Transportation Research Part C 66 (2016) 48–60

Contents lists available at ScienceDirect

Transportation Research Part C

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

Bee Colony Optimization for innovative travel time estimation, based on a mesoscopic traffic assignment model

Mauro Dell'Orco [a,*], Mario Marinelli [a], Mehmet Ali Silgu [b]

[a] D.I.C.A.T.E.Ch., Technical University of Bari, Via Orabona 4, 70125 Bari, Italy
[b] Institute of Science and Technology, Technical University of Istanbul, Ayazaga Campus, Maslak, 34469 Istanbul, Turkey

ARTICLE INFO

Article history:
Received 31 January 2015
Received in revised form 22 September 2015
Accepted 1 October 2015
Available online 21 October 2015

Keywords:
Traffic flow
Travel time
Mesoscopic simulation
Bee Colony Optimization

ABSTRACT

In this article, we propose a framework for travel time prediction based on a time-discrete, mesoscopic traffic flow model, in which the measure of travel time is obtained as a link performance resulting from a dynamic network loading process. The spatiotemporal flow propagation on the road network is simulated incorporating the mesoscopic model and a linear link performance model, based on a travel time function. Acceleration levels are calculated explicitly, as a result of a fixed point problem. The traffic assignment to the network has been carried out through a completely new model, based on the Bee Colony Optimization (BCO) metaheuristics. In comparison with results of simulations carried out by using another mesoscopic model (DYNASMART), the travel times obtained with the proposed method appear more realistic.

1. Introduction

The interaction among transportation systems and land use characteristics, in the absence of appropriate planning strategies, creates an increasing traffic demand, which strives more and more the transport infrastructure, and causes chronic congestions. Such congestions result in an increase of generalized traveling costs, and in particular of travel time.

In recent years, travel time has become a focal point in many studies within the frame of both transportation planning and traffic engineering. The travel time measure keeps being not only an important criterion for the performance evaluation of transportation systems but also a major component for advanced innovative information systems. Accurate estimation and prediction of travel times and real-time traffic information are important factors that directly affect the viability of dynamic traffic systems' implementations. Thus, an adequate procedure for travel time estimation can be useful in avoiding inherent difficulties of real-time measurements on travel times. On one hand, a reasonable amount of large-scale studies on network-wide spatiotemporal traffic flow characteristics has been conducted considering explicitly travel times. On the other hand, at relatively smaller scales the specification of link-path flow characteristics is still an essential research topic. However, determining travel times in a correct and realistic, and thus reliable, manner is a common point in both cases.

In this work, we present a novel DTA-based travel time estimation model that lies its basis on a mesoscopic dynamic network loading, firstly presented by Dell'Orco (2006).

Representation of anisotropic traffic flow properties is a general problem in mesoscopic models; in our model, as time step reduces, the influence of vehicles that gradually enter a link on vehicles in front decreases, but cannot be ruled out entirely. For this reason, the proposed model is not completely anisotropic, but quasi-anisotropic.

* Corresponding author. Tel.: +39 0805963327; fax: +39 0805963329.
E-mail address: mauro.dellorco@poliba.it (M. Dell'Orco).

http://dx.doi.org/10.1016/j.trc.2015.10.001
0968-090X/

We have used the Sioux-Falls City road network to perform a dynamic network loading process by simulating the spatial and temporal flow propagation. Then, we have analyzed the performances of our model in reconstructing travel times on a road network through a completely new traffic assignment model, based on the Bee Colony Optimization (BCO) metaheuristics. In addition, we carried out a comparative evaluation, considering average speed and travel time obtained both by our model and by a popular commercial software.

In the following, we have first examined a variety of studies concerning fundamentally the travel time estimation/reconstruction, and then we have considered the most common mesoscopic models. Afterwards, the performance of our model in a reliable reconstruction of travel times is then compared to that obtained using DYNASMART. Finally, we conclude the paper with an evaluation of findings and future solution directions on the topic of interest.

2. Travel time estimation: a short literature review

2.1. Link travel time estimation

In mathematical analyses of transportation problems, an extensive variety of practical and theoretical studies taking into account travel time as a performance measure. Mainly, there are two approaches for performing travel time estimation: direct estimation by measurement, or indirect estimation.

2.1.1. Direct estimation of travel time by measurement

A measure of travel time can be derived in several ways by utilizing various types of traffic data collection systems. For example, you can track vehicles equipped with a cellular phone in terms of their cell location (Astarita and Florian, 2001; Pathirana et al., 2006). However, the accuracy of this method is heavily influenced by the size of the cellular phone network cell and the transmission quality (Pathirana et al., 2006).

Other approaches include automatic vehicle identification (AVI) (Turner et al., 1998), license plate recognition (Anagnostopoulos et al., 2006), tag reading and Bluetooth (Haghani et al., 2010).

Alternatively, probe vehicles equipped with a global positioning system (GPS) module and a transmission interface can transmit information regarding their spatial position and speed to a service center (Brockfeld et al., 2007). Methods for performing travel time estimation using GPS and GIS technologies have been proposed in addition to studies that explicitly consider the collection scheme of traffic data (Turner et al., 1998). In this study, to evaluate the performance of the flow model, we have assumed that travel time measurements obtained by a GPS equipped probe vehicle are actual travel times.

2.1.2. Indirect estimation of travel time

Indirect methods of travel time estimation can be subdivided into two main categories: trajectory based and traffic flow model based.

Trajectory methods are conceived to be the simplest and most widely accepted methods for the estimation of travel time from traffic sensor/detector data (Cortes et al., 2001). The trajectory methods proceed with the assumption that the point estimates of speed are representative of the average speed between contiguous detection points.

Nagel and Schreckenberg (1992) introduce a stochastic discrete automata model to freeway traffic. Through Monte-Carlo simulations, the model shows that, with increasing vehicle density, a shift from a laminar traffic flow to start-and-stop waves occurs, like in real freeway traffic.

