Transportation Research Part C 66 (2016) 3–26

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Network traffic control based on a mesoscopic dynamic flow model

Massimo Di Gangi [a], Giulio E. Cantarella [b,*], Roberta Di Pace [b], Silvio Memoli [b]

[a] Department of Civil Engineering, University of Messina, Italy
[b] Department of Civil Engineering, University of Salerno, Italy

ARTICLE INFO

Article history:
Received 15 February 2015
Received in revised form 24 September 2015
Accepted 1 October 2015
Available online 2 November 2015

Keywords:
Network traffic control
Meta-heuristics
Multi-criteria optimisation
Mesoscopic flow modelling

ABSTRACT

The paper focuses on Network Traffic Control based on aggregate traffic flow variables, aiming at signal settings which are consistent with within-day traffic flow dynamics. The proposed optimisation strategy is based on two successive steps: the first step refers to each single junction optimisation (green timings), the second to network coordination (offsets). Both of the optimisation problems are solved through meta-heuristic algorithms: the optimisation of green timings is carried out through a multi-criteria Genetic Algorithm whereas offset optimisation is achieved with the mono-criterion Hill Climbing algorithm. To guarantee proper queuing and spillback simulation, an advanced mesoscopic traffic flow model is embedded within the network optimisation method. The adopted mesoscopic traffic flow model also includes link horizontal queue modelling. The results attained through the proposed optimisation framework are compared with those obtained through benchmark tools.

1. Background and motivation

This paper proposes a Network flow based Traffic Control method carried out through two successive steps: the first step refers to each single junction optimisation (green timings), the second to network coordination (offsets).

Furthermore, the adopted traffic flow modelling is a mesoscopic packet based approach, TRAFFMED (Traffic Analysis and Flow Forecasting Mesoscopic Dynamic) developed from an existing model for evacuation plan design (Di Gangi, 2011).

In the following the most used approaches for network traffic control are summarised with respect to the existing literature.

1.1. Strategies for network signal setting design

In general, strategies for signalised junctions may be classified as flow based (A) or vehicle arrival based (B).

In the case of flow based strategies (A), the input data are aggregate variables. Such strategies can be implemented as:

(A.1) fixed timing plans (pre-timed), as long as the settings are constant over periods of the day, depending on the flow evaluation based on historic values;

* Corresponding author.
E-mail addresses: mdigangi@unime.it (M. Di Gangi), g.cantarella@unisa.it (G.E. Cantarella), rdipace@unisa.it (R. Di Pace), smemoli@unisa.it (S. Memoli).

http://dx.doi.org/10.1016/j.trc.2015.10.002
0968-090X/

(A.2) timing plan selection, when the plan is periodically updated in real-time by choosing a plan from among a library of pre-timed signal plans depending on detected flows;
(A.3) timing plan computation, when the plan is periodically updated in real-time by computing a new plan depending on detected flows.

In the case of strategies based on disaggregate input data (B), the signal settings are obtained by detecting vehicles approaching the junction.

1.1.1. Flow based strategies

One of the most used flow based network control strategies (strategy A.1) is the TRANSYT method, first developed by Robertson in 1969 and then enhanced in successive releases (Vincent et al., 1980; Chard and Lines, 1987; Binning et al., 2010). Such a method is based on traffic modelling through cyclic flow profiles of arrivals at each junction; results of the traffic model are used to compute a performance index, P.I. (i.e. the sum of a weighted linear combination of delays and the number of stops per unit time) assuming signal timings and node offsets as optimisation variables and stage composition and sequence as input. In recent versions apart from the Platoon Dispersion Model (PDM), the Cell Transmission Model (CTM; see Daganzo, 1994) can be used for traffic flow modelling; meta-heuristic algorithms (Hill Climbing or Simulated Annealing) are applied for signal setting optimisation.

Notwithstanding their wide-spread use, fixed timing strategies may perform poorly when actual flows are greatly different from those used for optimisation due to within-day fluctuations, as well as day-to-day variations. In order to overcome such a limitation, some authors have developed network control strategies based on real-time observed flows, despite them generating high operational costs (in terms of sensors, communications, local controllers, etc.). Methods which fall within such a group are SCOOT (Split Cycle Offset Optimisation Technique; Hunt et al., 1981; Bretherton et al., 1998; Stevanovic et al., 2009) and SCATS (Luk, 1984; Stevanovic et al., 2008). These require traffic data to be updated on-line in order to get input flow for the optimiser (such as TRANSYT, Binning et al., 2010) and arrange green timings, offsets and cycle time duration. These methods match the off-line approach with on-line data (strategies A.2 and A.3).

Whichever optimisation method is employed, Network flow based Traffic Control strategies require within-day-dynamic traffic flow modelling. Several approaches can be adopted for within-day dynamics in a transportation network. They can be classified with respect to: (i) users' variables, which can be aggregate variables, such as path link flow and link density or disaggregate variables such as trajectory and the position of a single user; (ii) the level of service variables, such as travel time or space mean speed which may refer to a flow of users or to each single user. Three main groups of within-day dynamic models are identified:

- Macroscopic models where users' behaviour variables are aggregate (link density or entry flows can be obtained from the vehicle position on the link) as well as level of service variables (space mean speed, link performance functions are derived from fundamental diagram);
- Mesoscopic models where users' behaviour variables are disaggregate (packets of users or single users are considered; link density or entry flows can be obtained from packets/users position on the link) and the level of service variables are aggregate (such as space mean speed; link performance functions are derived from the fundamental diagram);
- Microscopic models where users' behaviour variables are disaggregate (single users are considered; link density or entry flows can be obtained from the users' position on the link) as well as the level of service variables (time speed and link performance functions are derived from the drivers' behaviour models such as car-following models).

Almost all Network flow based Traffic Control strategies proposed in literature use macroscopic models (see Cantarella et al., 2015 for an in depth analysis of the state of the art), whereas very rarely, microscopic models are used in embedded optimisation methods. Very few authors have investigated the effect of traffic management in a mesoscopic traffic simulation (DYNAMIT; Ben-Akiva et al., 1996; DYNASMART, Jayakrishnan et al., 1994; CONTRAM; Leonard et al., 1989). This paper proposes a Network flow based Traffic Control strategy (NTC TRAFFMED) using a mesoscopic discrete packet modelling. Indeed, major emphasis is not placed on the mesoscopic model but on the integration within the optimisation procedure. To the authors' knowledge such an approach has been never pursued.

The State of the art in terms of mesoscopic traffic flow models is discussed in more details in the following Section 1.2.

1.1.2. Vehicle activated strategies

For the sake of completeness, a brief review on network traffic responsive control strategies (i.e. vehicle-actuated control), which require knowledge of vehicle arrivals (strategy B), has been hereunder developed.

DYPIC (Robertson and Bertherton, 1974) is a backward dynamic programming algorithm based on the rolling horizon procedure for the heuristic solution search. Three steps can be identified: first of all, a planning horizon is split into a 'head' period with detected traffic information and a 'tail' period with synthesised traffic information; secondly, an optimal policy is calculated for the entire horizon and implemented only for the 'head' period and finally, for the next discrete time interval, when new detected information is available, the process rolls forward and repeats itself. Gartner (1983) gives a detailed description of the rolling horizon approach in OPAC. Rather than aiming at dynamic programming, OPAC uses a technique named Optimal Sequential Constrained Search to plan for the entire horizon, penalising queues left after the horizon.

PRODYN (Henry et al., 1983), also by adopting the rolling horizon approach, optimises timings via a forward dynamic programming (FDP). The FDP records the optimal trajectory of control policy in the planning horizon and the process rolls forward as it is defined in the rolling horizon approach.

UTOPIA (Urban Traffic Optimization by Integrated Automation, Mauro and Di Taranto, 1989; Mauro, 2002) is a hybrid control system that combines on-line dynamic optimisation and off-line optimisation. This is achieved by adopting a hierarchy structure with a wide-area level and a local level. An area controller continuously generates and provides a reference plan for local controllers. The local controllers then adapt the reference plan and coordinate signals dynamically in adjacent junctions. The rolling horizon approach is then used again by the local controller to optimise single junction design variables. To automate the process of updating reference plans that are generated by TRANSYT, an AUT (Automatic Updating of TRANSYT) module is developed. These traffic data are continuously updated by evaluating the mean of the flows collected by some of the detectors in the network. The data are processed to predict traffic flow profiles for different parts of the day to be used when calculating new reference plans. AUT generates the data to be used in the TRANSYT computation.

1.2. Mesoscopic traffic flow modelling

1.2.1. Brief state of the art

Mesoscopic models may be classified in terms of flow representation. Two different approaches may be identified in the literature: the packet, say the group of users/the single user representation and the single vehicle representation. A packet of vehicles acts as one entity and its speed on each road (link) is derived from a speed–density function defined for that link. Each packet is dealt with as a single entity which experiences the same traffic conditions. Several authors have proposed methods based on packets of vehicles to reduce the computed effort with respect to available computer resources. This feature is significant in order to classify papers proposed in the past since computing resources which are currently available make it possible to consider each packet made up of one vehicle only and, therefore, this distinction is no longer available.

A further classification of packet based models can be made in terms of a discrete packet (Leonard et al., 1989; Cascetta et al., 1991; Dell'Orco, 2006; Celikoglu and Dell'Orco, 2007) and a continuous packet (de Romph, 1994; Di Gangi, 1992). In the first case, a discrete distribution of vehicles in the packet is considered. All users are grouped in a single point (for instance the head of the packet) and, therefore, are located contemporarily at the same position over the link. In the continuous packet based approach, the vehicles are considered uniformly distributed (in time or space) in the packet, which is thus identified by two main points i.e. the head and the tail of the packet. Due to their inherent difficulties related to numerical problems of internal consistency when instantaneous density variations between adjacent simulation steps occur, only a few authors in literature (Di Gangi, 1992, 1996) have investigated continuous packet models and the most relevant contributions rely on discrete packet methods. It is worth noting that Continuous packets are relevant only when packets are made up of more than one vehicle.

As alternatives to packets representation, DYNAMIT (Ben-Akiva et al., 1996) moves individual vehicles along segments according to speed–density (Underwood, 1961; Drake et al., 1967; Drew, 1968) relationships and a queuing model whereas DYNASMART (Jayakrishnan et al., 1994) represents individual vehicles by point particle. The latter also uses macroscopic speed–density relationships but adopts a more detailed representation of signalised junctions to model delays at these facilities. A very similar product to DYNASMART is the release 1 of INTEGRATION (Van Aerde and Yagar, 1988). In the case of signalised junctions some authors have investigated the use of stochastic queue servers at the nodes. During each simulation interval, vehicle dynamics are defined in accordance with the macroscopic speed–density relationship and a queue-server at the nodes accounts for delays caused by traffic signals, downstream capacity limits and interaction with additional traffic. Examples of such models are the traffic flow models in DYNAMEQ (Mahut et al., 2002) and MEZZO (Burghout, 2004).

Mesoscopic traffic flow models may be further classified in terms of queuing representation in link based models, which are in turn grouped further depending on the performance function, and node based models (Celikoglu and Dell'Orco, 2007; Celikoglu et al., 2009), usually referring to the models which consider the flow splitting rates. In particular, link based models are also divided in (i) travel time models (CONTRAM, Taylor, 2003; Bliemer, 2006; Astarita, 1996; Adamo et al., 1999) and (ii) exit function models (such as the M–N model proposed by Merchant and Nemhauser, 1978a,b; the CTM model developed by Daganzo 1994, 1995; the point queue- PQ-model proposed by Smith, 1983 and the model proposed by Kuwahara and Akamatsu, 1997).

Travel time models may also be classified depending on the link representation: by considering the whole link so that the travel time is the sum of running travel time and queuing time (Jayakrishnan et al., 1994) or by considering the link divided in a running part (free flow moving) and a queuing part. In order to capture the effect of dynamic horizontal queuing and then of the spillback simulation in the proposed model, a link based approach (falling in the category of travel time models) considering the link split in a running and a queuing part is adopted. In particular, unlike Ran and Boyce (1996), in which the representation of the running and the queuing part of the link is instantaneous (not time dependent), the adopted mesoscopic traffic flow model was based on a 'variable' queue lengths representation which is affected by the previous simulation (see He, 1997). Such a representation makes it possible to clearly identify the boundary of the two parts of the link and then to avoid unreliable delay computations. Unlike the model presented in He (1997), based on fixed outflow capacity, the proposed model, as in Bliemer (2006) considers a dynamic queue-running part representation by changing outflow capacities ('dynamic queue approach'). Finally, as in Ben-Akiva et al. (1996), Jayakrishnan et al. (1994), Celikoglu and Dell'Orco (2007), Bliemer (2006), in the proposed contribution a path choice model is implemented.

