning of new, upgraded or modified road infrastructures; as well as for the development and testing of traffic flow estimation algorithms, traffic control strategies and other operational tools (Kotsialos and Papageorgiou, 2000). The models

* Corresponding author. Tel.: +30-28210-37345.
E-mail address: natasa@dssl.tuc.gr

Selection and/or peer-review under responsibility of the Organizing Committee of ISTS'14
doi:10.1016/j.trpro.2015.03.012

include a number of physical or non-physical parameters, with unknown values, which should be appropriately specified, in case of real applications, so as to reproduce the network and traffic flow characteristics with the highest possible accuracy. The macroscopic traffic flow models include a lower number of parameters compared to microscopic models; also, they have an analytical form, which allows their usage for various significant traffic engineering tasks (estimation, control strategy design) beyond simulation. Before employing a traffic flow model in practice, it is important to first calibrate it against real traffic data. The calibration procedure aims to appropriately specify the model parameter values, so that the representation of the network and traffic flow characteristics is as accurate as the model structure allows. The most common approach is to minimize the discrepancy between the model's estimation and the real traffic data, by use of appropriate optimization tools.

Within the, quite limited, literature on macroscopic traffic flow model calibration, various methods have been employed to solve the parameter estimation problem. In particular, Grewal and Payne (1976) utilized the least-squares method and an extension of Kalman filter; Michalopoulos et al. (1993) and Helbing (1996) have applied some trial-and-error method, Cremer and Papageorgiou (1981), Cremer and May (1986), Papageorgiou et al. (1989), Sanwal et al. (1996), Kotsialos et al. (2002) and Monamy et al. (2012) have used the deterministic Complex algorithm of Box (1965). Ngoduy et al. (2004) and Spiliopoulou et al. (2014) have employed the deterministic Nelder-Mead algorithm (Nelder and Mead, 1965), which is, actually, similar to the above mentioned Complex algorithm. Poole and Kotsialos (2012) utilized stochastic genetic algorithms and Ngoduy and Maher (2012) the stochastic cross-entropy method (de Boer et al., 2005). Surprisingly, though, there is no work addressing the suitability and effectiveness of the employed optimization algorithms for this specific parameter estimation problem. Thus, the goal of this study is to test and compare various optimization algorithms, both stochastic and deterministic, for the calibration of a macroscopic traffic flow model, namely the METANET model (Messmer and Papageorgiou, 1990; Papageorgiou et al., 2010), using real traffic data from a freeway stretch. In particular, the optimization algorithms that will be considered are the deterministic Nelder-Mead algorithm, the stochastic genetic algorithms and the stochastic cross-entropy method. These algorithms will be compared in terms of optimum cost function value, computation time and accuracy of the produced model.

The paper is organized as follows: Section 2 presents the selected macroscopic traffic flow model and the unknown model parameters to be estimated. Section 3 describes the model calibration procedure which is formulated as a least-squares minimization problem; and Section 4 presents the considered optimization algorithms that are employed to solve the parameter estimation problem. Section 5 describes the considered freeway stretch and the traffic data used in the current investigations; followed by Section 6 which includes the calibration results and the comparison of the employed optimization algorithms. Finally, Section 7 summarizes the main conclusions and remarks of the study.

2. Macroscopic traffic flow model

The macroscopic traffic flow model METANET (Messmer and Papageorgiou, 1990; Papageorgiou et al., 2010) will be calibrated for a particular freeway site. Within METANET model the freeway is divided into homogeneous, consecutively numbered sections i, with respective lengths L_i and number of lanes λ_i, as shown in Fig. 1. The time is also discretized into uniform intervals of duration T. For each discrete time k=0,1,…,K, the model calculates at each section i, the density, the flow and the mean speed according to the following equations:

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{L_i\lambda_i}\left[q_{i-1}(k) - q_i(k)\right] \tag{1}$$

$$q_i(k) = v_i(k)\rho_i(k)\lambda_i \tag{2}$$

$$v_i(k+1) = v_i(k) + \frac{T}{L_i}v_i(k)\left[v_{i-1}(k) - v_i(k)\right] + \frac{T}{\tau}\left[V^e\left(\rho_i(k)\right) - v_i(k)\right] - \frac{\nu T\left[\rho_{i+1}(k) - \rho_i(k)\right]}{\tau L_i\left[\rho_i(k) + \kappa\right]} \tag{3}$$

where τ (a time constant), ν (an anticipation constant) and κ are model parameters. The function $V^e(\rho_i(k))$ corresponds to the fundamental diagram and is calculated as follows:

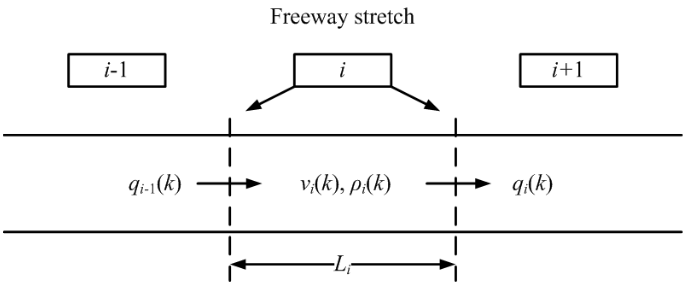

Fig. 1 Freeway discretization within METANET.

$$V^e\left(\rho_i(k)\right) = v_{f,i} \exp\left[-\frac{1}{a_i}\left(\frac{\rho_i(k)}{\rho_{cr,i}}\right)^{a_i}\right] \tag{4}$$

where $v_{f,i}$ is the free speed, $\rho_{cr,i}$ is the critical density (for which the flow at section i is maximized) and α_i is a further model parameter for section i. Moreover, the mean speed calculated by the model should not exceed a minimum value $v_{\min}$. In Papageorgiou et al. (1989), two additional terms were proposed for more accurate modeling of merging and lane drop phenomena. In particular, the impact on mainstream speed due to an on-ramp merging flow is considered by adding the term $-\delta T q_\mu(k) v_i(k) \big/ L_i \lambda_i \left(\rho_i(k) + \kappa\right)$ into (3) for the merging section, where δ is a model parameter and q_μ is the inflow from the on-ramp. Moreover, in order to take into account the impact on speed due to intensive lane-changing at lane-drop areas, the term $-\varphi T \Delta\lambda \rho_i(k) v_i(k)^2 \big/ L_i \lambda_i \rho_{cr,i}$, is added to (3) for the section immediately upstream of the lane drop, where φ is a model parameter and $\Delta\lambda$ is the number of dropped lanes.