Van Lint and Van der Zijpp (2003) propose an algorithm for the off-line estimation of route-level travel times for uninterrupted traffic flow facilities, based on an improved trajectory method; afterwards, Van Lint (2004) proposes and analyzes some different models and methods. Still Van Lint et al. (2005) proposes a framework for prediction of freeway travel time, both accurate and robust with respect to missing or corrupt input data, to be applied in a real-time environment.

Wang et al. (2014) based their citywide, real-time model for estimating the travel time of any path on the GPS trajectories of vehicles received in current time slots and over a period of history as well as map data sources.

Several authors, like Hoogendoorn (2000), Coifman (2002), and Oh et al. (2003), just to mention a few, have presented traffic flow theory-based models, ensuring initial and boundary conditions as well as the flow conservation principle.

However, some of the models perform well solely in case of free or almost free flow patterns (Hoogendoorn, 2000; Oh et al., 2003), while some other models are only applicable for congested flow patterns (Nam and Drew, 1998).

Treiber and Helbing (2002) proposed a robust adaptive smoothing method for the spatiotemporal approximation of flow patterns, which takes into account the characteristic propagation speeds observed in both uncongested and congested conditions.

Corthout et al. (2012) focused on the problem of possible non-uniqueness of the solution of Dynamic Network Loading (DNL) models and provided approaches on how one can deal with.

Bekhor et al. (2013) presented a system for the collection and analysis of free-flow travel speeds on a road network, mainly focusing on the road safety and possible counter-measures employed to reduce excessive speeds.

2.2. Mesoscopic models

Mesoscopic models hold the middle between the macroscopic and microscopic representation of traffic since they allow the explicit tracking of vehicles with aggregate link performances. In these models, vehicles are grouped together in packets, represented by a single point, usually the head of the packet (CONTRAM by Leonard et al., 1989; Cascetta et al., 1991). DynaMIT (Ben-Akiva et al., 1998) uses a cell model for vehicle movements along the link. A cell is considered as a platoon of vehicles that travels with a certain speed over the link. Overtaking is not allowed. In other models (see for example DYNASMART by Jayakrishnan et al., 1994), individual vehicles move following macroscopic speed/density relationships on the link, and a queue server at the nodes accounts for delays caused by traffic signals and interactions with traffic from different directions.

Burghout et al. (2005) proposed a hybrid mesoscopic–microscopic model (MEZZO), which applies a microscopic simulation to areas of specific interest, and a mesoscopic simulation to the surrounding network. In the first case, all vehicle movements and driver decisions are explicitly simulated; in the second case, aggregated behaviors are used to simulate the surrounding network.

Despite the wide range of models, the need for efficient estimation methods, to provide consistent reconstructions for various flow patterns, still exists. Therefore, the main motivation for the present study is the simulation of spatial and temporal propagation of flow dynamics; we have followed a simplified dynamic network loading process to reconstruct dynamic traveling times satisfying the flow propagation consistency over the road network.

3. Proposed modelling approaches

3.1. Dynamic network loading model

In this study, we analytically formulate a quasi-anisotropic flow model and seek the solution through a simulation approach. The proposed model is not completely anisotropic because it suffers from a common problem of mesoscopic models: vehicles gradually entering a link influence the speed of vehicles in front of them. However, in our model, as time step reduces, this influence decreases, but cannot be ruled out entirely. For this reason, we called the model "quasi-anisotropic". The measure of travel time is obtained as a link performance, resulting from a dynamic network loading process. We followed the link-performance function approach in order to obtain travel times by an explicit linear formulation (Friesz et al., 1993). Then, we assessed the performance of the proposed model also by comparisons to results obtained by a different model.

In travel time function based link performance formulations, the travel time of a user entering a link i at time t is introduced as $\tau^i(t)$. As the propagation of flows through a link is described by the relationships between flow variables at each point in time, the travel time $\tau^i(t)$ can be expressed as given by Eq. (1), derived from the pioneering travel time function based formulation of Friesz et al. (1993),

$$\tau^i(t) = f\left(u^i(t), n^i(t), w^i(t)\right) \tag{1}$$

where $u^i(t)$, $n^i(t)$, and $w^i(t)$ are respectively the inflow to, number of vehicles at, and outflow from link i at time t. Most of the travel-time-based performance formulations proposed in the literature are special cases of the form given in Eq. (1). See, for example, the travel time function-based approaches (Friesz et al., 1993; Rubio-Ardanaz et al., 2003; Celikoglu, 2007), or the link exit function-based approaches (Merchant and Nemhauser, 1978; Wie et al., 1995). Instead, the proposed method calculates link traveling times as a function of flow speed, with accelerations and decelerations resulting from a mesosimulation process (Dell'Orco, 2006; Celikoglu and Dell'Orco, 2007; Celikoglu et al., 2009), and tries to satisfy a two-phase, i.e. congested and uncongested, fundamental diagram by simulating flow dynamics. In the following, the theoretical background of the employed method, for dynamic link/path loading is provided.

3.1.1. Model formulation

Let P be the set of feasible paths on the network; the set of vehicles, leaving in the same interval j and following the same path p ($p \in P$) is called a packet (j, p). We have studied the movement of vehicles for discrete time intervals $[t - \Delta t]$, $[t + \Delta t]$; in these reference periods, speed can be assumed stepwise constant or uniformly accelerated and decelerated. The speed of a generic packet (j, p) is generally both a function of space s (and thus of the entry time on the link, $t_{j,p}$) and of the time $tV = f(s, t)$, where $s = f(t, t_{j,p})$. In literature, most of the existing approaches are simplified: the speed is assigned to each packet on entering the link and does not change over time ($\partial V/\partial t = 0$), or is assumed equal for all packets running at the same time t on the link ($\partial V/\partial s = 0$). Considering a generic link in which:

$m_{j,p}$	is the total number of vehicles belonging to a packet (j, p);
$a^i(t)$	is the acceleration (or deceleration) during the interval $[t + \Delta t]$ constant and common, in the reference period, for all vehicles on link i
$n^i_{j,p}(t)$	is the number of vehicles in a packet (j, p) on link i at time t;
$s^i_{j,p}(t)$	is the position on link i of the head of the packet (j, p) at time t.