1.2.2. Adopted mesoscopic traffic flow modelling

The adopted mesoscopic traffic model is based on discrete packet representation and each packet is made up of a single vehicle only. It is a link based model, falling within the class of the travel time models, and makes it possible to explicitly represent: horizontal queues; proportional traffic leaving a node through an explicit path choice model; the dynamic generation of the path-flows incidence matrix.

Furthermore, in the considered traffic flow model a speed–density relationship is adopted to evaluate the speed of a packet on the running part only and it is not used as a cost function. In this way, double counting the delay experienced by the vehicles is avoided, firstly due to the traversal speed (which is computed as a function of the actual density) and then due to the queuing delay (which is obtained as a result of the simulation).

Moreover, as in Bliemer (2006), we consider a dynamic queue-running part representation by including link capacity restrictions which are due to the signal timings variations over the time steps of simulation.

A more detailed overall model description is shown in Section 3.

It is worth noting that this paper does not aim to propose a mesoscopic model formalisation but our main contribution relies on embedding the mesoscopic traffic flow simulation within a network control strategy.

1.2.3. Proposed NTC strategy

The goal of this paper is twofold:

- proposing a Network Traffic Control (NTC) procedure based on mesoscopic traffic flow modelling (TRAFFMED); the control procedure is based on two steps: in the first step, decision variables are the green timings and a multi-criteria optimisation is adopted; in the second step, decision variables are the offsets and a mono-criterion optimisation is applied; the proposed optimisation procedure is a further development of one described in Cantarella et al. (2015) which is based on macroscopic traffic flow modelling;
- testing a discrete packet approach for mesoscopic traffic flow modelling, TRAFFMED, aiming to consider the link queuing dynamic simulation (including spillback simulation).

The paper is organised as follows: in Section 2 the Network Traffic Control problem in terms of objective functions, decision variables, constraints and metaheuristic solution algorithms are shown; in Section 3 a detailed description of the proposed mesoscopic traffic flow model is provided; in Section 4 the results from the application of the proposed methodology on a toy network are shown and compared with a benchmark tool; in Section 5 some considerations are expressed regarding the effects and the developments of control strategy on an urban level.

2. Network traffic control

The Network Traffic Control procedure presented in this section aims at optimising green timings and node offsets (see below for a formal definition) as decision variables.

A two-step optimisation is proposed, thus two separate optimisation levels are identified:

- in the first level the single junction optimisation is carried out (green timings are decision variables);
- in the second level the network coordination is performed (offsets are decision variables).

Single junction optimisation is carried out following the multi-criteria approach and the considered objective functions are the capacity factor (defined in Section 2.1) and the total delay (further/other objective functions may be considered such as fuel consumption; air pollution etc.); the stage composition and sequence as described by the stage matrix are assumed known; the set of optimal solutions, identifying the points of the Pareto front, are found through Genetic Algorithms. Network optimisation is carried out following mono-criterion optimisation. In this case, once the green timings are known, the cycle length and the stage matrix and composition (derived from multi-criteria single junction optimisation), the node offsets are optimised considering the network total delay minimisation, and are computed by applying the traffic flow model described in Section 3.

The proposed control strategy may operate off-line or on-line, in both cases no traffic flow prediction is carried out and the decision variables are optimised every control interval considering the most recently observed flows (see Fig. 1).

In our application the off-line conditions are adopted and TRAFFMED acts as plant model and meanwhile it provides the flows for decision variables design.

Furthermore, in Fig. 2, the network traffic control procedure is shown in more details.

2.1. Variables and constraints

Assuming that the stage composition and sequence are explicitly considered, let:

c be the cycle length, assumed known or as a decision variable (common to all junctions);

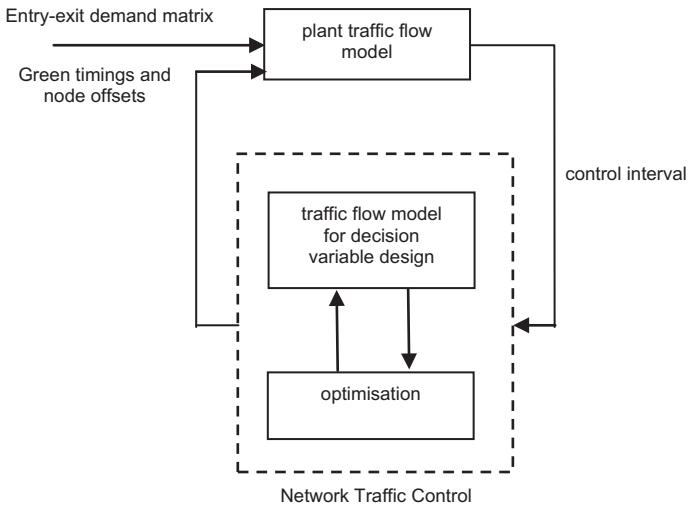

Fig. 1. Detailed description of the whole framework.

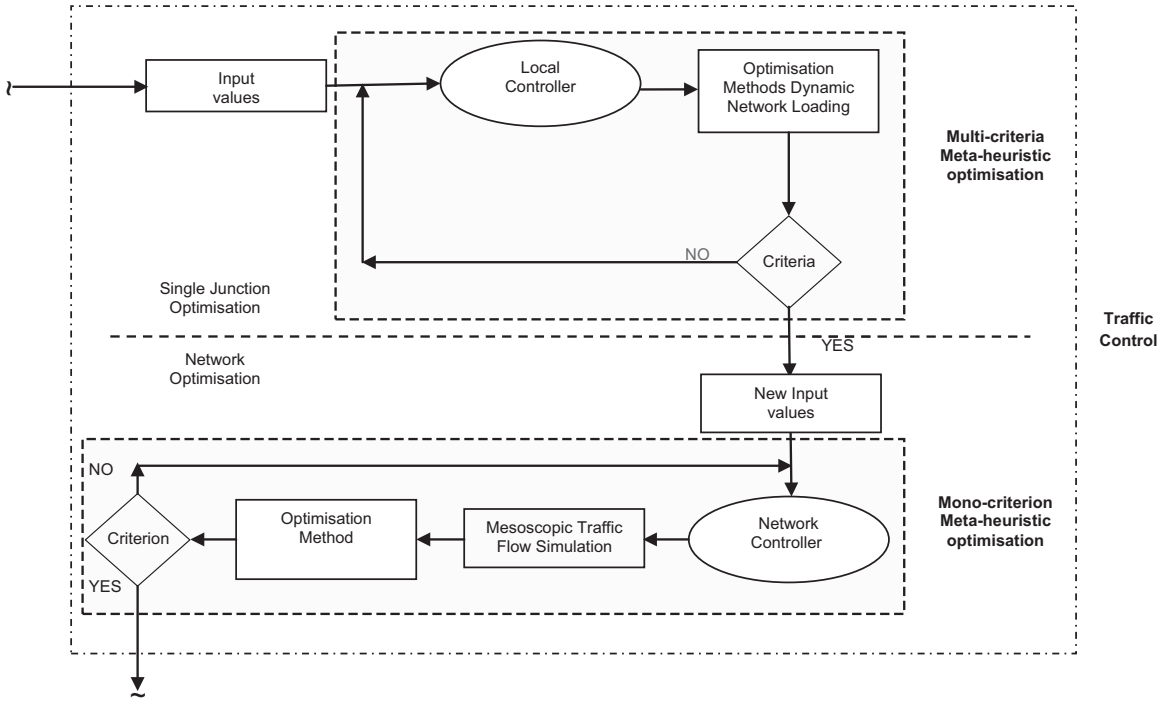

Fig. 2. Detailed description of the Network Traffic Control procedure.

for each single junction (not explicitly indicated).

t_j be the duration of stage j as a decision variable;
Δ be the approach-stage incidence matrix, or stage matrix for short, with entries $\delta_{kj} = 1$ if approach k receives green during stage j and =0 otherwise, assumed known;
t_{ar}, be the so-called all red period at the end of each stage to allow the safe clearance of the junction, assumed known (and constant for simplicity's sake);
l_k be the lost time for approach k, assumed known;
$g_k = \sum_j \delta_{kj} \cdot t_j - t_{ar} - l_k$ be the effective green timings for approach k, needed to compute the total delay as described in Eq. (2);

y_k be the arrival flow for approach k, assumed known;
s_k be the saturation flow for approach k, assumed known;

Some constraints were introduced in order to guarantee:

– stage durations being non-negative

$t_j \geq 0 \quad \forall j$

– effective green timings being non-negative

$g_k \geq 0 \quad \forall k$

[this constraint is usually guaranteed by the non-negative stage duration, but for a too short cycle length with respect to the values of all-red period length and lost times, say the cycle length is less than $\sum_j \mathrm{Max}_k\,(\delta_{kj} \cdot l_k + t_{ar})$ this condition might not be met];

– consistency among the stage durations and the cycle length

$$\sum_j t_j = c$$

– the minimum value of the effective green timing

$g_k \geq g_{\min} \quad \forall k$

A further constraint may be included in order to guarantee that the capacity factor is greater than 1 (or any other value assumed as a threshold)

$$\left(\frac{(s_k \cdot g_k)}{(c \cdot y_k)} - 1\right) \geq 0, \quad \forall k$$

After having checked that the capacity factor for each approach k is greater than 1, such a constraint may be added, otherwise a solution may not exist whatever the objective function is.
For each junction i in the network let

ϕ_i be the node offset, defined as the time shift between the start of the plan for the junction i and the start of the reference plan, say the plan of the junction number 1, $\phi_1 = 0$;
ϕ_{ij} be the link offset between each pair of adjacent junctions, $\phi_{ij} = -\phi_{ji}$ needed for computing network total delay.

Let the junction network be represented by an undirected graph with a node for each junction and an edge for each pair of adjacent junctions (the actual traffic directions are irrelevant). According to this representation

– if such a network is loop less, all the $m - 1$ link offsets are independent (as many as the independent node offsets) and may be used as decision variables; arterials are a special case of such kinds of networks;
– on the other hand, if the network contains k independent loops, the number of independent link offsets will be equal to $m - k$; in this case, it is better to use the $m - 1$ independent node offsets as optimisation variables.

Finally let's assume

$c \geq \phi_I \geq 0.$

2.2. Objective functions

The objective functions adopted for the multi-criteria single junction optimisation were:
The junction capacity factor computed as

$$\mathrm{CF}_k(g_k) = \mathrm{Max}_k \frac{(s_k \cdot g_k)}{(c \cdot y_k)} \tag{1}$$

where $\frac{(s_k \cdot g_k)}{(c \cdot y_k)}$ is the capacity factor for approach k.
The total delay, TD^S, computed through the two terms Webster's formula (Webster, 1958) as

$$\mathrm{TD}^S(g_k) = \sum_k y_k \cdot \left(\frac{0.9 \cdot c \cdot \left(1 - \frac{g_k}{c}\right)^2}{2 \cdot \left(1 - \frac{y_k}{s_k}\right)}\right) + \frac{0.9 \cdot y_k}{2 \cdot (s_k \cdot \frac{g_k}{c}) \cdot \left(\left(\frac{s_k}{y_k}\right) \cdot \left(\frac{g_k}{c}\right) - 1\right)} \tag{2}$$

It should be noted that the Webster formula does not take into account the offsets so it is only to be applied for single junction optimisation.

Further objective functions may be introduced such as air pollution, and fuel consumption.

The Network optimisation was carried out by minimising the total delay, TD^N, computed through the traffic flow model described in the Section 3.2 (see Eq. (6)).

2.3. Solution algorithms

The optimisation framework presented in this paper (as described in the previous sections) introduces a multi-objective procedure to obtain green timings at a single junction (for efficiently finding optimal values of competitive objective functions that were the Capacity Factor and the Total Delay) and a Network optimisation to obtain minimum Network Total Delay. In both cases meta-heuristics were applied. As a matter of fact, such algorithms make it possible to effectively address optimisation problems (see Ozan et al., 2015; Ceylan and Ceylan, 2012) when conflicting objectives are identified, leading to multi-criteria optimisation (as it occurs in the proposed single junction optimisation) or when the objective function may not be expressed in a closed form, thus, derivatives are not easily available (as occurs in the Network optimisation problem).

A brief description of the adopted meta-heuristics, Genetic Algorithm (GA) at a single junction and Hill Climbing (HC) at network level, is shown hereunder.