At bifurcation locations (e.g. off-ramps), the inflow is split to the exiting sections according to known turning rates $\beta_j(k)$. In addition, at bifurcation locations, a downstream density $\rho_{i+1}(k)$ is needed in (3) for the section i entering the bifurcation; this density reflects the upstream influence of the downstream traffic conditions. However, as we have at least two downstream sections at bifurcations, the following formula was proposed (Messmer and Papageorgiou, 1990) for usage:

$$\rho_{i+1}(k) = \sum_{\mu \in O_i} \rho_\mu^{\,2}(k) \Big/ \sum_{\mu \in O_i} \rho_\mu(k) \tag{5}$$

where $\rho_{i+1}(k)$ is the virtual density downstream of section i, which is used in (3) of section i; and $\rho_\mu(k)$ is the density of each section downstream of section i, O_i being the set of exiting sections. The quadratic average used in (5) accounts for the fact that congestion may spill back to a section i from any of its downstream sections (e.g., in case of spillback from a saturated off-ramp), even if the rest downstream sections are not congested. Notice that (5) does not include any parameter to be calibrated.

As presented above, the model includes a number of parameters, whose values may differ for different freeway sites and depend on factors such as network geometry, driver behavior, percentage of trucks, weather conditions etc. Thus, the reliability and accuracy of the model depends on the appropriate specification of its parameter values; and hence a calibration exercise is required before using the model in a potential real application.

3. Model calibration procedure

The model parameter calibration (or parameter estimation) procedure aims at enabling a macroscopic traffic flow model to represent traffic conditions of a freeway network with the highest achievable accuracy. The estimation of the unknown model parameters is not a trivial task, since the system equations are highly nonlinear in both the parameters and the state variables. Consider that a macroscopic discrete-time state-space model is described by the following state equation,

$$\begin{aligned} &\mathbf{x}(k+1) = f\left[\mathbf{x}(k),\mathbf{u}(k),\mathbf{d}(k),\mathbf{p}\right] \qquad k = 0,1,\ldots,K-1 \\ &\mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \tag{6}$$

where k is the discrete time index; $\mathbf{x}$ stands for the state vector, $\mathbf{u}$ is the control vector, $\mathbf{d}$ is the disturbance vector and $\mathbf{p}$ is the parameter vector. METANET model can readily assume the state-space form of (6) for any freeway network. In particular, the state vector $\mathbf{x}$ includes the section densities and mean speeds, the control vector $\mathbf{u}$ contains the implemented control measures (e.g. variable speed limits), the external variable vector $\mathbf{d}$ consists of the origin speeds and inflows, the turning rates at bifurcations, and the downstream densities; and $\mathbf{p}$ includes the unknown model parameters that need to be specified.

If the initial state $\mathbf{x_0}$ is given and the external vectors $\mathbf{u}(k)$ and $\mathbf{d}(k)$ are known over a time horizon k=0,…,K−1, then the parameter estimation problem can be formulated as a nonlinear least-squares output error problem which aims at the minimization of the discrepancy between the model calculations and the real traffic data by use of the following cost function,

$$J(\mathbf{p}) = \sqrt{\frac{1}{K}\sum_{k=0}^{K-1}\left[\mathbf{y}(k)-\mathbf{y}^{\mathbf{m}}(k)\right]^2} \tag{7}$$

subject to (6); where $\mathbf{y}(k)$=$\mathbf{g}[\mathbf{x}(k)]$ is the measurable model output vector (typically consisting of flows and mean speeds at various network locations) and $\mathbf{y}^{\mathbf{m}}(k)$ includes the real measured traffic data (consisting of flows and speeds at the corresponding network locations). The model parameter values are selected from a closed admissible region of the parameter space, which may be defined on the basis of physical considerations. The determination of the optimal parameter set must be performed by means of a suitable nonlinear programming routine, whereby for each choice of a new parameter vector $\mathbf{p}$, the value of the performance index (PI) (7) may be computed by a simulation run of the model equations as shown in Fig. 2.

The nonlinear, non-convex least-squares optimization problem of parameter calibration is known to have multiple local minima (see Ngoduy and Maher (2012) for an illustration), and hence gradient-based optimization algorithms are not an option. In previous calibration studies, see Section 2 for an overview, various derivative-free optimization algorithms have been employed to solve the parameter estimation problem, without though justifying the reason for selecting the particular algorithms. Within this study various optimization algorithms are tested and compared in order to investigate which optimization methodology is more suitable for the problem of macroscopic traffic flow model calibration.

4. Optimization algorithms

Three derivative-free optimization algorithms are employed to solve the parameter estimation problem; in particular, the deterministic Nelder-Mead algorithm, a stochastic genetic algorithm and the stochastic cross-entropy method. In the following, the selected algorithms are shortly described along with their potential advantages and weak points.

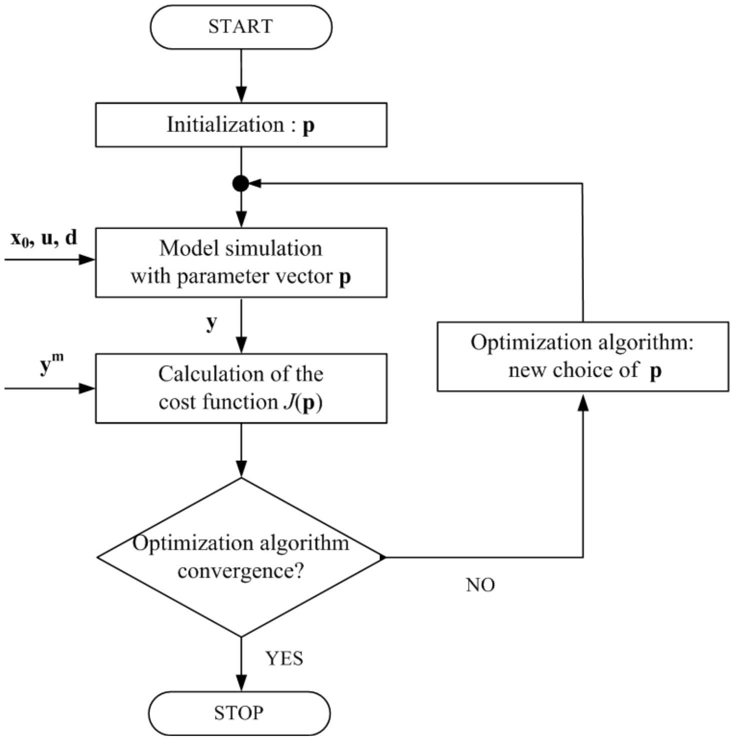

Fig. 2 Model calibration procedure.