$N^i(t)$ is the number of vehicles in exiting flow from link i at time t;
$N^i(t)'$ is the number of vehicles that exceeds the capacity and cannot exit from link i at time t;
d^i is the length of link i;
$V^i(t)$ is the speed at time t on link i, common to all vehicles on the link;
$w^i(t)$ is the exiting flow from link i at time t; and speed is a function of average density $k^i(t) : V^i(t) = V^i\left(k^i(t)\right)$, where $k^i(t) = N^i(t)/d^i$.

Therefore, the total number of vehicles on a link can be obtained as $n^i(t) = \sum_{p\in P}\sum_{j\leqslant t} n^i_{j,p}(t)$.

In literature, a great number of mesoscopic models of traffic flow exists. In these models, any number of groups of objects can exist at the same time and interactions between them can be implemented.

Recently, dynamic mesoscopic network loading problems have been categorized based on whether vehicles are grouped into discrete or continuous packets. Because of inherent difficulties of the continuous packet approach, in this study we have used the discrete packet approach. A discrete packet is made up of a pile of vehicles, all located at the head of the packet: if the head of a packet occupies a link during interval $[t + \Delta t]$, all vehicles belonging to that packet are assumed to occupy the same link. The speed has been assumed equal for all packets running on link i at the same time $t(\partial V/\partial s = 0)$ and acceleration (or deceleration) is constant and common for all vehicles on the link in the reference period. In addition to the assumption that the movement of vehicles is uniformly accelerated, the existence of a relation between speed and density is assumed. Therefore, the variables $V^i(t)$, $n^i_{j,p}(t)$, and $s^i_{j,p}(t)$ can be derived from relationships given by Eqs. (2)–(5).

In particular, the method starts setting $V^i(0) = 4 \cdot \frac{C^i}{\gamma^i_{max}}$ and $s^r_{j,p}(0) = 0$, then calculates $s^i_{j,p}(t)$ as a function of $V^r(t-\Delta t)$, $V^i(t)$, and $s^r_{j,p}(t-\Delta t)$:

$$s^i_{j,p}(t) = s\left(V^r(t-\Delta t), V^i(t), s^r_{j,p}(t-\Delta t)\right) \tag{2}$$

With the updated value of $s^r_{j,p}(t)$, it is possible to calculate $n^i_{j,p}(t)$ as a function of $s^i_{j,p}(t)$ and $s^i_{j+1,p}(t)$:

$$n^i_{j,p}(t) = n\left(s^i_{j,p}(t), s^i_{j+1,p}(t)\right) \tag{3}$$

then, the method updates the value of $N^i(t)$ as a function of $N^i(t-\Delta t)$ and $n^i_{j,p}(t)$:

$$N^i(t) = N\left(N^i(t-\Delta t)', n^i_{j,p}(t)\right) \tag{4}$$

Finally, it is possible to obtain the new value of $V^i(t)$ as a function of the updated value of $N^i(t)$:

$$V^i(t) = V\left(N^i(t)\right) \tag{5}$$

In equations from (2)–(5), $r = i$ or $i-1$, according to the link which the movement of the packet (j,p) starts from, $n^i_{j,p}(t) = 0$ when $j > t$, and s, n and V are continuous functions. The outflow at time t on link i, $w^i(t)$, is also obtained as a function of $N^i(t)$ as given by Eq. (6):

$$w^i(t) = w\left(N^i(t)\right) \tag{6}$$

The speed can be assumed equal for all vehicles on the link, or assigned to each packet at the entry time. In the first case, $V^i(t) = V\left(N^i(t)\right)$ and, therefore, the speed of each packet depends on the vehicle number behind the packet itself. However, the influence of this assumption on the model is lower as Δt approaches zero ($\Delta t \to 0$). In fact, the more Δt approaches zero, the more the number of vehicles entering the link in that time interval reduces, thus reducing their influence on the common speed on the link, calculated as in Eq. (5). Because of this approximation, the model based on the relationships given by Eq. (3) shifts to the state given by Eq. (7).

$$n^i_{j,p}(t) = \begin{cases} 0, & \text{if } s^i_{j,p}(t) = 0 \\ m_{j,p}, & \text{if } s^i_{j,p}(t) > 0 \end{cases} \tag{7}$$

The expressions of the mesosimulation model evolve to the equations given in the following. Eqs. (8) and (9) are analogous to Eqs. (2) and (7), related to the time interval $[t, t+\Delta t]$:

$$s^i_{j,p}(t+\Delta t) = s\left(V^r(t), V^i(t+\Delta t), s^r_{j,p}(t)\right) \tag{8}$$

$$n^i_{j,p}(t+\Delta t) = \begin{cases} 0, & \text{if } s^i_{j,p}(t+\Delta t) = 0 \\ m_{j,p}, & \text{if } s^i_{j,p}(t+\Delta t) > 0 \end{cases} \tag{9}$$

Eq. (10) provides the number of vehicles on the link i for the same time interval:

$$n^i(t+\Delta t) = \sum_{p\in P}\sum_{j\leqslant t} n^i_{j,p}(t+\Delta t) \tag{10}$$

The total number of vehicles on the link i at time t can be obtained from Eq. (11), taking into account also $N^i(t)'$:

$$N^i(t+\Delta t) = N\left(N^i(t)', n^i(t+\Delta t)\right) \tag{11}$$

According to Eqs. (5) and (6), we can now calculate the speed and the outflow at time $t+\Delta t$:

$$V^i(t+\Delta t) = V\left(N^i(t+\Delta t)\right) \tag{12}$$

$$w^i(t+\Delta t) = w\left(N^i(t+\Delta t)\right) \tag{13}$$

Considering the physical assumptions on motion, the speed and space variables can be obtained as given by Eqs. (14) and (15) respectively.

$$V^i(t+\Delta t) = V^i(t) + (a^i(t)\cdot\Delta t) \tag{14}$$

$$s^i_{j,p}(t+\Delta t) = s^i_{j,p}(t) + \left(V^{i-1}(t)\cdot\Delta t\right) + \left(\frac{1}{2}\cdot a^i(t)\cdot\Delta t^2\right) \tag{15}$$