2.3.1. Multi-criteria Genetic Algorithm

For multi-criteria optimisation two objective functions are considered. In order to properly select the solution, the dominance criterion is applied and the Pareto front (which is the representation of the set composed by the vector of the objective functions) must be defined. In accordance with the literature, in this paper the GA is adopted as the solution algorithm which is a nature inspired algorithm and is based on the biological evolutionary process reproduced by an iterative procedure.

The solutions are identified by chromosomes each one composed by genes which represent, in our case, the stage durations; the chromosome corresponds to a vector of stages. Each solution is evaluated in terms of fitness described by a specific expression; on the basis of the fitness value, each chromosome could be selected to be parent in a reproductive cycle. In this cycle chromosomes might be modified by the application of two genetic operators, the crossover which generates a new solution by mixing genes of two chromosomes and the mutation which induces random variations of genes in a current solution (chromosome). Summing up, this meta-heuristic algorithm is based on an iterative procedure as summarised herein:

[1] The Population generation, in which the chromosomes are randomly generated; [2] the Fitness Evaluation (Baker, 1985), which is strictly related to the probability of being selected to be a parent, in fact, it is computed by normalising the fitness function with respect to the fitness functions of all chromosomes; [3] the Selection, which is based on the fitness function and depends on the ranking evaluation; moreover, in the case of equal fitness, the solution is selected on the basis of the crowding distance computed as in the NSGA-II (Deb, 2002; Deb et al., 2002), by considering the combination in the Euclidean distance of two criteria (such as the capacity factor and the total delay); the crossover [4] and mutation [5] operators, which might be applied for several times until the new population is obtained; the final step [6] is identified with the Stop criterion which is defined in terms of both maximum number of generations (and iterations) and non-improvement of the solution. A detailed description of the adopted algorithm herein briefly summarised may be founded in Cantarella et al. (2015).

2.3.2. Hill Climbing algorithm

Hill Climbing is a neighbourhood-based meta-heuristic algorithm, without memory, deterministic in its basic version. The name originates from its ability to generate a succession of solutions by exploring the objective function surface which, if plotted, could be thought of as a series of hills and valleys in a multiple-dimensional space.

The algorithm was applied to optimise the node offsets for each controller and it is launched at the end of the multi-criteria GA procedure to get green timings at a single junction.

The algorithm first calculates the initial Network Delay (DN), and then begins an iterative process; Let n be the number of junctions and k be the number of independent loops in the network, on each iteration, the algorithm moves to junction $i = 1,\ldots,n-k$ and tests out a series of $j = 1,\ldots,h$ adjustments (as percentages of the cycle time C) on independent node offsets. The adjustments are stored in a vector $H = \{5,15,40,15,40,15,5,1,1\}$[1]. Hill climbing methods, such as the one herein described, do not guarantee finding the global minimum. To reduce the possibility of finding a poor local optimum, we use both large and small adjustments for the successive optimisation of each junction. The 15% adjustments find an approximate local minimum of the objective function whilst the 40% adjustments avoid becoming 'trapped' in that minimum. To ensure that H can contain a 1 s adjustment, the value of elements $H_{j=8}$ and $H_{j=9}$ are interpreted as 1 s and not as a percentage. At junction i and at iteration $j = 1$, the adjustment $H_j \cdot C$ is applied to the initial node offset ϕ_i so that the new adjusted node offset is $\phi_i(\text{adj}) = \phi_I + H_j \cdot C$. Such a new node offset is then tested to obtain the trial delay $D(\text{trial})$ on the link approaching the junction i. This is compared to the delay value $D(i)$ obtained considering the initial node offset ϕ_i. If Delay has been reduced ($D(\text{trial}) < D$

[1] Other values should be assigned to the adjustment vector; the assumed values are those generally adopted.

(i)), the new adjusted node offset ϕ_i(adj) is saved as the best known and the search direction is kept for the next iteration; otherwise it is discarded (maintaining the previous node offset) and the search direction is inverted so that H_{j+1} becomes $-(H_{j+1})$. After the first junction has been 'visited' by all the adjustments $H_{j=1,\ldots,h}$, the same process is applied for the second junction and so on until $i = n - k$. Residual dependent node offsets are then computed via an algebraic sum. Fig. 3 presents an illustration of how the HC algorithm works.

3. Traffic flow modelling

The adopted traffic flow model (as an enhancement of the model presented in Di Gangi, 2011) is discussed in this section. In particular, a detailed description of network representation, travel times and cost variables, flow representation and traffic dynamics is shown.

In the following, intervals will be denoted by integer indices, whilst time instants and lengths will be denoted by real values.

The considered mesoscopic model is based on a discrete packet approach, where the simulation is carried out over discrete *simulation time intervals*, t, further divided into *traffic dynamics sub-intervals*, k with length δ.

3.1. Packet generation

The generic packet P, is characterised by a departure time η within the *departure interval*, h (which is the simulation interval during which the departure occurs) and an origin/destination pair rs; path choice is evaluated at the beginning of each departure interval, say at the origin and may be updated at the beginning of each simulation interval t, if en route rerouting

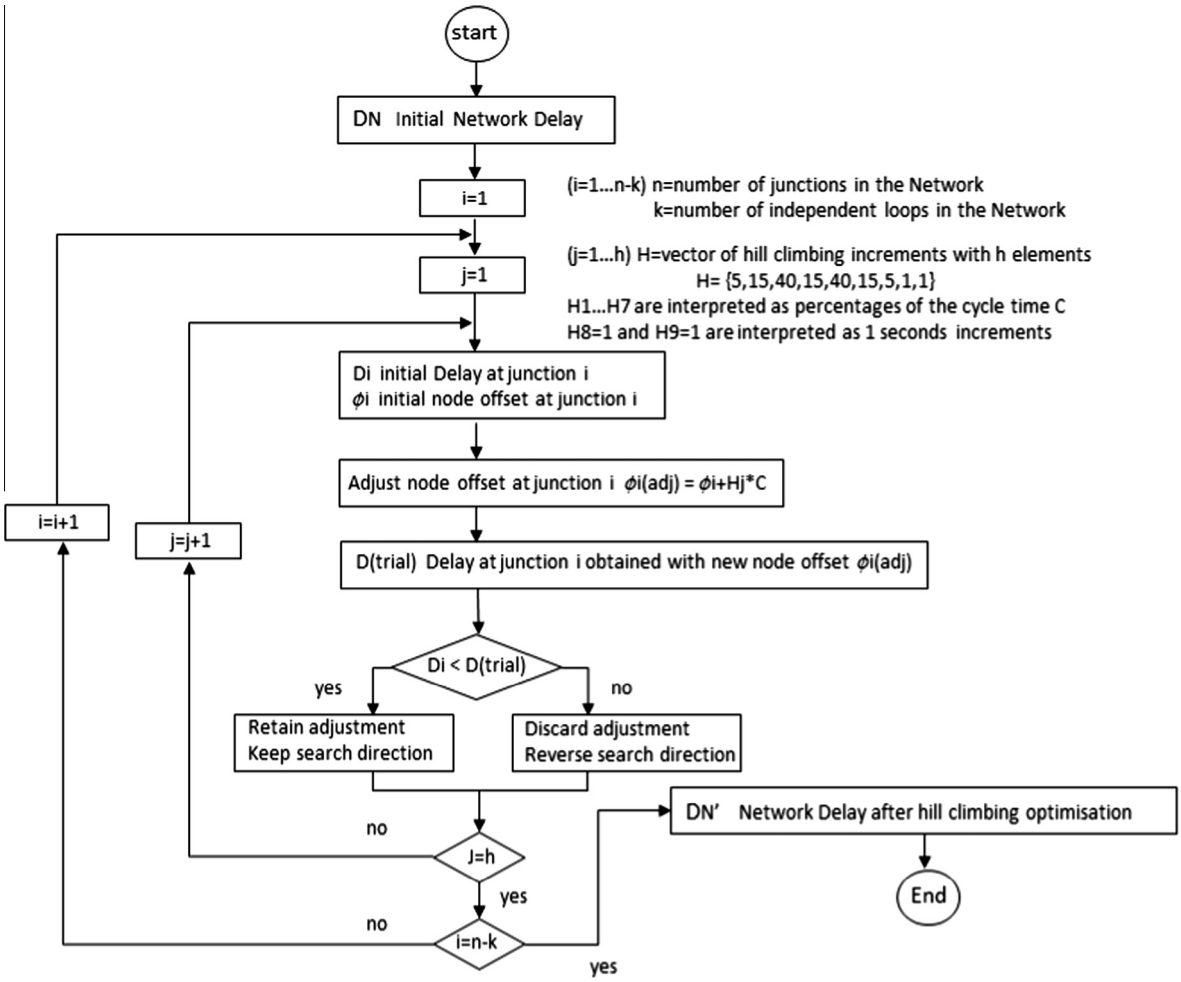

Fig. 3. General description of the Hill Climbing algorithm.

occurs. For instance, the length of the simulation interval t and the length of the departure interval h, may be equal to 5 min and each simulation interval is further divided into sub-intervals with each one equalling 5 s.

The simulation is initialised considering a warm up period in order to guarantee the traffic congestion effects and avoid unrealistic free flow conditions.

A packet P moves in the network (see below Traffic dynamics for more details) and it is subject to queuing phenomena. The network is represented by a graph $G(N,A)$, with N the set of nodes and A the set of links.

Let

L_a be the length of link a,
x_a^S be the abscissa of a section S in link a that divides the link into two parts named respectively running part $[0,x_a^S)$ and queuing part $[x_a^S,L_a]$,
$\varsigma_a = L_a - x_a^S$ be the part of the link occupied by the queue.

The position of section S is obtained by the dynamic network loading (a detailed description is provided in Fig. 4) and is updated at each time interval t considering the number of packets in the queuing part of the link at the previous interval $t-1$; the speed on the running part of the link at interval t depends on the outflow conditions of the previous interval $t-1$. This approach is slightly inconsistent since the level of service in the interval is considerably affected by flows and queues in the previous interval and not in the current. This assumption makes it possible to greatly simplify the computation whilst the inconsistence can be limited by reducing the length for the sub-interval, τ.

Outflow conditions on each link are considered homogeneous and constant for the entire duration of a sub-interval, δ. They are estimated at the beginning of each sub-interval and maintained for its entire duration, thus avoiding the occurrence of the internal fixed point problem. Summing up, the movement of a generic packet depends on its position on the link (running or queuing) and it can change its outflow conditions by moving forward from the running part to the queuing part of the link.

At each simulation time, the length of the queue on the link (and, consequently, the abscissa of section S) is updated and it is possible to take into account the eventual occurrence of queue spillback.

3.2. Traffic dynamics

In this section both the movement rules of packets within each part of the link and the queue spillback simulation are described. These rules also make it possible to compute the travel time on the running part and on the queuing part.

Let $\Gamma_{rs\eta} = (N^{rs} \subset N, A^{rs} \subset A)$ be a sub-graph composed by the set of links that belong to the feasible paths connecting the O/D pair rs computed at time η, and let x be the abscissa on link a belonging to sub-graph $\Gamma_{rs\eta}$ representing the position, at time τ of interval t, of packet P left at the time η of the departure interval $h \leqslant t$ (if $h = t$ then $\eta < \tau$).

All variables of the traffic dynamics are declared below. In particular, let

ρ_a^k be the density of the running part of the link a at beginning of sub-interval k given by the total amount of vehicles in each packet on link a during sub-interval k;

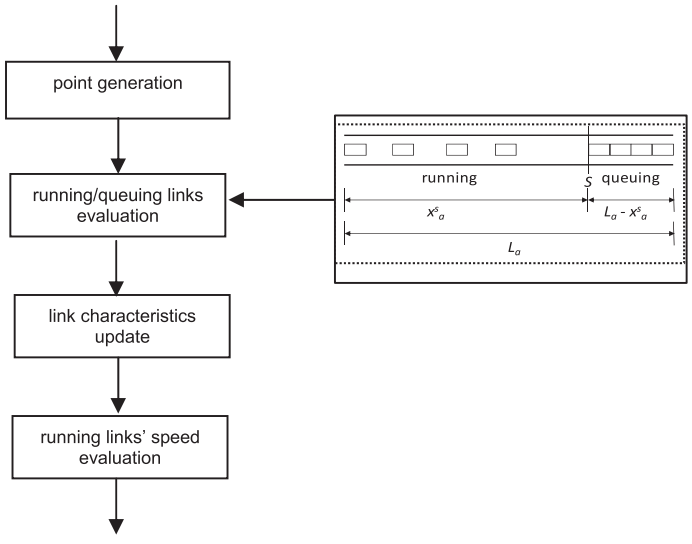

Fig. 4. Detailed description of the dynamic network loading.

v_a^k be the speed on the running part of the link a, updated at the beginning of each sub-interval k, as a function of density ρ_a^k;
$\rho^a{}_{\text{crit}}$ be the critical density on the link a, i.e. the density corresponding to the capacity in the fundamental diagram;
Q_a be the capacity at the final section of the link a;
δ be the length of the generic sub-interval k and let $\tau \in [0, \delta]$.