4.1. Nelder-Mead algorithm

The Nelder-Mead method (Nelder and Mead, 1965; Lagarias et al., 1998) is one of the best known algorithms for multidimensional unconstrained optimization. The method does not require any derivative information, which makes it suitable for problems with non-linear, discontinuous or stochastic cost function.

The method uses a simplex, i.e. an n-dimensional geometrical shape with $n+1$ vertices. Every vertex x_i , where $i = 1,...,n+1$, corresponds to a potential solution which in turn corresponds to a cost function value, $f(x_i)$. The algorithm starts with an initial working simplex and then performs a sequence of transformations aiming at reducing the cost function value at its vertices. In particular, at each iteration the algorithm orders the simplex's vertices with respect to the corresponding cost function values e.g. $f(x_1) \le f(x_2) \le ... \le f(x_{n+1})$ and calculates the centroid x_c of all vertices excluding the worst vertex x_{n+1}. Then, it computes the new working simplex from the current one as follows. First, an attempt is made to replace only the worst vertex x_{n+1} with a better point by using *reflection*, *expansion* or *contraction*. If this succeeds, the accepted point becomes the new vertex of the working simplex. Otherwise, the algorithm *shrinks* the simplex towards the best vertex x_1 . In this case, n new vertices are computed. Simplex transformations are controlled by four parameters: α for *reflection*, β for *contraction*, γ for *expansion* and δ for *shrinkage*. The above procedure continues until the working simplex becomes sufficiently small or when the function values $f(x_i)$ are close enough to each other.

In contrast to many other direct search methods which call, at each iteration, for multiple cost function evaluations, Nelder-Mead typically requires only one or two function evaluations, except when performing the shrinkage transformation which is, actually, quite rare in practice. As a result, it often gives significant ameliorations of the cost function value quite fast. On the other hand, in some cases the method may perform a large number of iterations without significant improvement of the cost function value. To cope with this problem, restarting the algorithm several times, with reasonably small number of allowed iterations per each run may be a heuristic solution. Moreover, the evolution of the working simplex and the produced best solution are dependent on the initial working simplex, since the algorithm searches for new points using the vertices of the working simplex. To face this fact, multiple algorithm runs should be carried out using different initial vertices for the working simplex.

4.2. Genetic Algorithm (GA)

A genetic algorithm (Goldberg, 1989; Holland, 1992) is a heuristic search method which belongs to the larger class of evolutionary algorithms. GA mimics the process of biological evolution and uses techniques inspired by natural selection, mutation and crossover. It is suitable for a variety of optimization problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear.

The method uses a population of candidate solutions to an optimization problem and evolves it towards better solutions. The evolution starts from an initial population of randomly generated individuals (solutions) which are evaluated through the cost function (fitness). At each iteration, called generation, the algorithm selects individuals (parents) from the current generation and uses them to produce the individuals (offspring) for the next population. To do so, the GA uses three main types of rules:

- *Selection rules* select individuals (parents), with probabilities based on their fitness, that contribute to the population of the next generation. Some of the individuals in the current population that have best fitness are chosen as *elite*. These elite individuals are passed directly and unchanged to the next population.
- *Crossover rules* combine (random) couples of parents to form offspring for the next generation, thus exchange information between two candidate solutions.
- *Mutation rules* apply random changes to individual parents, which may introduce new features (i.e. new parameter space regions) to the population.

Through the stochastic operations of *selection*, *crossover* and *mutation,* the population "evolves" towards better solutions, over successive generations, and the algorithm stops when one of the stopping criteria is met.

The main advantage of GA is its flexibility to search complex solution spaces; thanks to its stochastic operations it is less likely to restrict the search to a local minimum area in contrast to point-to-point movement optimization techniques. On the other hand, each iteration requires as many cost function evaluations as the population size, which increases substantially the computational cost, especially for problems with computationally expensive cost function or problems which require large population size. Nevertheless, since the evaluation of the cost function for each individual is independent of all others, the parallelization of GA is an option. Finally, it is important to tune the algorithm’s parameters, i.e. the *population size*, the *elite rate*, the *crossover probability* and the *mutation rate* in order to find appropriate settings for the problem being examined.

4.3. Cross-entropy method (CE method)

The cross-entropy method (Rubinstein and Kroese, 2004; de Boer et al. 2005) is a general Monte-Carlo approach to combinatorial and continuous multi-extremal optimization and importance sampling. The method originates from the field of rare event simulation, where very small probabilities need to be accurately estimated.

The algorithm starts from an initial population of potential solutions generated using a continuous, usually uniform, distribution f_0. At each iteration *k*, the solutions are evaluated through the cost function and sorted into ascending order; and the best ρ% solutions comprise the elite sample. The corresponding probability density function, $\hat{g}_k$, of this elite sample is estimated, e.g. using a Kernel density estimator as proposed by Ngoduy and Maher (2012), and the probability distribution of the population is updated using the equation:

$$\hat{f}_{k+1} = \hat{f}_k (1-a) + \hat{g}_k a \tag{8}$$

where *a* is a smoothing parameter, typically in the range [0.7, 0.9]. The updated density equation $\hat{f}_{k+1}$ is used in the next iteration to generate the new random sample of solutions; the algorithm continues leading, over iterations, to more spiked shapes of the population probability distribution and it stops when one of the stopping criteria is met.