Therefore, the travel time on a link within the reference period $[t, t+\Delta t]$ can be calculated as given by Eq. (16).

$$\tau^i(t+\Delta t) = T\left(V^i(t), a^i(t), s^i_{j,p}(t), \Delta t\right) \tag{16}$$

The fixed-point problem on the variable $V^i(t+\Delta t)$, formulated through the Eqs. (8)–(13) is resolved as by Celikoglu and Dell'Orco (2007) and Celikoglu et al. (2009). The speed $V^i(t+\Delta t)$ is at first calculated through Eq. (14): $V^i(t+\Delta t) = V^r(t) + a^i(t)\cdot\Delta t$, where $r = i$ or $i-1$, according to the link which the movement of the packet (j,p) starts from. Then, applying the model (Eqs. (8)–(13)), the speed value is calculated through successive iterations as given in Eq. (17), where $V^i_y(t+\Delta t)$ is the value of speed $V^i(t+\Delta t)$ at iteration y.

$$V^i_{y+1}(t+\Delta t) = \left(\frac{1}{y}\cdot V\left(N\left(n\left(s\left(V^i_y(t+\Delta t)\right)\right)\right)\right)\right) + \left(\frac{(y-1)}{y}\cdot\left(V^i_y(t+\Delta t)\right)\right) \tag{17}$$

The iteration stops when the difference between two consecutive speed values is not greater than a fixed threshold. Then, the current value of acceleration (or deceleration) is calculated as given in Eq. (18) and used as input in successive calculations:

$$a^i(t+\Delta t) = \frac{\left(V^i(t+\Delta t) - V^r(t)\right)}{\Delta t} \tag{18}$$

The constraint given through Eq. (19) calls for the adjustment of the number of vehicles on a link during the above-mentioned iteration process, where C^i denotes the capacity of the link i:

$$N^i(t+\Delta t) = \begin{cases} N^i(t)' + n^i(t+\Delta t) & \text{if } N^i(t)' + n^i(t+\Delta t) \leqslant C^i\cdot\Delta t \\ C^i\cdot\Delta t & \text{if } N^i(t)' + n^i(t+\Delta t) > C^i\cdot\Delta t \end{cases} \tag{19}$$

The following pseudo-code may help to clarify the simulation procedure.

Step 0
INITIALIZE
$n^i_{j,p}(0) = 0$; $s^i_{j,p}(0) = 0$; $N^i(t)' = 0$; $V^i(0) = 4\cdot\frac{C^i}{\gamma^i_{\max}}$; $a^i(0) = 0$
DO
Step 1
COMPUTE SEQUENTIALLY
$s^i_{j,p}(t+\Delta t) = s^i_{j,p}(t) + \left(V^i(t)\times\Delta t\right) + (\frac{1}{2}\times a^i(t)\times\Delta t^2)$

$$n^i_{j,p}(t+\Delta t) = \begin{cases} 0, & \text{if } s^i_{j,p}(t+\Delta t) = 0 \\ m_{j,p}, & \text{if } s^i_{j,p}(t+\Delta t) > 0 \end{cases}$$

$n^i(t+\Delta t) = \sum_{p\in P}\sum_{j\leqslant t} n^i_{j,p}(t+\Delta t)$

$N^i(t+\Delta t) = \begin{cases} N^i(t)' + n^i(t+\Delta t) & \text{if } N^i(t)' + n^i(t+\Delta t) \leqslant C^i \times \Delta t \\ C^i \times \Delta t & \text{if } N^i(t)' + n^i(t+\Delta t) > C^i \times \Delta t \end{cases}$

$V^i(t+\Delta t) = 4 \cdot \frac{C^i}{\gamma^i_{\max}} \cdot \left(1 - \frac{N^i(t+\Delta t)}{d^i \cdot \gamma^i_{\max}}\right)$

$a^i(t+\Delta t) = \frac{\left(V^i(t+\Delta t) - V^i(t)\right)}{\Delta t}$

Step 2

The current iteration clock is y. SET ε = 0.1

DO

COMPUTE

$V^i_{y+1}(t+\Delta t) = \left(\frac{1}{y} \times \left(V^i(t) + a^i_y(t+\Delta t) \times \Delta t\right)\right) + \left(\frac{(y-1)}{y} \times \left(V^i_y(t+\Delta t)\right)\right)$

$a^i_{y+1}(t+\Delta t) = \frac{V^i_{y+1}(t+\Delta t) - V^i(t)}{\Delta t}$

$s^i_{j,p}(t+\Delta t) = s^i_{j,p}(t) + \left(V^i(t) \times \Delta t\right) + \left(\frac{1}{2} \times a^i(t) \times \Delta t^2\right)$

$n^i_{j,p}(t+\Delta t) = \begin{cases} 0, & \text{if } s^i_{j,p}(t+\Delta t) = 0 \\ m_{j,p}, & \text{if } s^i_{j,p}(t+\Delta t) > 0 \end{cases}$

$n^i(t+\Delta t) = \sum_{p\in P}\sum_{j\leqslant t} n^i_{j,p}(t+\Delta t)$

$N^i(t+\Delta t) = \begin{cases} N^i(t)' + n^i(t+\Delta t) & \text{if } N^i(t)' + n^i(t+\Delta t) \leqslant C^i \times \Delta t \\ C^i \times \Delta t & \text{if } N^i(t)' + n^i(t+\Delta t) > C^i \times \Delta t \end{cases}$

$V^i(t) = V^i_{y+1}(t+\Delta t)$

$V^i_{y+1}(t+\Delta t) = 4 \cdot \frac{C^i}{\gamma^i_{\max}} \cdot \left(1 - \frac{N^i(t+\Delta t)}{d^i \cdot \gamma^i_{\max}}\right)$

$y = y + 1$

WHILE $\left|\frac{\left(V^i_{y+1}(t+\Delta+) - V^i(t)\right)}{V^i(t)}\right| \leqslant \varepsilon$