With reference to the link model described above, the following cases may occur:

if $x < x_a^S$, the packet P is located on the running part of the link, it moves forward at a speed v_a^k; then within the interval t, the packet P may reach a maximum abscissa x_a^S; consequently
if $(x_a^S - x) < (\delta - \tau) \cdot v_a^k$, the packet P enters the queuing part of the link a, before the end of the interval, at time τ' computed as in follows:

$$\tau' = \tau + \left(\frac{(x_a^S - x)}{v_a^k} \right) \tag{3}$$

otherwise, at the end of the interval it remains located on the running part;
if $x \geqslant x_a^S$, the packet P moves on the queuing part of the link a; the length of the queuing part travelled by the packet P by the end of the interval is given by:

$$\Delta = \frac{((\delta - \tau) \cdot Q_a)}{\rho^a_{\text{crit}}}; \tag{4}$$

if $x + \Delta > L_a$, the packet P leaves the link a during the interval t at time:

$$\tau'' = \tau + \frac{\left((L_a - x) \cdot \rho^a_{\text{crit}}\right)}{Q_a} \tag{5}$$

otherwise it remains on the queuing part of the link a.

When packet P reaches the end of link a, it is necessary to identify the next link of its followed path. Let A^+ be the choice set of available links (made up by the links of $\Gamma_{rs\eta}$ whose initial node correspond to the final node of link a), a choice weight π_a^Γ is associated to each one of these links so that the link-choice problem can be expressed as:

$$a^+ \in \Gamma_{rs\eta} | \pi^\Gamma_{a+} > \pi^\Gamma_{ai} \quad \forall a_i \in A^+, a_i \neq a^+.$$

Since $\Gamma_{rs\eta}$ is a DAG (Directed Acyclic Graph), all links in the set A^+, belong to at least one path connecting the rs pair, starting from r at time η, the arrival at destination of the packet is ensured.

As described above, for each O/D pair, rs, and departure time η a DAG sub-graph $\Gamma_{rs\eta} = (N^{rs} \subset N, A^{rs} \subset A)$ of the network is associated to the packet P. Such a sub-graph is composed of the set of links that belong to the feasible paths connecting the O/D pair rs computed at time η. A choice weight π_a^Γ is associated to each link $a' \in A^{rs}$; the sub-graph $\Gamma_{rs\eta}$ is generated by implicit paths enumeration (i.e. Dial's STOCH; see Dial, 1971) and π_a^Γ is directly obtained (considering the total travel time on the link), thus the link-path incidence matrix is dynamically generated at the beginning of each interval t as well as the path flow patterns.

Before entering the next link a^+ it must be verified that (i) link a^+ can accept the incoming packet and (ii) the residual capacity of link a allows the packet P to move to the next link a^+.

When the packet P reaches the end of the link a, before moving forward to the next link a^+, it has to be verified that the length of the running part of the link a^+ is not null, that it is $x_{a+}^S > 0$.

If $x_{a+}^S > 0$, the packet P may enter the link a^+ otherwise it means that the entire length of link a^+ is occupied by a queue and the packet P remains on link a until the queue length on link a^+ is smaller than the length of the link a^+, that is for the time until the condition $L_{a+} - x_{a+}^S < L_{a+}$ is satisfied.

Once verified that the length of the running part of a link $a^+ \in A^+$ is not null, that is $x_{a+}^S > 0$, it must be verified that the residual capacity of the link a allows the packet P to move to the next link a^+.

Let

$n_e(P)$ be the number of elements of packet P, that is, $n_e(P) = 1$,
$U_a(\tau)$ be the number of packets which left the link until time τ,
Q_a be the capacity at the final section of the link a,
δ/T be the ratio between the length of interval k and the time unit T considered for the capacity.

If $\frac{[U_a(\tau) + n_e(P)]}{\tau} \leqslant Q_a \cdot \frac{\delta}{T}$, the packet P may leave the link a, otherwise the link a is saturated and the packet P remains on the link as long as a residual capacity, which allows the packet to leave the link, becomes available.

With reference to the introduced notation, it is possible to express the total delay (the total queuing time):

$$TD^N(\phi_{ij}) = \sum_a \frac{(\varsigma_a(\phi_{ij}) \cdot \rho^a_{crit})}{Q_a} \quad (6)$$

3.3. Implementation remarks

In order to implement the above described method the following inputs must be defined:

- the study interval length;
- the simulation interval length;
- the traffic dynamics interval length;
- the number of vehicles in each packet P assumed one vehicle at most;
- the demand flows over the simulation interval;
- the path choice model applied at beginning of each choice model interval;
- the departure time choice model, applied after the path flows have been defined; in a simple approach departures are uniformly distributed over time with constant headway given by the ratio between the length of simulation and the flow;
- the speed–density function to compute the congested speed, as explained in more detail in Appendix A.

4. Application

In this section the case study and the comparison between the results obtained through the proposed method and the benchmark tool are discussed. In particular, the results of two applications are shown, simple network (A), also considered for validation, and a more complex network (B). The proposed method was implemented in a code developed in Python (according to the JetBrains PyCharm Community Edition 3.0.2 framework was adopted) and run on a server machine which has an Intel(R) Xeon(R) CPU E5-1603, clocked at 2.8 GHz and with 4 GB of RAM.

4.1. The case study (A)

In this section an empirical validation test of NTC-TRAFFMED is discussed. The proposed approach is shown in Fig. 5; in particular, two steps were considered:

- the first step aims at validating the network traffic control procedure;
- the second step aims at validating the traffic flow model.

In the first step, the results carried out by NTC (Network Traffic Control procedure corresponding to the same adopted in the application shown in Section 4.2) considering the Cell Transmission Model (CTM) were compared with those carried out by a benchmark tool TRANSYT©TRL still considering the same traffic flow model; in this step the network traffic control procedure (published in Cantarella et al., 2015) is compared with the optimisation procedure in TRANSYT©TRL. Moreover, the same set of input conditions (path choice at each choice model interval, free flow speed, saturation flow, the demand flows over the simulation interval) is adopted in order to guarantee the consistency of the comparison; the adopted values of input parameters are described in the following section.

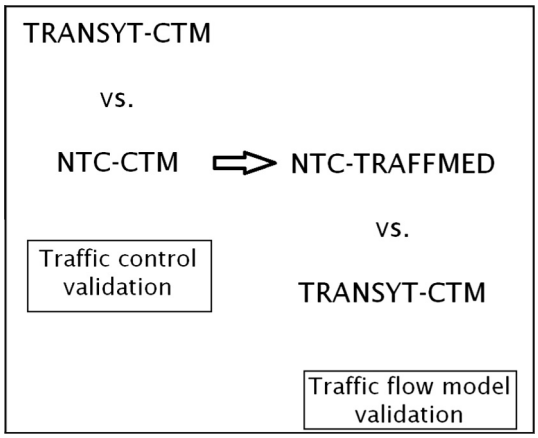

Fig. 5. Overview of a two-step validation test.

In the second step, the NTC-TRAFFMED is compared with the NTC-CTM used at the previous step and then the procedures are characterised by the same network traffic control procedure but by two different traffic flow models.

The considered network was composed of a bi-directional triangular network connecting 3 zones, as shown in Fig. 6. Traffic signal 1, which was located at node 1, was set to be the master signal relative to which the offset of traffic signal 2 (at node 2) and 3 (at node 3) was referenced. The demand entry-exit flows were summarised in Table 1.

A cross comparison among the three optimisation strategies is shown in the following tables (see Tables 2 and 3) in terms of performance indicators (TD, total delay and DOS, degree of saturation) and in terms of node offsets.

The obtained results showed that NTC CTM mimics TRANSYT CTM well whilst it slightly differs from NTC-TRAFFMED; in particular, in terms of total delays, similar values are obtained from NTC-CTM and TRANSYT-CTM and a slightly different value is obtained from NTC-TRAFFMED. De facto, the total delays estimation, as well known, depends on the adopted traffic flow model; in terms of DOS, as in previous cases, similar values are obtained from NTC-CTM and TRANSYT-CTM and an outperforming value is obtained from NTC-TRAFFMED. Finally, as expected, different results are obtained in terms of node offsets.

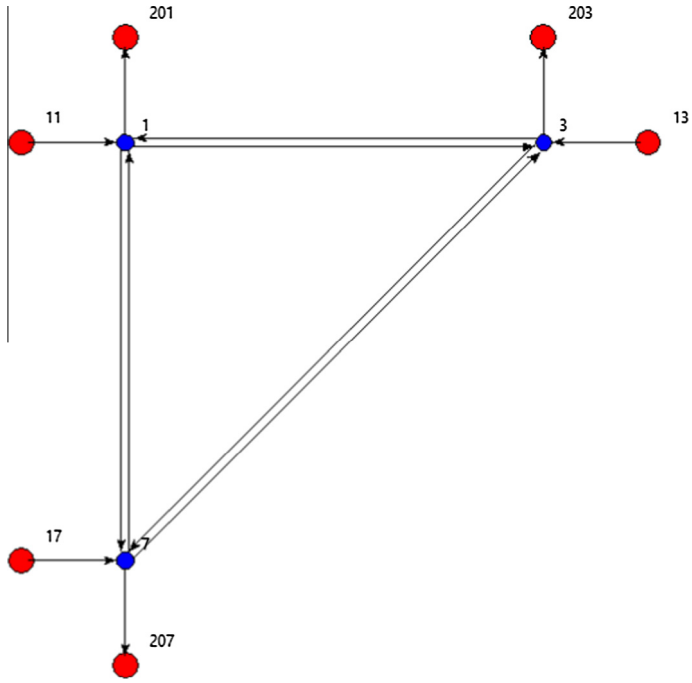

Fig. 6. Network used for calibration/validation.

Table 1
Entry-exit matrix.

Entry [veih/h]	Exit [PCU/h]		
	201	203	207
11	#	73	45
13	28	#	50
17	65	45	#

Table 2
Performance indicators of NTC-TRAFFMED, NTC-CTM and TRANSYT-CTM.

Performance indicator	NTC-TRAFFMED	NTC-CTM	TRANSYT-CTM
TD [PCU-h/h]	9.58	10.83	10.61
DOS [%]	37.00	44.00	46.00

In conclusion, the above results show that the effectiveness of the proposed optimisation method in NTC is comparable with the one in TRANSYT, see comparison between NTC-CTM and TRANSYT-CTM; moreover, that Network Traffic Control with mesoscopic traffic modelling may produce better results, as shown by the DOS indicator that is independent of the method (unlike the total delay).

A further validation of the optimisation procedure was carried out by considering TRAFFMED for decision variable design and CTM as the plant model ($NTC_{TRAFFMED}$–CTM). Results displayed in Table 4 shows the effectiveness of the proposed mesoscopic model for decision variable design within the optimisation procedure, in fact, NTC-TRAFFMED outperforms $NTC_{TRAFFMED}$–CTM.

4.2. The case study (B)

In order to further test the proposed optimisation strategy NTC-TRAFFMED, a grid network (see Fig. 7) was considered, made up of 4 origins, 4 destinations (totalling 17 nodes including connectors) o–d pairs, 9 nodes and 12 bidirectional links (totalling 32 links including connectors), each one with one lane for each direction; the saturation flow of each lane is

Table 3
Node offsets of three traffic flow models.

Node offset	NTC-TRAFFMED	NTC-CTM	TRANSYT-CTM
$\Phi 2$–1 [s]	79	57	52
$\Phi 3$–1 [s]	60	28	28

Table 4
Performance indicators of NTC-TRAFFMED and $NTC_{TRAFFMED}$–CTM.