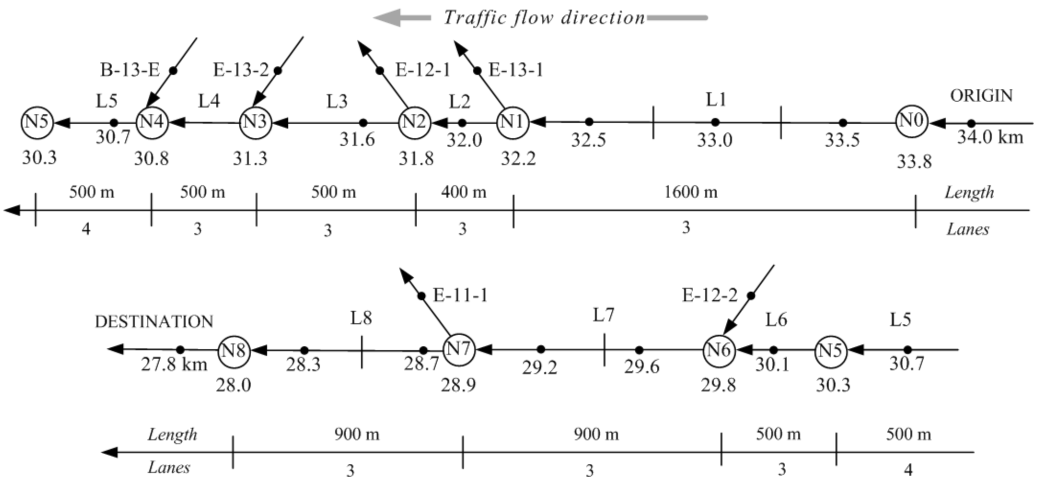

Fig. 3 Representation of the considered freeway stretch within METANET.

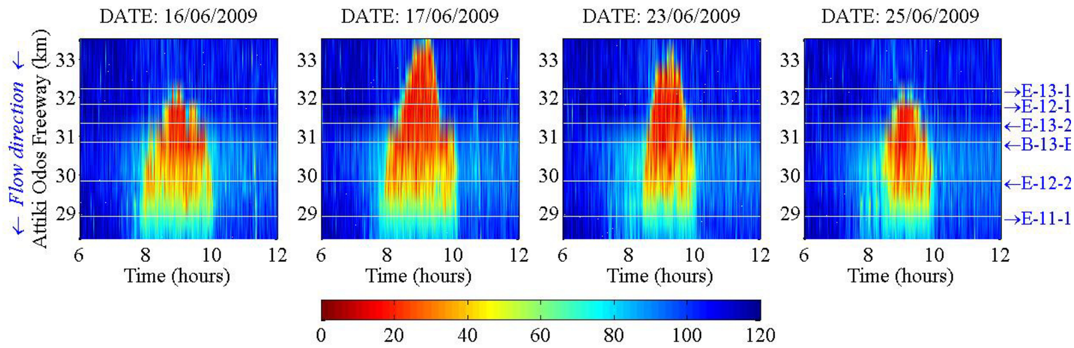

Fig. 4 Time-space diagram of measured speed at the considered freeway stretch for four different days.

As with the previous algorithms, the CE method does not require any derivative information, thus it may be applied to problems where the objective function is discontinuous, non-differentiable or highly nonlinear. In contrast to other stochastic methods, the selection of the potential solutions is not a completely random process since the utilized distribution is affected by the best solutions of each iteration. The main disadvantage of the method is that it requires as many cost function evaluation as the size of the population, resulting to large computational cost and slow convergence. Again, it is important to tune the algorithm's parameters, i.e. the *population size*, the elite rate ρ and the smoothing parameter a in order to find appropriate settings for the problem being examined.

5. Test site and traffic data

The test network considered in this study is a part of Attiki Odos freeway (34th to 28th km, direction Airport to Elefsina) in Athens, Greece. This freeway stretch includes three on-ramps and three off-ramps, as shown in Fig. 3. In order to model the network by use of METANET, the freeway stretch is represented through 9 nodes (N0−N8, see Fig. 3) and 8 links (L1−L8, see again Fig. 3), where each node corresponds to a bifurcation or a junction or any location marking a change of the network geometry; whereas the homogeneous road stretches between these locations are represented by links. Each network link is subdivided in sections of equal length; see for example link L1 which is divided in 3 sections, with the vertical short lines denoting the section borders. Fig. 3 displays the

Table 1 Optimal model parameter sets estimated by use of different optimization algorithms.

Optimization algorithm	**Model parameters**						
	τ **(s)**	ν **(km^2/h)**	δ **(h/km)**	φ **(h/km)**	ρ_{cr} **(veh/km/lane)**	a	v_f **(km/h)**
Nelder-Mead algorithm	18.6	24.5	1.2	1.1	35.5	1.5	117.8
Genetic algorithm	18.1	21.1	0.2	1.5	36.2	1.4	118.1
Cross-entropy method	27.2	33.1	0.5	1.0	34.4	1.5	118.8

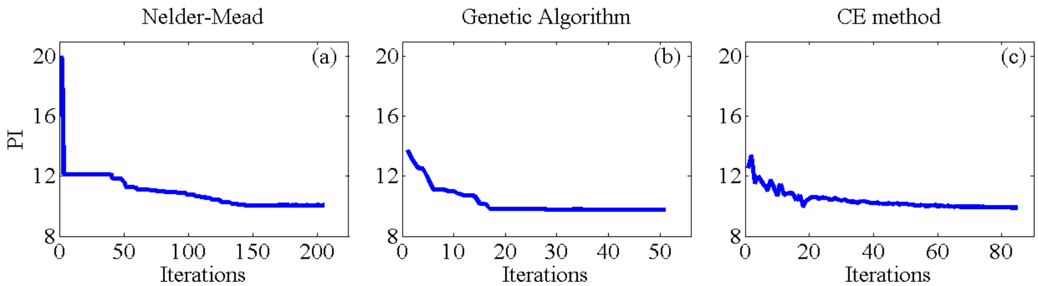

Fig. 5 Performance index value over iterations during the model calibration procedure using (a) the Nelder-Mead algorithm, (b) the genetic algorithm and (c) the cross-entropy method.

length, number of sections and number of lanes for each link, the exact location of the on-ramps and off-ramps, as well as the location of the available detector stations which are depicted by bullets.