WHILE $N^i(t+\Delta t) > 1$

3.1.2. Adherence to the fundamental diagram approach

In congested flow states, when the number of vehicles exceeds the capacity of a link, we assumed the existence of a buffer area in the upstream node of the link, for temporary storage of vehicles exceeding the capacity. In order to avoid circulation blockage, flow is assumed to creep with a minimum speed of 8 km/h in a congested state when the density exceeds the maximum. In a under-saturated flow state, the speed-density relationship by Greenshields (1935) reported in Eq. (20) is used to compute the speed $V^i(t)$ of each packet. The value of free flow speed on link i, V^i_{ff}, is given by Eq. (21), where $\gamma^i_{\max}$ is the maximum density on, and C^i is the capacity of link i; the average density $\gamma^i(t)$ is given by Ed. 22. By substituting Eqs. (21) and (22) in Eq. (20) results Eq. (23).

$$V^i(t) = V^i_{ff} \cdot \left(1 - \frac{\gamma^i(t)}{\gamma^i_{\max}}\right) \tag{20}$$

$$V^i_{ff} = 4 \cdot \frac{C^i}{\gamma^i_{\max}} \tag{21}$$

$$\gamma^i(t) = N^i(t)/d^i \tag{22}$$

$$V^i(t) = 4 \cdot \frac{C^i}{\gamma^i_{\max}} \cdot \left(1 - \frac{N^i(t)}{d^i \cdot \gamma^i_{\max}}\right) \tag{23}$$

4. Modelling approach

To solve the traffic assignment problem on a network by the Quasi-Anisotropic Mesosimulation Model (QAMM), we have implemented an original model, based on the Bee Colony Optimization (BCO). Note that the DNL mesoscopic model itself is not new, as it is based on a mesoscopic dynamic network loading, firstly presented by Dell'Orco (2006). It has lately been proposed (Celikoglu and Dell'Orco, 2008) as a tool for acceleration behavior description in congested traffic. BCO method too has been previously presented as Bee System (BS) by Lučić and Teodorović (2001), and tested on a large number of numerical examples. However, in these previous papers an out-and-out assignment model was missing; thus, the novelty of this paper is coupling the mesoscopic model with the BCO to obtain a new assignment model.

In our approach, the rate of flow assigned to a single link diverging from each node of the traffic network should be optimized by considering all possible assignments to paths diverging from that node. In this sense, the traffic assignment is a complex combinatorial problem. The BCO, which we tailored for the specific problem in this work, represents a new meta-heuristic capable of solving difficult combinatorial optimization problems. We have obtained that BCO is able to produce optimal or near-optimal solutions in a reasonable amount of computer time.

In the following, after a brief explanation of BCO, the new model is presented.

4.1. Bee Colony Optimization

Bee colonies in nature are studied in the framework of the so-called Swarm Intelligence (SI). First, Beni and Wang (1989) introduced the concept of Swarm Intelligence (SI) in the context of cellular robotic systems. Generally, SI systems are made of a population of agents that individually do not show an "intelligent" behavior. However, local interactions with each other and with their environment lead to an "intelligent" global behavior. As well as bee colonies, examples of SI include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling.

Lučić and Teodorović (2001) first proposed an application of the Bee Colony Optimization (BCO) to combinatorial problems in Transportation Engineering. The BCO is a stochastic, random search technique that belongs to the class of population-based algorithms. This technique uses an analogy between the way in which bees in nature search for food, and the way in which optimization algorithms search for an optimum of combinatorial optimization problems. Since its appearance, it had various successful applications. The BCO has also been applied to the vehicle routing problem, the routing and wavelength assignment (RWA) in all-optical networks (Marković et al., 2007), the ride-matching problem (Teodorović and Dell'Orco, 2008), the traffic sensors locations problem on highways (Edara et al., 2008), and the static scheduling of independent tasks on homogeneous multiprocessor systems (Davidović et al., 2009).

Interesting surveys of the bees' behavior inspired algorithms could be found in Karaboğa and Akay (2009) and Teodorović (2009).

In the following, we summarize briefly the BC algorithm. For a more detailed explanation, refer to Teodorović et al. (2006) and Teodorović (2008).

At the beginning of the process, bees are located in the hive, represented by a square in Fig. 1. Then, they depart from the hive and fly through an artificial network, whose nodes represent specific traffic assignments (Fig. 1).

The bees' flights are divided into iterations. In every iteration, the bees' visit to an artificial node represents the choice of a specific traffic assignment. Thus, a bee's path through the artificial network represents a sequence of assignments chosen at time *t*. In Fig. 1, every artificial node of the physical network contains all possible paths, previously specified, between the given O–D pair. We have set in advance the number B of bees and the maximum number I of iterations.

During the search process, artificial bees communicate directly. When flying through space, the bees perform a forward pass or a backward pass. During the first forward pass (Fig. 1), every bee visits a possible assignment choosing splitting rates α_{ik} for each node. In every iteration, a bee chooses a new node in the artificial network. After the forward pass, bees perform a backward pass, returning to their hive.

In the hive, all bees participate in a decision-making process. We assume that every bee can obtain the information about the quality of the solutions generated by all other bees. That is, bees exchange information about the quality of the assignments. Every bee decides whether to: (i) abandon the created solution and become again uncommitted follower; (ii) continue to expand the same solution without recruiting the nestmates; or (iii) dance and thus recruit the nestmates before returning to the created partial solution.