Performance indicator	NTC-TRAFFMED	$NTC_{TRAFFMED}$–CTM
TD [PCU-h/h]	9.58	9.81
DOS [%]	37.00	40.00

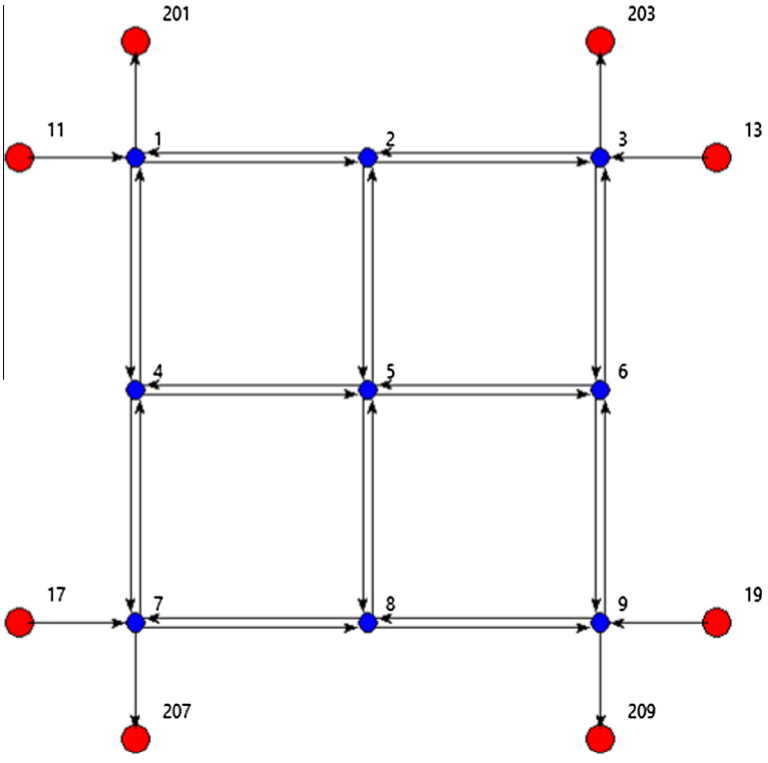

Fig. 7. Test network with 9 signalised junctions.

Table 5
Entry-exit matrix.

Entry [veih/h]	Exit [PCU/h]			
	201	203	207	209
11	#	480	384	336
13	432	#	288	384
17	480	624	#	423
19	336	576	432	#

assumed equal to 1800 PCU/h. In terms of the links' length, links connecting node 5 with other nodes (2–5, 5–6; 5–8) are equal to 400 m, other links on the network are equal to 800 m. The speed could be obtained by any speed density function (fundamental diagram – stable regime) such as the link performance BPR function, and the conical function (see Spiess, 1990). In this paper the BPR-like function $v(f) = v_0/(1 + a(f/Q)^b)$ was adopted by assuming suitable values of parameters $a = 2.0$ and $b = 3.5$ (see more details in Appendix A).

The o–d pair demand flows are shown in the following Table 5. The path choice behaviour was modelled considering three paths for each o–d pair (a selective approach to path choice specification) varying over the simulation intervals depending on network conditions, and a logit choice model. The perceived utility of each path from origin o to any destination d is assumed distributed with variance σ_o^2 equal to 20% of the average of the minimum path costs across all destinations. Thus, the dispersion parameter for origin o is given by $\theta_o = (\sqrt{6}/\pi) \cdot \sigma_o$.

In our application the total simulation time was 60 min and was divided into 12 departure intervals, h, each of them equal to 5 min and was coincident with the simulation interval, t; during t, the path choice remains constant and packets departures are uniformly distributed; each simulation interval t was further divided into sub-intervals with length δ equal to 5 s.[2]

The network was preloaded at 60% of the links' capacities (6480 PCU in all) in order to properly compute the congested speed from the beginning of the simulation.

An example of the path flow patterns (at time interval 3), is shown in Table 6.

In the same table the (adjusted) flows and the discharge headway according to each simulation interval are also shown. It is worth nothing that in most cases only one path has a positive flow but this path may change over the simulation intervals.

The traffic signals were updated every 15 min, which we refer to as the control interval, by implementing the optimisation procedure, on the basis of the flows measured at the previous time interval.

In particular, the traffic flow data are obtained by using the same mesoscopic model as previously described and are used as inputs of the Network Traffic Control procedure in order to get consistent values of the decision variables (green timings at local level and offsets at network level). Therefore, let the length of the simulation interval be equal to 5 min, the Dynamic network loading is run for n successive times until $n \cdot 5$ equals the length of the control interval (15 min).

4.3. Analyses

In this sub-section the effect of the proposed procedure was evaluated by considering some indicators such as the Mean Maximum Queue (MMQ) which is the mean over the simulation intervals of the position of the back of the queue at its peak during the cycle, measured in PCU back from the stop line, the Total Delay (i.e. TD) and the Capacity Factor (i.e. CF). As mentioned before, the optimisation was computed every 15 min, until the end of the simulation time, thus, results were compared by considering three successive intervals; the first time interval was not considered for any comparison due to the well-known effect of the warm up simulation (see results in italics in following tables).

The results were evaluated firstly at a disaggregated level over time (at each time interval) and an aggregated level for the entire network and secondly at a disaggregated level over time (as mentioned previously, at each time interval) and a disaggregated level for individual regions of the network.

With respect to the network results shown in Table 7, it should be observed that higher values of MMQ, TDs and CFs are obtained at time interval 3, however, at time interval 4 the network may be considered as under-saturated.

A further analysis was carried out by considering the results which were disaggregated for every region within the network; in particular, a clustering of links was identified by distinguishing internal, border and external regions as shown in Fig. 8 (i.e. links are clustered in external, internal and on the border between the internal and the external region). Their corresponding Queue Lengths (measured in meters), TDs and CFs, computed respectively as the average, the maximum and the minimum over links composing each region, are shown in the following Table 8.

As in the previous case, we exclude the results obtained for the first time interval.

Table 8 shows that congestion is equally distributed over three regions confirming procedure suitability in queuing phenomena management.

[2] The computer time needed for 1 h of simulation depends on the length of the sub-interval: if the length of sub-interval is equal to 5 s, the simulation takes 12 s; if the length of sub-interval is equal to 10 s, the simulation takes 16 s; if the length of sub-interval is equal to 15 s, the simulation takes 22 s.

Table 6
Path flow patterns at time interval 3.

O	D	Path	Path nodes							Flow rates	Flow [veh/300 s]	Headway [s/veh]
11	209	1	{11	1	4	5	6	9	209}	0.73	21	14.3
11	209	2	{11	1	4	5	8	9	209}	0.27	7	42.9
11	209	3	{11	1	4	5	8	5	6}	0.00	0	0.0
11	203	1	{11	1	4	5	6	3	203}	1.00	40	7.5
11	203	2	{11	1	2	3	203	–	–}	0.00	0	0.0
11	203	3	{11	1	4	5	8	5	6}	0.00	0	0.0
11	207	1	{11	1	4	7	207	–	–}	0.98	32	9.4
11	207	2	{11	1	4	5	8	7	207}	0.02	0	0.0
11	207	3	{11	1	4	5	4	7	207}	0.00	0	0.0
17	201	1	{17	7	4	1	201	–	–}	0.99	39	7.7
17	201	2	{17	7	4	5	2	1	201}	0.01	0	0.0
17	201	3	{17	7	8	5	2	1	201}	0.00	0	0.0
17	203	1	{17	7	4	5	6	3	203}	0.99	51	5.9
17	203	2	{17	7	8	5	6	3	203}	0.01	0	0.0
17	203	3	{17	7	4	5	8	5	6}	0.00	0	0.0
17	209	1	{17	7	8	9	209	–	–}	0.97	34	8.8
17	209	2	{17	7	4	5	6	9	209}	0.03	0	0.0
17	209	3	{17	7	4	5	8	9	209}	0.01	0	0.0
19	201	1	{19	9	8	5	2	1	201}	0.99	27	11.1
19	201	2	{19	9	8	5	4	1	201}	0.01	0	0.0
19	201	3	{19	9	6	3	2	1	201}	0.00	0	0.0
19	203	1	{19	9	6	3	203	–	–}	1.00	47	6.4
19	203	2	{19	9	8	5	6	3	203}	0.00	0	0.0
19	203	3	{19	9	6	3	6	3	203}	0.00	0	0.0
19	207	1	{19	9	8	7	207	–	–}	1.00	35	8.6
19	207	2	{19	9	8	5	8	7	207}	0.00	0	0.0
19	207	3	{19	9	8	5	4	7	207}	0.00	0	0.0
13	201	1	{13	3	2	1	201	–	–}	1.00	35	8.6
13	201	2	{13	3	6	3	2	1	201}	0.00	0	0.0
13	201	3	{13	3	2	5	2	1	201}	0.00	0	0.0
13	209	1	{13	3	6	9	209	–	–}	1.00	31	9.7
13	209	2	{13	3	6	3	6	9	209}	0.00	0	0.0
13	209	3	{13	3	2	5	6	9	209}	0.00	0	0.0
13	207	1	{13	3	2	5	8	7	207}	0.65	15	20.0
13	207	2	{13	3	6	5	8	7	207}	0.29	6	50.0
13	207	3	{13	3	6	9	8	7	207}	0.06	1	00.0

Table 7
Results over the time intervals achieved by NTC-TRAFFMED.

Interval	NTC-TRAFFMED		
	MMQ [PCU]	TD^N [PCU-h/h]	CF
1	*116.58*	*60.44*	*1.25*
2	109.50	58.82	<1[a]
3	164.61	106.27	<1[a]
4	106.33	36.26	1.58

[a] Oversaturation.

4.4. Comparisons: NTC-TRAFFMED vs. TRANSYT-CTM

In this sub-section the effect of the procedure was evaluated by considering the comparisons of the results (a more detailed description of considered indicators will be given below) collected from our simulation environment with ones carried out by using TRANSYT©TRL. Furthermore, the considered flow rates in TRANSYT©TRL were consistent with the link-path incidence matrix generated in NTC-TRAFFMED.

As previously described, in the optimisation procedure the green timings and the (node) offsets were optimised in two separate steps, and the stage matrix at each single junction was given.

The obtained results were evaluated by considering, as in previous sections, the indicators such as the Mean Maximum Queue (MMQ), the Total Delay (i.e. TD) and the Capacity Factor (i.e. CF). The traffic flow model adopted in TRANSYT©TRL in the case of optimisation and simulation is the Cell Transmission Model (CTM, Daganzo, 1994). The input parameters for each link that can be set in the TRANSYT-CTM are: (i) the cruise speeds (assumed 30 kph), whence the cell length, the number of cells and the maximum occupancy is obtained; the maximum flow (or the cell saturation flow, evaluated by taking account of link characteristics), whence the maximum number of vehicles that can flow into cell i from time t to $t+1$ is obtained. NTC-CTM was not applied for brevity's sake.

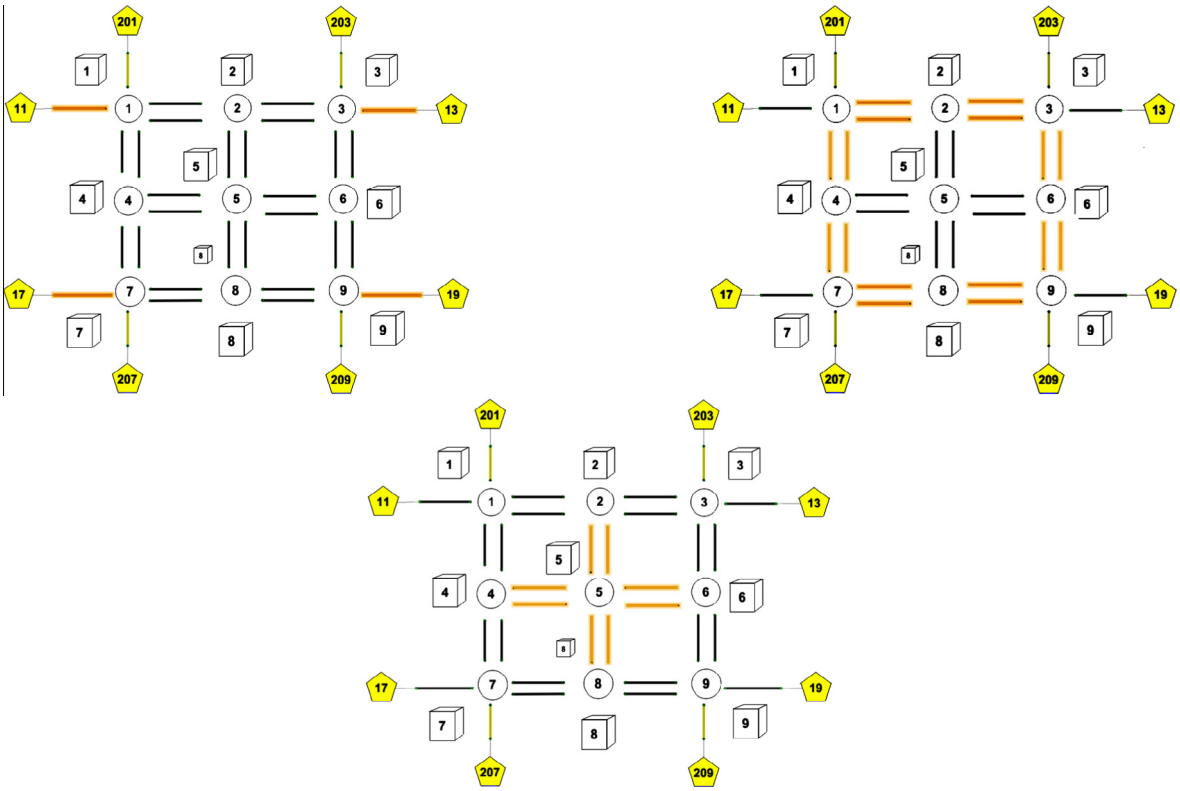

Fig. 8. Links composing the external, border, internal region.