The real traffic data used in this study were provided by ATTIKES DIADROMES S.A., which is the freeway operating company. In particular, the provided traffic data includes flow and speed measurements at the corresponding detector station locations, with a time resolution of 20 seconds, for the time period of June 2009. The traffic data analysis showed that, within this particular freeway stretch, recurrent traffic congestion is formed during the morning peak hours. Fig. 4 illustrates the space-time diagram of real speed measurements for 4 different days: 16/06/2009, 17/06/2009, 23/06/2009 and 25/06/2009. It is observed that congestion is created during 8−10 a.m.; the congestion originates at the 29th km of the freeway stretch and spills back several kilometers upstream, up to the 32nd km, and on some days up to the 33rd km. Figure 3 shows that the congestion creation area is actually a diverge area, with the off-ramp E-11-1 receiving high exit flow during the morning peak hours, according to real traffic data. The high exit flow rate, in combination with the limited capacity of the off-ramp, leads to the creation of congestion, which propagates upstream for several kilometers on the freeway mainstream. The test network and traffic data presented above, are used to calibrate and validate the METANET traffic flow model. It should be noted that the main criterion for selecting these 4 days was that, during the morning hours 6-12 a.m., no incident and no detector failure occurred at the examined freeway stretch, which can, of course, not be reproduced by any traffic flow model.

6. Calibration and validation results

The calibration procedure, as described in Section 3, was applied to METANET model using real traffic data from 16/06/2009 and a simulation step equal to T = 5s. The model parameter vector consists of the free flow speed v_f, the critical density ρ_{cr} and the parameters α, τ, ν, δ and φ which are common for all the freeway sections. Thus, one single fundamental diagram is considered for all freeway sections. Moreover, the model includes two extra parameters which are known from previous validation exercises to be of minor importance and are, therefore, given constant values, in order to reduce the dimension of the parameter vector. In particular, κ is set equal to 10

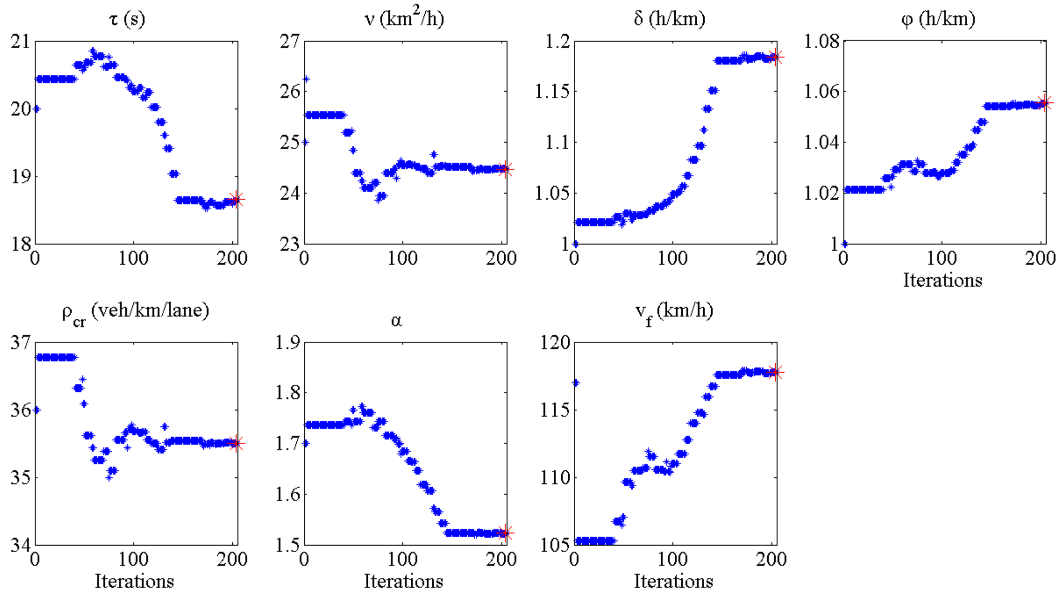

Fig. 6 Evolution of the model parameter values over iterations, for the best vertex of the working simplex, during the application of the Nelder-Mead algorithm.

veh/km/lane and v_{min} is set to 7 km/h. Furthermore, the utilized Performance Index (PI) (see (7)) includes the model estimation of speed and the real speed measurements at the corresponding detector station locations (see Fig. 3).

All simulations were performed using a desktop computer with 2.4 GHz CPU and 2.0 GB of RAM. The calibration procedure, including the traffic flow model and the optimization algorithms, has been programmed in MATLAB (R2010a). In the following sections, the optimization results of each utilized algorithm are presented first, followed by the evaluation of the algorithms' performance using various criteria. It should be noted that, for each utilized algorithm, various calibration tests were carried out using different values for the algorithms' parameters and only the best obtained results are presented here below.

6.1. Nelder-Mead algorithm (NM)

The Nelder-Mead algorithm was employed using the following parameters: $\alpha = 1$, $\beta = 2$, $\gamma = -0.5$ and $\delta = 0.5$ (see Section 4.1 for a description of the algorithm's parameters). Moreover the utilized termination criteria are the cost function convergence or the working simplex convergence, with tolerance equal to 0.1 and the maximum allowed number of iterations which was set equal to 1000.

Table 1 includes the optimal parameter values estimated by use of the Nelder-Mead algorithm. Figure 5a presents the convergence of the algorithm, which achieves a PI value equal to 10.1 after 204 iterations. It is observed that, although the algorithm starts from a high PI value, it achieves a significant improvement already in the first iterations. Figure 6 displays the evolution of the model parameter values, for the best vertex, over iterations. It may be seen that the algorithm converges to an optimal parameter set by searching, for new better solutions, around the vertices of the working simplex.

6.2. Genetic algorithm

The genetic algorithm was employed with *population size* equal to 500, *elite rate* equal to 0.01, *crossover rate* equal to 0.8 and *mutation rate* equal to 0.1. The termination criteria utilized are again the cost function convergence and the maximum allowed number of iterations (generations) which was set equal to 1000.