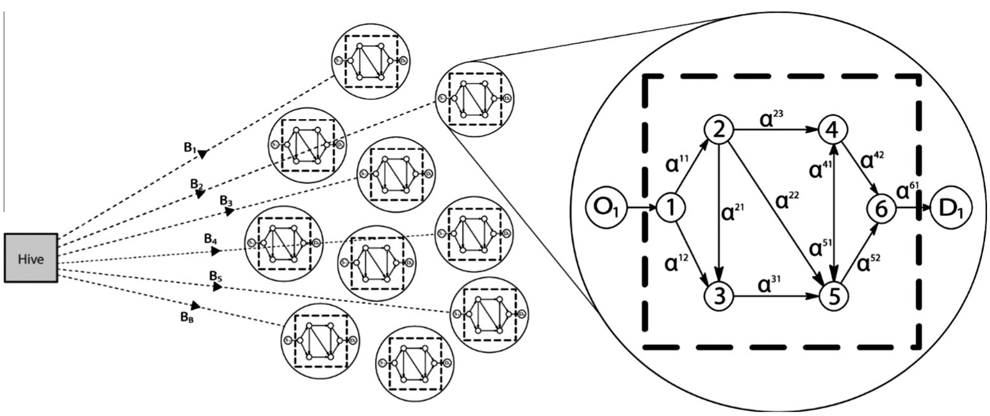

Fig. 1. Forward pass of BCO algorithm.

During the second forward pass, bees expand solutions previously created (trying to find a new network assignment) and then perform again the backward pass returning to the hive. In the hive bees again participate in a decision-making process, perform third forward pass, and so on. The best solution discovered during the first iteration is saved, and then the second iteration begins. Within the second iteration, bees again incrementally construct solutions of the problem. There is one or more created solutions at the end of each iteration.

The following pseudo-code of the Bee Colony Optimization is executed at each time step during simulation.

Initialization. Determine the number of bees B, the number of iterations I. Find a feasible solution x of the problem. This solution is the initial best solution.
Set i = 1. Until i = I, repeat the following steps:
Set j = 1. Until j = m, repeat the following steps:

Forward pass: Allow bees to fly from the hive and to choose splitting percentages for each network node.

Backward pass: Send all bees back to the hive. Allow bees to exchange information about quality of the assignment chosen and to decide whether to abandon the created solution and become again uncommitted followers, continue to expand the same solution without recruiting the nest mates, or dance and thus recruit the nestmates before returning to the created solution.
Set j = j + 1.
If the best solution x obtained during the ith iteration is better than the best-known solution, update the best-known solution (x = xi).
Set i = i + 1.

Alternatively, forward and backward passes could be performed until some other stopping condition is satisfied. The stopping condition is the maximum total number of forward/backward passes.

4.1.1. Traffic assignment model

We have obtained the traffic assignment on the sample network through the search of the System Optimum (SO) based on the marginal travel time. Peeta (1994) proposed a System Optimum approach introducing two types of marginal travel times that are global and local. Peeta and Mahmassani (1995) proposed a common approach in calculating marginal link travel times, and suggested seeking a three-point quadratic fitting using simulation result of three successive time steps.

In this work, we have defined the marginal time at time t $\overline{\tau_i}(t)$ as the additional cost that a unit of flow adds to the total travel cost experienced by the packets traveling the specified link at time t. This is evaluated by sending a virtual packet, with a specified number of vehicles, to travel in parallel with the actual packets. The marginal travel time is then calculated, based on the travel time difference between the virtual and actual packets, by the following equation:

$$\overline{\tau_i}(t) = \tau_i(N^i(t)) + \frac{\tau_i(N^i(t) + \Delta n) - \tau_i(N^i(t))}{\Delta n} \cdot N^i(t) \tag{24}$$

where Δn is the number of additional vehicles in the virtual packet that enters link i.

4.1.1.1. Node transfer and BCO-based Dynamic Traffic Assignment (DTA). Summary of notations:

- $G(N,L)$ = a transportation network, consisting of a set N of nodes, and a set L of directed links connecting the nodes;
- BW_k = set of links merging into a node k ($k \in N$);
- FW_k = set of links diverging from a node k ($k \in N$);
- C^r = capacity of link r ($r \in FW_k$).
- O = set of origin nodes o ($O \subset N$);
- D = set of destination nodes d ($D \subset N$);
- P = set of paths p, connecting o–d pairs;
- $u^{kr}(t)$ = total flow exiting from node k and entering the link r at time t;
- $u^{ikr}(t)$ = partial flow entering the link r from link i through the node k at time t;
- $w^{ki}(t)$ = total flow exiting from link i and entering the node k at time t;
- $w^{ikr}(t)$ = partial flow exiting from link i and entering the link r through the node k at time t;
- α^{ki} = flow splitting rate assigned to link i diverging from node k.

The following relationships between the total inflow $u^{kr}(t)$ and the partial flow $u^{ikr}(t)$, the total outflow $w^{ki}(t)$ and the partial outflow $w^{ikr}(t)$ hold:

$$u^{kr}(t) = \sum_{i \in BW_k} u^{ikr}(t) \tag{25}$$

$$w^{ki}(t) = \sum_{r \in FW_k} w^{ikr}(t) \tag{26}$$

subject to the following constraints:

$$u^{kr}(t) \leqslant C^r; \qquad \forall r \in FW_k, \quad \forall k \in N \tag{27}$$

$$\sum_{r \in FW_k} u^{kr}(t) \leqslant \sum_{i \in BW_k} w^{ik}(t); \qquad \forall i \in BW_k, \quad \forall r \in FW_k, \ \forall k \in N \tag{28}$$

$$\sum_{r \in FW_k} u^{kr}(t) = \begin{cases} \sum_{i \in BW_k} w^{ik}(t), & \text{if } \sum_{i \in BW_k} w^{ik}(t) \leqslant \sum_{i \in FW_k} C^r \\ \sum_{i \in FW_k} C^r & \text{if } \sum_{i \in BW_k} w^{ik}(t) > \sum_{i \in FW_k} C^r \end{cases}; \qquad \forall i \in BW_k, \quad \forall r \in FW_k \tag{29}$$

Eq. (27) states that the total flow entering the link r at time t must be not greater than the capacity of link r.

Eq. (28) states that, according to the flow conservation principle, the total flow demanding to enter diverging links at time t must be not greater than the total flow exiting from the merging links at time t.

Eq. (29) expresses the capacity constraints.