Table 8
Results over the time intervals of simulation for the external, internal and border regions, achieved by NTC-TRAFFMED.

Interval	Queue length [m]			TD^N [PCU-h/h]			CF		
	EXT	BOR	INT	EXT	BOR	INT	EXT	BOR	INT
NTC-TRAFFMED									
1	*78.82*	*17.74*	*12.53*	*18.10*	*3.32*	*1.72*	*1.96*	*1.25*	*2.13*
2	65.87	21.11	6.97	25.73	4.31	0.87	0.52	1.12	4.55
3	56.04	27.93	39.57	16.36	28.7	32.74	0.83	0.67	0.72
4	34.94	19.95	22.39	2.85	5.62	4.25	1.59	1.69	1.75

The comparisons were made firstly on results which were disaggregated over time (at each time interval) and aggregated for the entire network and secondly on results which were disaggregated over time (as mentioned previously, at each time interval) and disaggregated for individual regions of the network.

In Table 9 the network results are shown. The first column refers to the time intervals of the simulation, as described above. The MMQ, the TDs and the CFs reported in the remaining columns represent the aggregate network values.

As aforementioned, we will not compare results carried out at the first time interval of simulation. However, further considerations can be made on numerical results of successive time intervals.

In particular, it can be observed that:

i. at the second interval of simulation, NTC-TRAFFMED outperforms TRANSYT-CTM with respect to the TD (i.e. 58.82 PCU-h/h against 99.52 PCU-h/h);
ii. at the third interval of simulation the results reached by both approaches may be considered comparable (i.e. 106.27 PCU-h/h in NTC-TRAFFMED and 97.05 PCU-h/h in TRANSYT-CTM);
iii. even though at the second and the third interval of simulation, the oversaturation conditions are reached in both of the cases, at the last time interval NTC-TRAFFMED outperforms TRANSYT-CTM not only with respect to the TD (i.e. $TD_{NTC\text{-}TRAFFMED} = 36.26$ [PCU-h/h] and $TD_{TRANSYT\text{-}CTM} = 319.2$ [PCU-h/h]) but also with respect to the CF (i.e. $CF_{NTC\text{-}TRAFFMED} = 1.58$ and $CF_{TRANSYT\text{-}CTM} < 1$).

Table 9
NTC-TRAFFMED vs. TRANSYT-CTM.

Interval	NTC-TRAFFMED			TRANSYT-CTM		
	MMQ [PCU]	TD^N [PCU-h/h]	CF	MMQ [PCU]	TD^N [PCU-h/h]	CF
1	*116.58*	*60.44*	*1.25*	*85.04*	*26.12*	*1.35*
2	109.50	58.82	<1[a]	126.86	99.52	<1[a]
3	164.61	106.27	<1[a]	118.33	97.05	<1[a]
4	106.33	36.26	1.58	344.76	319.2	<1[a]

[a] Oversaturation.

Table 10
Results over the time intervals of simulation for the external, internal and border regions, achieved by NTC-TRAFFMED and TRANSYT-CTM.

Interval	Queue length [m]			TD^N [PCU-h/h]			CF		
	EXT	BOR	INT	EXT	BOR	INT	EXT	BOR	INT
NTC-TRAFFMED									
1	*78.82*	*17.74*	*12.53*	*18.10*	*3.32*	*1.72*	*1.96*	*1.25*	*2.13*
2	65.87	21.11	6.97	25.73	4.31	0.87	0.52	1.12	4.55
3	56.04	27.93	39.57	16.36	28.7	32.74	0.83	0.67	0.72
4	34.94	19.95	22.39	2.85	5.62	4.25	1.59	1.69	1.75
TRANSYT-CTM									
1	*32.43*	*19.54*	*8.49*	*2.24*	*1.96*	*1.06*	*2.04*	*1.37*	*2.94*
2	153.95	6.63	4.92	31.78	1.75	0.72	0.42	2.13	7.14
3	153.59	4.14	3.67	31.78	1.27	0.68	0.52	1.12	4.55
4	245.99	5.57	4.23	115.74	0.73	1.05	0.42	2.14	2.94

With reference to the results of the 4th interval for NTC-TRAFFMED and TRANSYT-CTM in Table 10, they can be observed, both in terms of MMQ and of TD, and meaningful differences can be pointed out (The TD is reduced by a factor of 10). Such a result is due to the different traffic flow model adopted in the optimisation framework. As a matter of fact, the CTM in its basic version allows vehicles to be held at the upstream cells rather than advancing forward to the available-capacity downstream cells, which leads to the holding-back phenomenon (see Doan and Ukkusuri, 2012). Unlike CTM, our proposed TRAFFMED model allows for over the time intervals of the simulation to distribute the queue over the network links, leading to a satisfactory aggregate network Total Delay.

A further analysis was carried out by considering the results which were disaggregated for every region within the network as described in the previous Fig. 8. Their corresponding Queue Lengths (measured in meters), TDs and CFs, computed respectively as the average, the maximum and the minimum over links composing each region, are shown in the following Table 10. As in the previous case, we exclude any consideration of the results carried out at the first time interval.

Table 10 shows that it may be appropriate to compare the results obtained by NTC-TRAFFMED and TRANSYT-CTM for the three network regions at progressive time intervals of simulation. In fact, numerical results ensure that indicators of external regions in TRANSYT-CTM are usually dominated by ones collected in NTC-TRAFFMED and vice versa that indicators of internal and border regions in TRANSYT-CTM usually dominate those reached in NTC-TRAFFMED. However, if the averages of TDs and CFs are considered, a relative increase of the CF in TRANSYT-CTM, equal to 37% and a relative decrease of the TD, in NTC-TRAFFMED, equal to 50% is observed (presumably induced by the different optimisation strategy). In order to evaluate the effectiveness of the multi-criteria optimisation, a further analysis was carried out by comparing graphs (see Fig. 9) which represent the values of CFs optimised in NTC-TRAFFMED and TRANSYT-CTM. As in accordance with previous numerical considerations, it should be stated that TRANSYT-CTM induces higher values of CF over border and internal links unlike external links where higher values are obtained in NTC-TRAFFMED.

A final in depth investigation was carried out in terms of queues on the network. In particular, for each strategy, the Queues distribution at each time interval and for each cluster of links was obtained (see Fig. 10).

It appears noteworthy that the queues observed in TRANSYT-CTM in the internal and border links are relatively lower compared with ones experienced in the external links. In contrast, a homogeneous outcome (for each region and over the time intervals of simulation) of such an indicator may be revealed as regards the proposed strategy.

Indeed, the main advantage of NTC-TRAFFMED lies in a higher efficiency in managing the dynamics of overflow queues, unlike TRANSYT-CTM in which the higher values of queues are concentrated over all the time intervals of simulation on external areas.

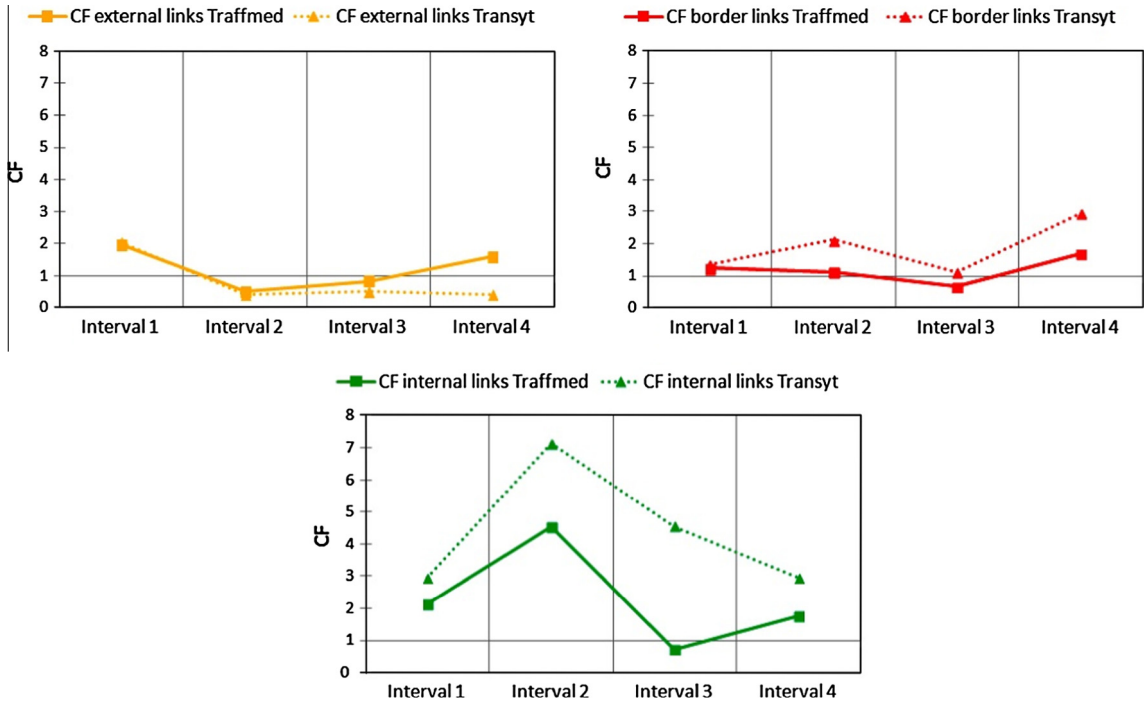

Fig. 9. CFs over the time intervals of simulation for the external, internal and border regions, achieved by NTC-TRAFFMED and TRANSYT-CTM.

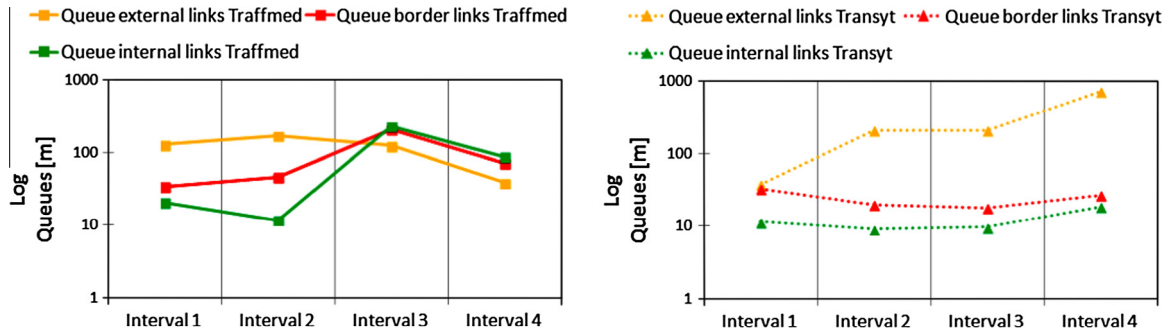

Fig. 10. Queues distribution over the time intervals of simulation for the external, internal and border regions, achieved by NTC-TRAFFMED (on the left) and TRANSYT-CTM (on the right).

5. Conclusions

The paper aims at developing a flow based Network Traffic Control method in which the optimisation is carried out through a two-step approach where the first step refers to each single junction optimisation (green timings), the second to network coordination (offsets).

The control strategy is affected by the optimisation procedure itself, as well as by the adopted traffic flow model. Due to this consideration, another focus of the paper lies in traffic flow modelling. In this research, an advanced mesoscopic approach with horizontal queuing is proposed (TRAFFMED – Traffic Analysis and Flow Forecasting Mesoscopic Dynamic). TRAFFMED is based on the representation of each link in a running and a queuing part whose length may change over time due the changes of the queue length.

Metaheuristic solution algorithms are applied since they seem to be more suitable to solve the optimisation problems for the single junction, where a multi-criteria optimisation is carried out (Cantarella et al., 2014), and for the network, where the objective function (i.e. total delay) is affected by traffic flow simulation and not expressed in a closed form.