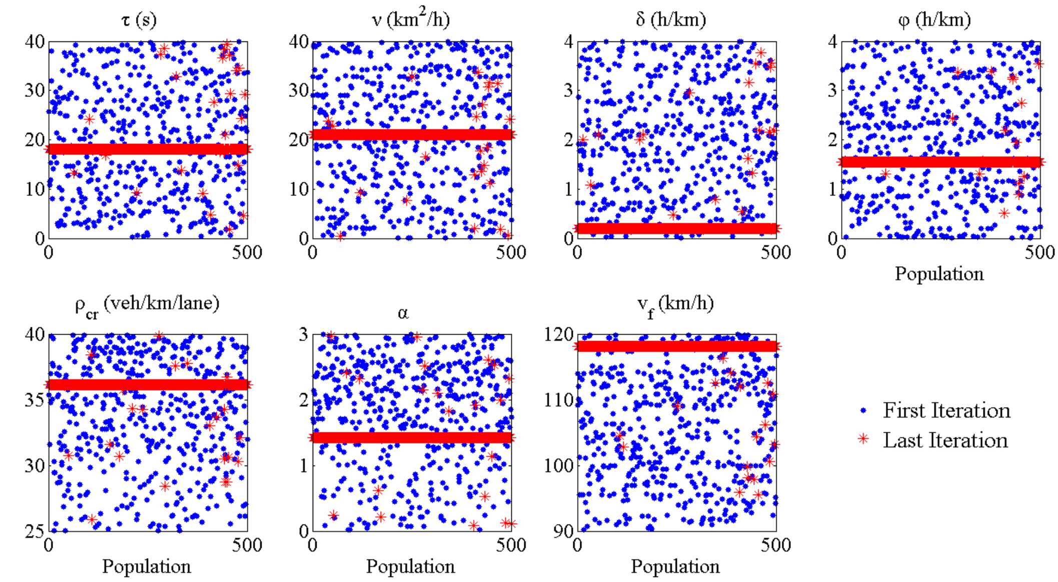

Fig. 7 Model parameter values for each individual of the population, in the first and the last iteration of the genetic algorithm.

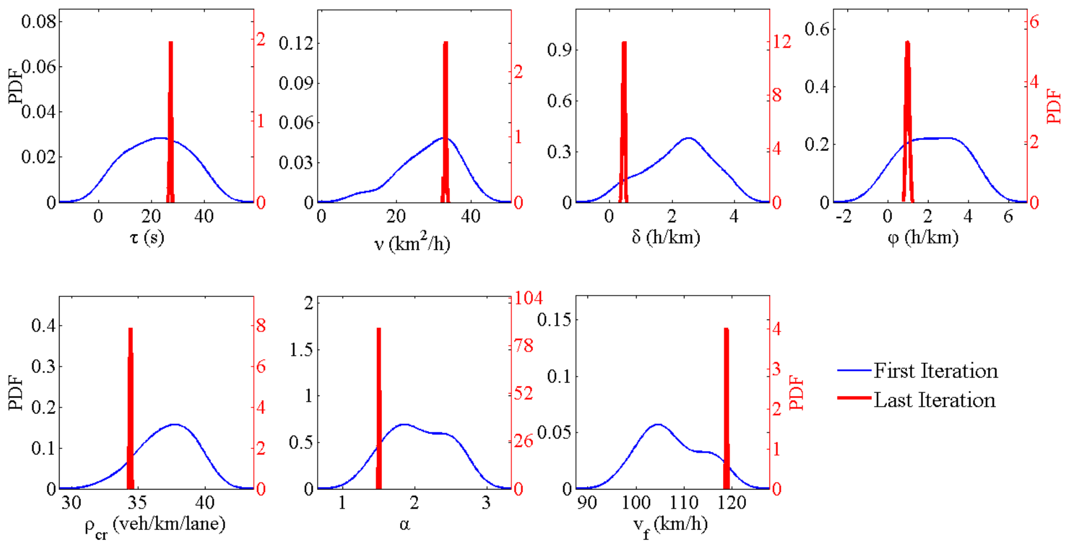

Fig. 8 Probability density function of each model parameter, in the first and the last iteration of the cross-entropy method.

Table 1 presents the optimal parameter set estimated using the genetic algorithm. Figure 5b displays the convergence of the algorithm over iterations, which achieves a PI value equal to 9.8 after 51 iterations. It is observed that the algorithm achieves a low PI value even from the first iteration thanks to the fact that it searches, simultaneously, at a large number of points within the solution space. Figure 7 presents the corresponding model parameter values included in the individuals of the population in the first and in the last iterations. It may be seen that, in the beginning of the calibration, the population includes randomly generated individuals; over the iterations the population evolves towards better solutions and in the very last iteration the majority of the population individuals are virtually identical (which results in much less red dots being visible).

Table 2 Performance index value for each optimal parameter set for four different dates.

Optimization algorithm	Validation results (PI)				
	16/06/2009	**17/06/2009**	**23/06/2009**	**25/06/2009**	**Average**
Nelder-Mead algorithm	10.1	10.9	12.4	8.4	10.5
Genetic algorithm	9.8	10.0	11.8	9.1	10.2
Cross-entropy method	9.9	10.6	12.4	8.3	10.3

Table 3 Optimization algorithms' performance.

Optimization algorithm	**Total number of iterations**	**Total number of function evaluations**	**Computation time (min)**
Nelder-Mead algorithm	204	317	0.5
Genetic algorithm	51	25500	122.9
Cross-entropy method	85	42500	197.8

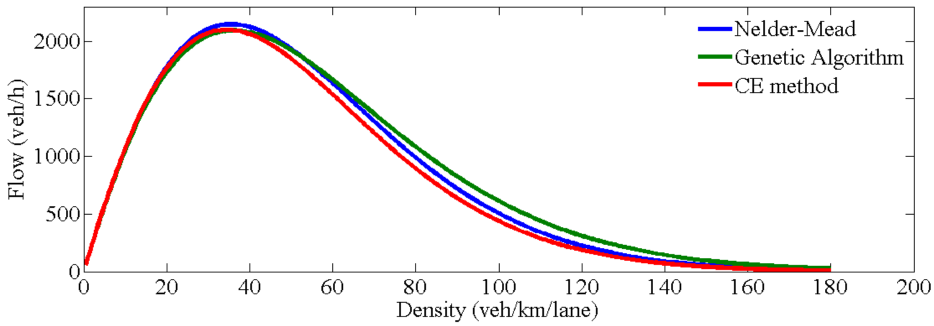

Fig. 9 Estimated fundamental diagram (FD) for all three obtained models.

6.3. *Cross-entropy method (CE method)*

The cross-entropy method was applied using *population size* equal to 500, *elite rate* 0.05 and smoothing parameter a equal to 0.8. The utilized termination criteria are the bandwidth of the kernel estimation function, which was set equal to 0.1 and the maximum allowed number of iterations which was set to 1000.