Splitting occurs when the flows are in conflict at a node with multiple merging and diverging links. Two split factors, path-based and link-based can be used in system dynamics. Path-based split factors are generally used in the network assignment context, while link-based splitting rates are considered more appropriate in modeling a single node, like in this study.

To handle the queues, we have made the hypothesis that vehicles exceeding the capacities of a link are temporarily stored in a virtual buffer λ located at the beginning of that link. The buffers are then gradually discharged as soon as the demand falls below the capacity.

The proposed model calculates split factors in a single node by considering all possible assignments to a diverging link. The rate of flow assigned to a single path is calculated through the BCO:

$$\alpha_p = \prod_{i \in p} \alpha^i \tag{30}$$

$$\min \ Z = \sum_{od \in OD} \sum_{p \in P_{od}} \left[\alpha_{p,t} \sum_{i \in I_p} \overline{\tau_i}(t) + \sum_{n \in N_p} N_{\lambda,t} \Delta t \right] \tag{31}$$

where $N_{\lambda,t}$ is the number of vehicles stored in the buffer λ at time t that represents a queue.

The aim of an optimization process is to solve the problem 31, which considers all paths' costs of all O–D pairs in the network at time t.

The optimization problem given by Eqs. (25)–(31) is solved at each time interval within the modeling horizon. The simulation process lasts as long as the inflows are completely discharged from the entire network. Due to dynamic nature of the model, at each time interval the procedure should evaluate all possible paths to a destination from every node reached by the traffic flow, facing in this way a "combinatorial explosion". To handle this problem, in this work we have used the Bee Colony Optimization (BCO) metaheuristics, finding the best paths at each time interval, and thus optimizing the traffic assignment.

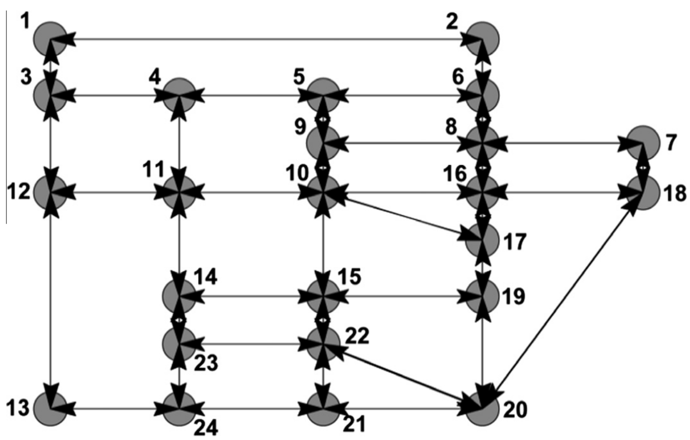

Fig. 2. Sioux-Falls City network.

Table 1
1 h O–D matrix.

Node numbers	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	7	7	37	15	22	37	59	37	95	37	15	37	22	37	37	29	7	22	22	7	29	22	7
2	7	0	7	15	7	29	15	29	15	44	15	7	22	7	7	29	15	0	7	7	0	7	0	0
3	7	7	0	15	7	22	7	15	7	22	22	15	7	7	7	15	7	0	0	0	0	7	7	0
4	37	15	15	0	37	29	29	51	51	88	103	44	44	37	37	59	37	7	15	22	15	29	37	15
5	15	7	7	37	0	15	15	37	59	73	37	15	15	7	15	37	15	0	7	7	7	15	7	0
6	22	29	22	29	15	0	29	59	29	59	29	15	15	7	15	66	37	7	15	22	7	15	7	7
7	37	15	7	29	15	29	0	73	44	139	37	51	29	15	37	103	73	15	29	37	15	37	15	7
8	59	29	15	51	37	59	73	0	59	117	59	44	44	29	44	161	103	22	51	66	29	37	22	15
9	37	15	7	51	59	29	44	59	0	205	103	44	44	44	66	103	66	15	29	44	22	51	37	15
10	95	44	22	88	73	59	139	117	205	0	293	147	139	154	293	323	286	51	132	183	88	191	132	59
11	37	15	22	110	37	29	37	59	103	286	0	103	73	117	103	103	73	7	29	44	29	81	95	44
12	15	7	15	44	15	15	51	44	44	147	103	0	95	51	51	51	44	15	22	29	22	51	51	37
13	37	22	7	44	15	15	29	44	44	139	73	95	0	44	51	44	37	7	22	44	44	95	59	59
14	22	7	7	37	7	7	15	29	44	154	117	51	44	0	95	51	51	7	22	37	29	88	81	29
15	37	7	7	37	15	15	37	44	73	293	103	51	51	95	0	88	110	15	59	81	59	191	73	29
16	37	29	15	59	37	66	103	161	103	323	103	51	44	51	88	0	205	37	95	117	44	88	37	22
17	29	15	7	37	15	37	73	103	66	286	73	44	37	51	110	205	0	44	125	125	44	125	44	22
18	7	0	0	7	0	7	15	22	15	51	15	15	7	7	15	37	44	0	22	29	7	22	7	0
19	22	7	0	15	7	15	29	51	29	132	29	22	22	22	59	95	125	22	0	88	29	88	22	7
20	0	7	0	22	7	22	37	66	44	183	44	37	44	37	81	117	125	29	88	0	88	176	51	29
21	7	0	0	15	7	7	15	29	22	88	29	22	44	29	59	44	44	7	29	88	0	132	51	37
22	29	7	7	29	15	15	37	37	51	191	81	51	95	88	191	88	125	22	88	176	132	0	154	81
23	22	0	7	37	7	7	15	22	37	132	95	51	59	81	73	37	44	7	22	51	51	154	0	51
24	7	0	0	15	0	7	7	15	15	59	44	37	51	29	29	22	22	0	7	29	37	81	51	0

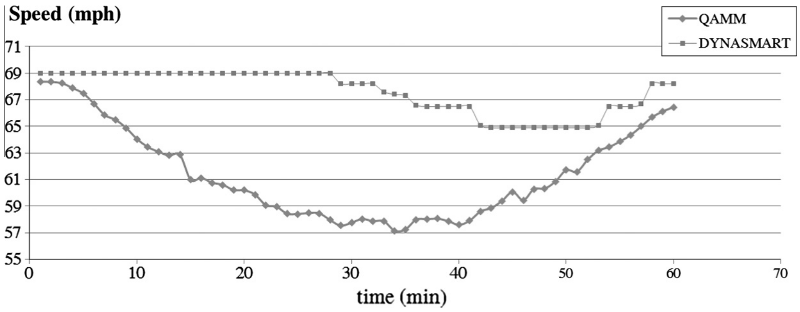

Fig. 3. Average speed (mph) obtained by QAMM and DYNASMART.