Two cases of study are considered: a simple three nodes network and a more complex grid network composed by 9 nodes; the first one was adopted for calibrating and validating the NTC-TRAFFMED, the second for the further analysis of the model performances. For this purpose, in the second application case, the obtained results (NTC-TRAFFMED) are

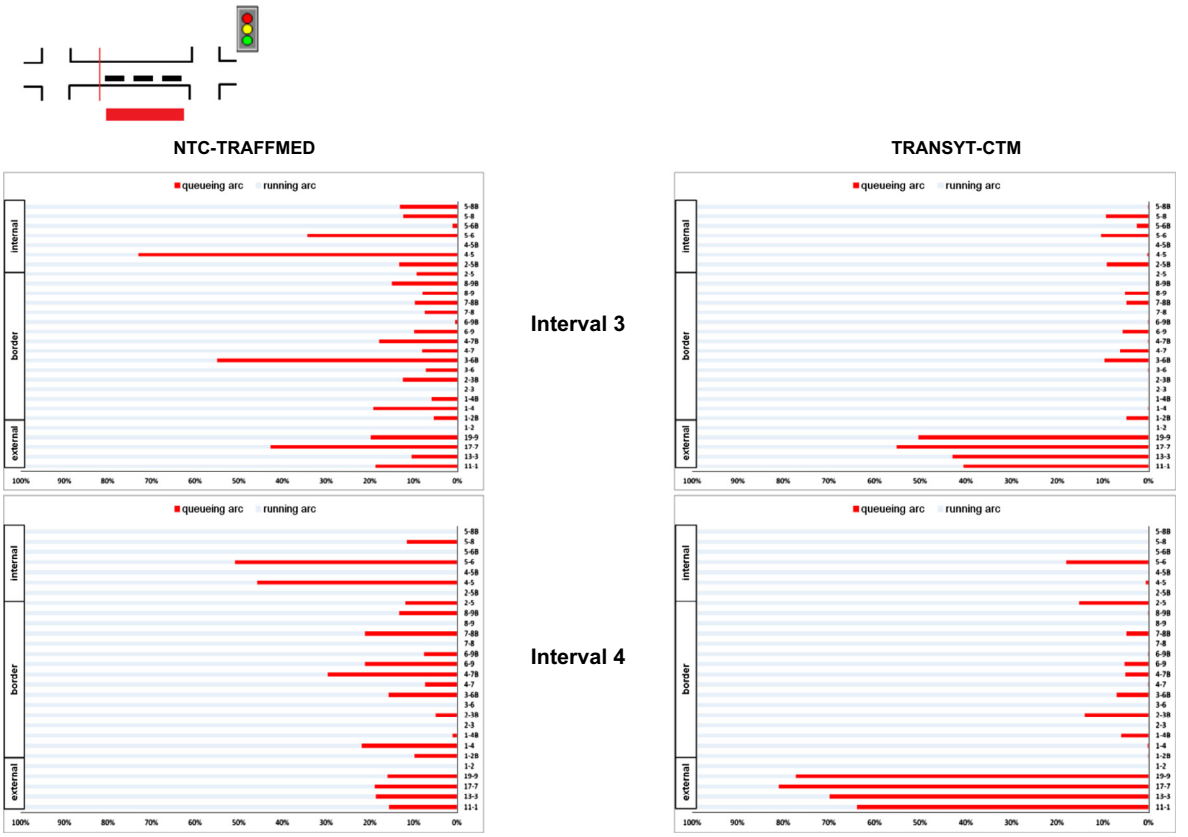

Fig. 11. Queues dynamic evolution over the time intervals (3 and 4) of simulation.

compared with those carried out from a benchmark tool (TRANSYT-CTM), with respect to simulated flow data. The most significant insight is observed in the case of links clustered in three network regions (external, border, internal); unlike the benchmark tool, in which queuing phenomena are significantly concentrated only on the external area, in the proposed approach queuing phenomena may be considered better spread over the network.

Percentage-based horizontal stacked bar plots (the red[3] bars refer to the queuing part, the light blue bars refer to the running part) are shown in Fig. 11, for the simulation intervals 3 and 4. In order to compare the control strategies mentioned above (a schematic representation of the running and the queuing part of the signalised link is illustrated at the top of Fig. 11). The representation of the links against the queue percentage aims to highlight the relevant difference between NTC TRAFFMED, in which from interval 3 to interval 4 queue distributions significantly change, and TRANSYT, in which from interval 3 to interval 4 queue distributions slightly change.

In future works, several issues will be addressed as far as the traffic control method is concerned:

(i) a network traffic control procedure where green timing, green scheduling and offsets are simultaneously optimised (scheduled synchronisation);
(ii) a thorough investigation of the platoon dispersion representation in the proposed model;
(iii) the integration of the proposed approach for network traffic control with a flow forecasting model;
(iv) the extension of the proposed mesoscopic NTC TRAFFMED model to store and a forward strategy (Gazis and Potts, 1963).

As regards the traffic modelling the calibration and validation of the proposed mesoscopic model will be carried out considering actual data. Furthermore, the mesoscopic traffic flow modelling herein adopted will be compared with other ones eventually with different levels of variables aggregation.

It will be worth considering also the proposed NTC strategy within the Urban Network Design Problem, including lane allocation optimisation as well as user path choice behaviour modelling.

[3] For interpretation of color in Fig. 11, the reader is referred to the web version of this article.

Acknowledgments

Authors wish to thank three anonymous reviewers for their helpful comments. This research has been partially supported by the University of Salerno, Italy, EU under local grant no. ORSA132945 financial year 2013, Campania Region PON- Ricerca e Competitività 2000-2013 grant no. 04a2_G "SMART TUNNEL", and PhD program on transportation (Ph.D. School in Environmental Engineering).

An earlier version of this research is contained in the preprints of Dynamic Traffic Assignment Conference 2014 and 11th International Conference on Advances in Civil Engineering (ACE2014).

Authors wish to thank to Nikolas Geroliminis, met at the University of Naples 'Federico II' for his helpful comments.

For friendship policy, the authors order is organised in alphabetical order with respect to the affiliations, then in alphabetical order of the authors name within each affiliation.

Appendix A. BPR-like link performance functions

The BPR (Bureau of Public Roads) function was developed in the late 1950's by fitting data collected on uncongested freeways and appeared in 1965 in HCM (Highway Capacity Manual) as:

$$t(f) = \frac{L}{v_0} \cdot \left(1 + 0.15 \cdot \left(\frac{f}{Q}\right)^4\right)$$

where:

$t(f)$ is the mean travel time,
L is the link length,
v_0 is the free flow speed,
f is the flow,
Q is the capacity.

Afterwards, BPR-like link performance functions have been introduced for urban applications as:

$$t(f) = \frac{L}{v_0} \cdot \left(1 + a \cdot \left(\frac{f}{Q}\right)^b\right)$$

where

a is a congestion parameter, such that $t(Q) = (a+1) \cdot \frac{L}{v_0}$,
b is a shape parameter, and
Q also take into account the green time/cycle length ratio.

In terms of speed–flow relationship (fundamental diagram – stable regime) the above function can be written as:

$$v(f) = \frac{v_0}{\left(1 + a \cdot \left(\frac{f}{Q}\right)^b\right)}$$

The original value of parameter a was 0.15, thus when flow (volume) equals the capacity, the congested travel time is only 15% greater of the free flow travel time and the standard BPR function may greatly over-estimates the speed at maximum capacity in particular in urban areas. Furthermore when the volume is very low, the predicted travel time is approximately equal to free-flow travel time.

To correctly estimate the travel time on urban links different values of parameters a and b are used such as $a = 2.0$ and $b = \{3.0, 3.5\}$ which seem to be more suitable values.

Such BPR-like functions, along with the standard BPR curve, are plotted in Fig. 12 for L equals 1 km, v_0 equals 50 kph and Q equals 2000 veh/h. The travel time, as expected, increases faster for urban link curve than for uncongested freeway curve. Overall, with BPR-like functions the travel time increases more quickly for small flow-capacity ratio whilst, on uncongested freeway, the travel time stays relatively constant for low flow. From these considerations the original BPR function does not seem to be well suited for urban applications.

Coherent considerations can be made with respect to the speed–flow curves, as shown in Fig. 13 for $a = 2.0$ and $b = \{3.0, 3.5\}$ only.

Spiess (1990) suggests the use of conical functions given as:

$$t(f,\alpha) = \left[2 + \left[\alpha^2 \cdot \left(1 - \frac{f}{Q}\right)^2 + \left(\frac{2\alpha - 1}{2\alpha - 2}\right)^2\right]^{0.5} - \alpha \cdot \left(1 - \frac{f}{Q}\right) - \left(\frac{2\alpha - 1}{2\alpha - 2}\right)\right] \cdot \frac{1}{v_0} \quad \text{with } \alpha > 1$$

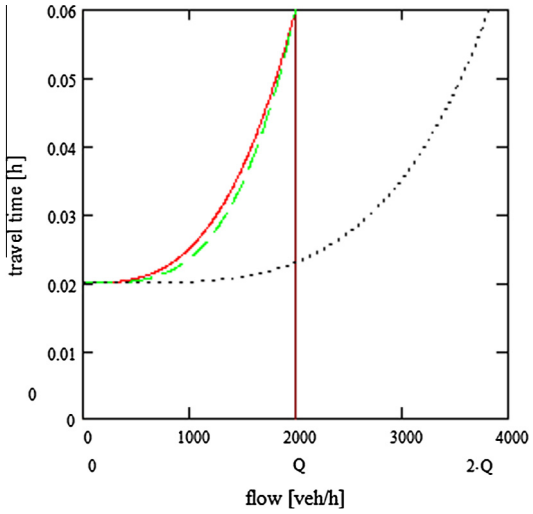

Fig. 12. Travel time over $L = 1$ km vs. flow ($v_0 = 50$ kph, $Q = 2000$ veh/h). Continuous red line refers to BPR like curve for $a = 2.0$ and $b = 3.5$. Dotted grey line refers to BPR standard curve. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

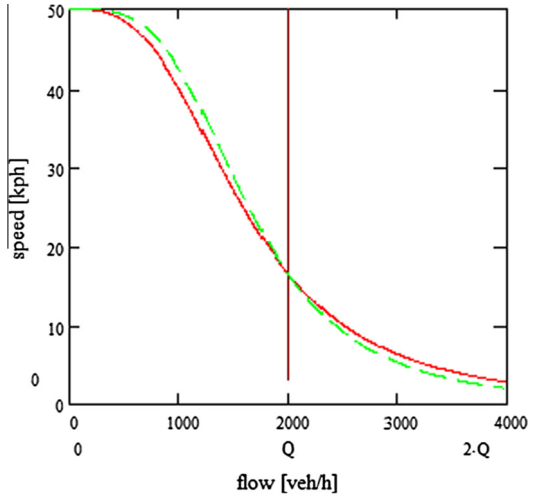

Fig. 13. Speed vs. flow (stable regime). Continuous red line refers to BPR like curve for $a = 2.0$ and $b = 3.0$. Dashed green line refers to BPR like curve for $a = 2.0$ and $b = 3.5$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

to estimate travel time on urban links. A minor flexibility in modelling the effect of congestion could be observed in conical functions, differently from BPR-like. As a matter of fact in the conical functions only a shape parameter α is considered. Whichever is the value of α, $t(Q) = 2L/v_0$. Conversely in the BPR-like function two parameters are identified: b which is a shape parameter and a which takes into account the congestion in such way that $t(Q) = (a + 1) \cdot (L/v_0)$. As shown in Fig. 14, BPR-like function with $a = 1$ (say $t(Q) = 2L/v_0$) and conical functions are not appreciably different.