Table 1 includes the optimal parameter set estimated by the algorithm. Figure 5c presents the convergence of the algorithm, which achieves a PI value equal to 9.9 after 85 iterations. As with the previous algorithm, the CE method achieves a low PI value already at the first iteration thanks to large number of potential solutions considered within the search space.

Figure 8 illustrates the probability density function (PDF) calculated for each model parameter, using all population solutions, at the first and the last iteration. It is observed that in the beginning of the calibration the PDFs of the model parameters are very smooth; over iterations, the shape of the PDFs is updated until it becomes very spiked, concentrated around the optimal parameter values.

6.4. Comparison of the algorithms' performance and results

As presented above, the calibration of METANET model for the particular freeway site, using three different optimization algorithms, resulted in three different model parameter sets, with similar but not the same parameter values (see Table 1). Table 1 indicates the very close proximity of, particularly, the optimal parameter values which are involved in the fundamental diagram (FD) function (4); Figure 9 traces the 3 respective FDs, indicating that they are virtually identical. On the other hand, it is known from previous model validation work (e.g. Papageorgiou et al. 1990) that the calibration PI features low sensitivity around the optimum if the parameters v and τ are changing values simultaneously. This is confirmed with the results of Table 1, where the ratio v/τ may be calculated to be 1.32, 1.17, 1.22 for the three respective optimization methods, despite the stronger deviation of the underlying absolute parameter values.

The resulting traffic flow models should reflect reliably the traffic characteristics of the considered network, thus they should be able to reproduce its typical traffic conditions. In order to test the accuracy and robustness of the produced models, the models are validated, i.e. are applied using different traffic data sets (from the same freeway stretch) than the ones used for their calibration. To this end, the models were applied using traffic data from 17/06/2009, 23/06/2009 and 25/06/2009. Table 2 presents the validation results in terms of PI values for all three models and all utilized traffic data sets. Moreover, Fig. 10 presents the space-time diagrams of the real measured speeds and the models' estimation of speed for all considered dates. It is observed that all models are able to reproduce the traffic conditions of other days with sufficient accuracy, achieving low PI values for both the calibration and the validation dates.

Apart from the quality of the produced model, the decision on the optimization algorithm to be employed for the calibration of a macroscopic traffic flow model, should also take into account other criteria. Table 3 includes, for each utilized optimization algorithm, the total number of iterations, the total number of cost function evaluations and the computation time. It is observed that, although the Nelder-Mead algorithm took a large number of iterations to converge, it required much less cost function evaluations, only 317, compared to the genetic algorithm and the CE method which needed 25500 and 42500, respectively. As a result, the computation time of the Nelder-Mead algorithm is considerably lower compared to the other two algorithms, since it converged in just 0.5 min, in contrast to the other algorithms which required 122.9 min and 197.8 min, respectively. Finally, comparing the genetic algorithm with the CE method it may be seen that, although the algorithms utilized the same population size, the CE method took 34 more iterations to converge, thus a higher number of cost function evaluations and bigger computation time.

Considering the above, all three optimization algorithms are able to estimate a robust model parameter set. Nevertheless, while employing an optimization algorithm, the available time for calibration should be taken into account as well as the required time for the cost function evaluation (e.g. here, one simulation run of the model). Note that multiple calibration tests will have to be carried out in order to decide on the number of the utilized model parameters and also in order to tune the algorithms' parameters for the particular problem. Finally, it should further be noted that, in this study, a low number of model parameters were considered, while in other problems with higher number of parameters the performance of the presented algorithms may differ.

7. Conclusions

Within this study, three different optimization algorithms were employed to solve the parameter estimation problem for a macroscopic traffic flow model. In particular, the deterministic Nelder-Mead algorithm, the stochastic genetic algorithm and the stochastic cross-entropy method were utilized in order to calibrate the METANET model for a particular freeway stretch using real traffic data. The optimization results showed that all three algorithms were able to converge to robust model parameter sets, albeit achieving different performances considering the convergence speed and the required computation time.

Acknowledgements

This research was co-financed by the European Union (European Social Fund - ESF) and by national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funded Project: ARCHIMEDES III, Investing in Society's Knowledge through the European Social Fund. The authors would like to thank ATTIKES DIADROMES S.A. for providing the utilised traffic data from Attiki Odos freeway in Athens, Greece and also Prof. Mike Maher and Dr. Dong Ngoduy for providing the source code for the kernel-based cross-entropy method.

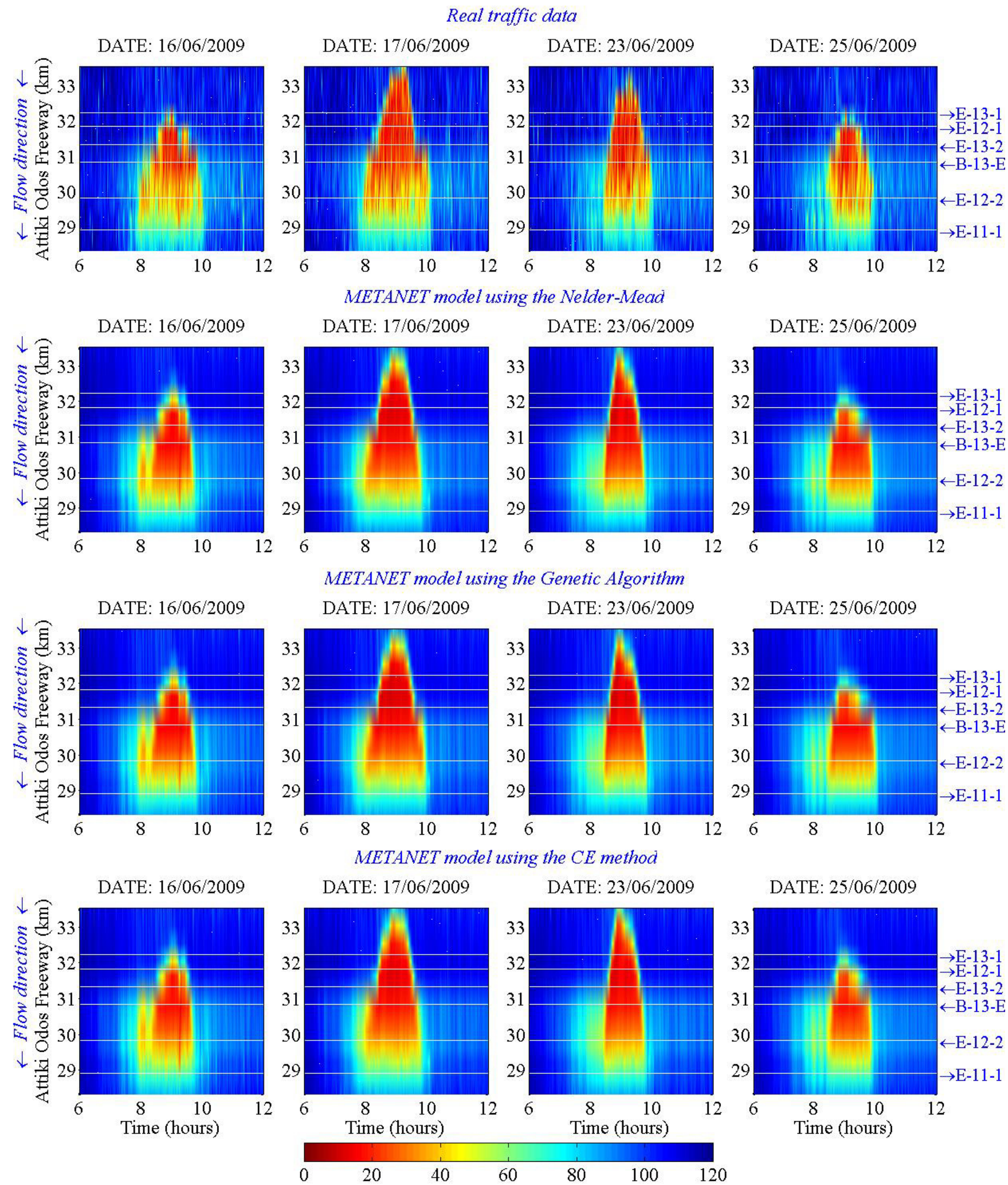

Fig. 10 Space-time diagrams of real measured speed and the models' estimation of speed for 16/06/2009, 17/06/2009, 23/06/2009 and 25/06/2009.

References

Box, M. J., (1965). A new method of constrained optimization and a comparison with other methods. *Computer Journal*, vol. 8, pp. 42–52.

Cremer, M., Papageorgiou, M. (1981). Parameter identification for a traffic flow model. *Automatica*, vol. 17 (6), pp. 837–843.

De Boer, P-T., Kroese, D.P, Mannor, S., Rubinstein, R.Y., (2005). A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, vol. 134, pp. 19–67.

Goldberg, D. (1989). Genetic Algorithms in Search. *Optimization and Machine Learning*. Reading, MA: Addison-Wesley Professional.

Grewal, M. S., Payne, H. J. (1976). Identification of parameters in a freeway traffic model. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6 (3), pp. 176–185.

Helbing, D. (1996). Derivation and empirical validation of a refined traffic flow model. *Physica A: Statistical Mechanics and its Applications*, vol. 233 (1), pp. 253–282.

Holland, J. (1992). Adaptation in Natural and Artificial Systems., MIT Press, Cambridge, MA.

Hoogendoorn, S. P., Bovy, P.H (2001). State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I. Journal of Systems and Control Engineering*, vol. 215 (4), pp. 283–303.

Kotsialos, A., Papageorgiou M. (2000). The importance of traffic flow modelling for motorway traffic control. *Networks and Spatial Economics*, vol. 1,pp. 179–203.

Kotsialos, A., Papageorgiou, M., Diakaki, C., Pavlis, Y., Middelham, F. (2002). Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. *IEEE Trans. on Intelligent Transportation Systems*, vol. 3, pp. 282–292.

Lagarias, J. C., Reeds, J. A., Wright, M. H., Wright, P. E. (1998). Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, vol. 9 (1), pp. 112–147.

Messmer, A., Papageorgiou, M. (1990). METANET: A macroscopic simulation program for motorway networks. *Traffic Engineering & Control*, vol. 31 (8–9), pp.466–470.

Michalopoulos, P. G., Yi, P., Lyrintzis, A. S. (1993). Continuum modelling of traffic dynamics for congested freeways. *Transportation Research Part B*, vol. 27 (4), pp. 315–332.

Monamy, T., Haj-Salem, H., Lebacque, J.-P. (2012). A Macroscopic node model related to capacity drop. *Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation*, vol. 54, pp. 1388–1396.

Nelder, J. A., Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, vol. 7, pp. 308–313.

Ngoduy, D., Hoogendoorn, S. P., van Zuylen, H. J. (2004). Comparison of numerical schemes for macroscopic traffic flow models. *Transportation Research Record*, vol. 1876, pp. 52–61.

Ngoduy, D., Maher, M. J. (2012). Calibration of second order traffic models using continuous cross entropy method. *Transportation Research Part C*, vol. 24, pp. 102–121.

Papageorgiou, M., Blosseville, J. M., Hadj-Salem, H. (1989). Macroscopic modelling of traffic flow on the Boulevard Périphérique in Paris. *Transportation Research Part B*, vol. 23 (1), pp. 29-47.

Papageorgiou, M., Blosseville, J.-M., Hadj-Salem, H. (1990). Modelling and real-time control of traffic flow on the southern part of Boulevard Périphérique in Paris - Part I: Modelling. *Transportation Research Part A*, vol. 24 (5), pp. 345–359.

Papageorgiou, M., Papamichail, I., Messmer, A., Wang, Y. (2010). Traffic simulation with METANET. In: *Fundamentals of Traffic Simulation*, J. Barceló, Editor, Springer, New York, pp. 399–430.

Poole, A. J., Kotsialos, A. (2012). METANET Model validation using a genetic algorithm. *Control in Transportation Systems,* vol. 13 (1), pp. 7–12.

Rubinstein, R.Y., Kroese, D.P. (2004). The cross-entropy method: a unified approach to combinatorial optimization. *Monte-Carlo Simulation, and Machine Learning*. Springer-Verlag, USA.

Sanwal, K. K., Petty, K., Walrand, J., and Y. Fawaz. An extended macroscopic model for traffic flow. *Transportation Research Part B*, Vol. 30, No. 1, 1996, pp. 1–9.

Spiliopoulou A., Kontorinaki M., Papageorgiou M., Kopelias, P. (2014). Macroscopic traffic flow model validation at congested freeway off-ramp areas. *Transportation Research Part C*, vol. 41, pp. 18–29.