Remembering that under steady-state condition the flow f is related to the density k and the mean speed v through the relation:

$$f = \rho \cdot v$$

the congested speed versus the density is given by solving the following equation:

$$\frac{v_0}{\left(1 + a \cdot \left(\frac{f}{Q}\right)^b\right)} - v = 0$$

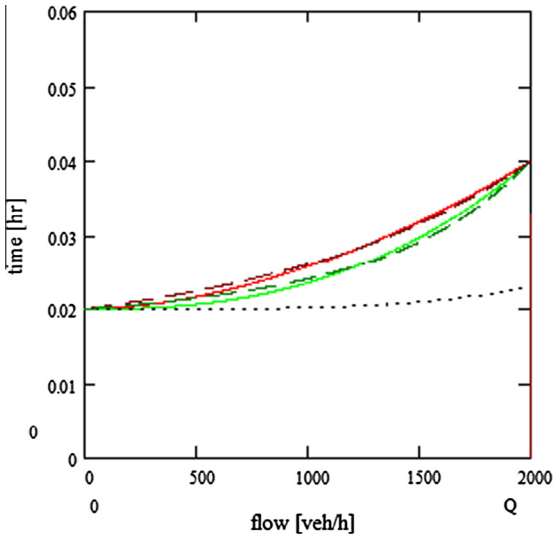

Fig. 14. Travel time over $L = 1$ km vs. flow ($v_0 = 50$ kph, $Q = 2000$ veh/h). Continuous red line refers to BPR like curve for $a = 1.0$ and $b = 1.8$; continuous green line refers to BPR like curve for $a = 1.0$ and $b = 2.5$. Dashed red line refers to conical curve for $\alpha = 2$; dashed green line refers to conical curve for $\alpha = 3$. Dotted grey line refers to BPR standard curve. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

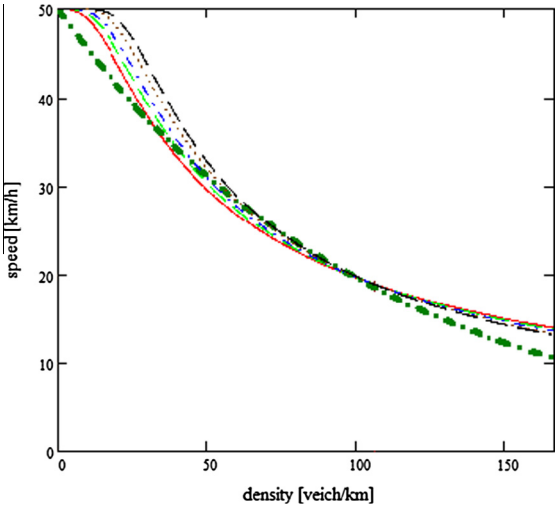

Fig. 15. Speed density function. Dashed black line refers to Underwood speed–density function; speed density function derived from the BPR like function are so identified: continuous red line for $a = 2.0$ and $b = 3.0$; continuous green line for $a = 2.0$ and $b = 3.5$; dotted blue line for $a = 2.0$ and $b = 4.0$; dashed brown line for $a = 2.0$ and $b = 5.0$; dashed black line for $a = 2.0$ and $b = 6.0$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

or

$$\left(\frac{a}{Q^b}\right) \cdot \rho^b \cdot v^{b+1} + v - v_0 = 0$$

It can be shown that such equation always has exactly one positive solution for any positive value of ρ, thus the above polynomial equation defines an implicit function $v(\rho)$ between speed and density (with parameters a, b, Q, v_0).

In Fig. 15 the function $v(\rho)$ for $a = 2$ and for different values of b is graphically represented. As it arises from the graph, varying the value of b does not affect significantly the shape of the function. The Underwood (Underwood, 1961) speed–density function given as:

$$v\,\text{Underwood}(\rho) = v_0 \cdot e^{\left(-\left(\frac{\rho}{\rho_{\text{crit}}}\right)\right)}$$

is also plotted to compare/validate the adopted curves.

References

Astarita, V., 1996. A continuous time link model for dynamic network loading based on travel time function. In: 13th International Symposium on Theory of Traffic Flow, pp. 79–103.

Adamo, V., Astarita, V., Florian, M., Mahut, M., Wu, J.H., 1999. Modelling the spill-back of congestion in link based dynamic network loading models: a simulation model with application. In: 14th International Symposium on Transportation and Traffic Theory.

Baker, J.E., 1985. Adaptive selection methods for genetic algorithms. In: Grefenstette, J.J. (Ed.), Proceedings of the 3rd International Conference on Genetic Algorithms and Applications. Lawrence Erlbaum, New Jersey, Hillsdale, pp. 100–111.

Ben-Akiva, M.E., Bierlaire, M., Koutsopoulos, H.N., Mishalani, R.G., 1996. DynaMIT: Dynamic network assignment for the management of information to travelers. In: Proceedings of the 4th Meeting of the EURO Working Group on Transportation, Newcastle.

Binning, J.C., Crabtree, M.R., Burtenshaw, G.L., 2010. Transyt 14 user guide. Transport Road Laboratory Report nr AG48. APPLICATION GUIDE 65 (Issue F).

Bliemer, B.C.J., 2006. Dynamic queuing and spillback in an analytical multiclass dynamic traffic assignment model. In: Proceedings of the 1st International Symposium on Dynamic Traffic Assignment, 21–23 June 2006. The University of Leeds, UK.

Bretherton, R.D., Wood, K., Bowen, G.T., 1998. SCOOT Version 4. In: Proceedings IEE 9th International Conference on Traffic monitoring and Control. London.

Burghout, W., 2004. Hybrid microscopic-mesoscopic traffic simulation. Doctoral Dissertation. Royal Institute of Technology.

Cantarella, G.E., de Luca, S., Di Pace, R., Memoli, S., 2015. Network Signal Setting Design: Meta-heuristic optimisation methods. Transp. Res. Part C: Emerg. Technol. 55, 24–45.

Cantarella, G.E., de Luca, S., Di Pace, R., Memoli, S., 2014. Signal setting design at a single junction through the application of genetic algorithms. In: Computer-based Modelling and Optimization in Transportation. Springer International Publishing, pp. 321–331.

Cascetta, E., Cantarella, G.E., Di Gangi, M., 1991. Evaluation of control strategies through a doubly dynamic assignment model. Transportation Research Rec. N.1306.

Celikoglu, H.B., Dell'Orco, M., 2007. Mesoscopic simulation of a dynamic link loading process. Transp. Res. C: Emerg. Technol. 15 (5), 329–344.

Celikoglu, H.B., Gedizlioglu, E., Dell'Orco, M., 2009. A node-based modeling approach for the continuous dynamic network loading problem. IEEE Trans. Intell. Transp. Syst. 10 (1), 165–174.

Ceylan, H., Ceylan, H., 2012. A hybrid harmony search and TRANSYT hill climbing algorithm for signalized stochastic equilibrium transportation networks. Transp. Res. C: Emerg. Technol. 25, 152–167.

Chard, B.M., Lines, C.J., 1987. Transyt—the latest developments. Traffic Eng. Control 28 (1987), 387–390.

Daganzo, C.F., 1994. The Cell-transmission Model. Part 2: Network Traffic. University of California, Berkeley, California.

Daganzo, C.F., 1995. The cell transmission model II: network traffic. Transp. Res. Part B 29, 79–93.

de Romph, E., 1994. A dynamic traffic assignment model—theory and applications (Ph.D. dissertation). Fac. Civil Eng., Delft Univ. Technol., Delft, The Netherlands.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: NSGA II. IEEE Trans. Evolut. Comput. 6, 182–197.

Dell'Orco, M., 2006. A dynamic network loading model for mesosimulation in transportation systems. Eur. J. Oper. Res. 175 (3), 1447–1454.

Dial, R.B., 1971. A probabilistic multi-path traffic assignment model which obviates path enumeration. Transp. Res. 5, 83–111.

Di Gangi, M., 2011. Modeling evacuation of a transport system: application of a multimodal mesoscopic dynamic traffic assignment model. IEEE Trans. Intell. Transp. Syst. 12 (4), 1157–1166.

Di Gangi, M., 1996. Modelling dynamic network loading on transportation networks through a continuous packet approach. Math. Modell. Syst. 2 (3), 175–196.

Di Gangi, M., 1992. Continuous-flow approach in dynamic network loading. In: Second International CAPRI Seminar on Urban Traffic Networks, Compendium, vol. I.

Doan, K., Ukkusuri, S.V., 2012. On the holding back problem in the cell transmission based dynamic traffic assignment models. Transp. Res. Part B 46 (9), 1218–1238.

Drake, J.S., Schofer, J.L., May, A.D., 1967. A statistical analysis of speed density hypotheses. Third International Symposium on the Theory of Traffic Flow Proceedings. Elsevier North Holland, Inc., New York.

Drew, D.R., 1968. Traffic Flow Theory and Control. McGraw-Hill Book Company (Chapter 12).

Gartner, N.H., 1983. OPAC: a demand-responsive strategy for traffic signal control. Transp. Res. Rec. 906, 75–81.

Gazis, D.C., Potts, R.B., 1963. The oversaturated intersection. In: Proceedings of the 2nd International Symposium on Traffic Theory, London, U.K., pp. 221–237.

He, Y., 1997. A flow-based approach to the dynamic traffic assignment problem: formulations, algorithms, and computer implementations (Doctoral dissertation, Massachusetts Institute of Technology).

Henry, J.J., Farges, J.L., Tuffal, J., 1983. The PRODYN real time traffic algorithm. In: Proceedings of the 4th IFAC-IFIP-IFORS Conference on Control in Transportation Systems, pp. 307–311.

Highway Capacity Manual, 1965. Highway Research Board. Special Report 87.

Hunt, P.B., Robertson, D.I., Bretherton, R.D., Winton, R.I., 1981. SCOOT – a traffic responsive method of coordinating signals. RRL Report LR 1041, Road Research Laboratory, U.K.

Jayakrishnan, R., Mahmassani, H.S., Hu, T.-Y., 1994. An evaluation tool for advanced traffic information and management systems in urban network. Transp. Res. C 2, 129–147.

Kuwahara, M., Akamatsu, T., 1997. Decomposition of the reactive dynamic assignment with queues for a many-to-many origin-destination pattern. Transp. Res. Part B 31.

Leonard, D.R., Gower, P., Taylor, N.B., 1989. CONTRAM: structure of the model. TRRL Research Report 178, Crowtorne.

Luk, J.Y.K., 1984. Two traffic-responsive area traffic control methods: SCAT and SCOOT. Traffic Eng. Control 25, 14–22.

Mahut, M., Florian, M., Tremblay, N., 2002. Application of a simulation-based dynamic traffic assignment model. In: IEEE 5th International Conference on Intelligent Transportation Systems, Singapore.

Mauro, V., 2002. UTOPIA—Urban Traffic Control—Main Concepts, Presented at the EU–China ITS Workshop, Beijing, China.

Mauro, V., Di Taranto C., 1989. Utopia. In: Proc. of the 2nd IFAC-IFIP-IFORS Symposium on Traffic Control and Transportation Systems, pp. 575–597.

Merchant, D.K., Nemhauser, G.L., 1978a. A model and an algorithm for the dynamic traffic assignment problem. Transp. Sci. 12, 183–199.

Merchant, D.K., Nemhauser, G.L., 1978b. Optimality conditions for a dynamic traffic assignment model. Transp. Sci. 12 (3), 200–207.

Ozan, C., Baskan, O., Haldenbilen, S., Ceylan, H., 2015. A modified reinforcement learning algorithm for solving coordinated signalized networks. Transp. Res. Part C: Emerg. Technol. 54, 40–55.

Ran, B., Boyce, D.E., 1996. Modeling Dynamic Transportation Networks: An Intelligent Transportation System Oriented Approach, second ed. Springer-Verlag, Berlin.

Robertson, D.I., 1969. TRANSYT: a traffic network study tool. RRL Report LR 253, Road Research Laboratory, England.

Robertson, D.I., Bertherton, R.D., 1974. Optimum control of an intersection for any known sequence of vehicular arrivals. In: Proceedings of the 2nd IFAC-IFIP-IFORS Symposium on Traffic Control and Transportation System, Monte Carlo.

Smith, M.J., 1983. The Existence of an Equilibrium Distribution of Arrivals at a Single Bottleneck. Transp. Sci. 18, 385–394.

Spiess, H., 1990. Technical note—conical volume-delay functions. Transp. Sci. 24 (2), 153–158.

Stevanovic, A., Kergaye, C., Martin, P.T., 2008. Field evaluation of SCATS traffic control in park city, UT, Presented at 15th World Congress on ITS, New York City.

Stevanovic, A., Kergaye, C., Martin, P.T., 2009. SCOOT and SCATS: a closer look into their operations, 09-1672. In: Proceedings of the 88th Annual Meeting of the Transportation Research Board, Washington, D.C.
Taylor, N.B., 2003. The CONTRAM dynamic traffic assignment model. Netw. Spat. Econ. 3 (3), 297–322.
Underwood, R.T., 1961. Speed, Volume, and Density Relationship. Quality and Theory of Traffic Flow, Yale Bur. Highway Traffic, New Haven, Connecticut, pp. 141–188.
Vincent, R.A., Mitchell, A.I., Robertson, D.I., 1980. User guide to TRANSYT version 8. Transport and Road Research Laboratory Report, LR888, Crowthorne, Berkshire, U.K.
Van Aerde, M., Yagar, S., 1988. Dynamic integrated freeway/traffic signal networks: problems and proposed solutions. Transp. Res. Rec. A 22A (6), 435–443.