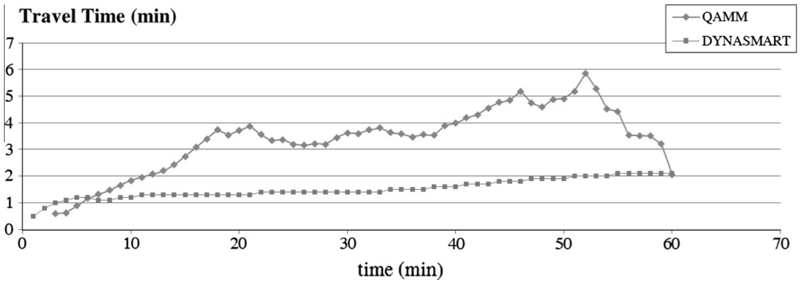

Fig. 4. Average actual travel time (min) obtained by QAMM and DYNASMART.

5. Test network and numerical results

We have used the Sioux-Falls City network illustrated in Fig. 2 to test the BCO-based DTA model. This network comprises 24 nodes and 76 arcs.

To evaluate the outcome of the proposed model, we have considered the OD matrix reported in Table 1 and compared the average speed and travel time obtained by our model to those obtained using DYNASMART. In particular, 552 OD pairs have been considered with a sinusoidal demand profile to load the network for 1 h. Demand values have been set to obtain a significant congestion level. Fig. 3 shows the average speed (mph) obtained during simulation with QAMM, carried out at the Technical University of Bari, and simulation with DYNASMART carried out at the Istanbul Technical University. The average actual travel times (min) obtained during simulation by QAMM and DYNASMART are reported in Fig. 4.

The differences in the graphs of Fig. 3 are mainly due to the different ways of calculating the speed in QAMM and in DYNASMART. Generally, mesoscopic models assume the speed as piecewise constant in the time intervals. This assumption introduces two drawbacks: (i) in each time interval, the speed is an average value, consequently smoothing maximum and minimum values, and possibly reducing the effects of congestion and (ii) passing from a time interval to a successive, the speed changes instantly, requiring an unrealistic value of acceleration. Instead, our model calculates for each time interval acceleration and speed as fixed-point values, providing in this sense more realistic values even in congested situations. In Celikoglu and Dell'Orco (2007) more extensive comparisons have been carried out, providing also numerical evidence. In this framework, we can say that our model, in comparison with the results of simulations with DYNASMART carried out at the Istanbul Technical University, has obtained a more realistic network dynamics. In fact, in our model the speed decreases as soon as the simulation starts and new packets enter the network while in DYNASMART remains constant for almost 30 min (Fig. 3). Moreover, in our model travel time decreases when the average speed increases and the demand flow decreases; instead, in DYNASMART the average travel time always increases despite the speed increases (Fig. 4).

In the light of the results presented in Fig. 4, we have obtained an effective travel time estimation, based on a traffic flow model. On its turn, the traffic assignment problem has been solved through an innovative metaheuristics, the Bee Colony Optimization.

6. Conclusions

In this study, a quasi-anisotropic traffic flow model is employed to reconstruct travel time in a road network, resulting from a dynamic network modeling process. The structure of the traffic assignment model is based on the Bee Colony Optimization (BCO), an innovative metaheuristic that evaluates at each time interval the optimal paths to a destination in order to minimize travel times. We have introduced the BCO to solve the high-level combinatorial problem, created when DTA considers all possible paths between O–D pairs. The network loading follows a discrete vehicle packet-based mesoscopic simulation approach, which explicitly considers a uniform acceleration of vehicle packets at each simulation time step. The performance of the quasi-anisotropic mesoscopic model has been tested simulating a dynamic loading process on a road network. QAMM solves the problem of dynamic network loading by taking into account analytical rules that express explicitly the link dynamics, flow conservation, flow propagation, and boundary conditions. Travel times are computed taking into account explicitly the acceleration of vehicle packets, thus validating the consistency of travel time propagation with the speed. In the presence of step-ups and step-downs of speed to a considerable extent, the adaptation of QAMM to shifts among flow patterns was found to be simultaneous.

An application to the Sioux-Falls network has been carried out to test the performance of our model in comparison to DYNASMART. Results of our model were found more realistic, possibly because the acceleration is explicitly calculated for each time interval, and thus the speed is not considered piecewise constant but calculated as a fixed-point value.

Such a performance of the model enables its utilization in a wide range of intelligent transport systems. For example, the reliable representation of traveling times enables the model to be calibrated for prediction processes within Advanced Traveller Information Systems (ATIS). The predictions on link traveling times can be obtained in terms of real-time traffic flow measurement inputs. These predictions can be basic inputs to real-time information dissemination applications, such as variable message signs for route guidance.

Acknowledgement

The Turkish author of this work was partially supported by The Scientific & Technological Research Council of Turkey (TUBITAK).

References

Anagnostopoulos, C.N., Anagnostopoulos, I.E., Loumos, V., Kayfas, E., 2006. A license plate recognition algorithm for intelligent transportation systems. IEEE Trans. Intell. Transp. Syst. 7 (3), 377–392.
Astarita, V., Florian, M., 2001. The use of mobile phones in traffic management and control. In: Proceedings of IEEE ITSC 2001. Oakland, CA, pp. 10–15.
Bekhor, S., Lotan, T., Gitelman, V., Morik, S., 2013. Free-flow travel speed analysis and monitoring at the national level using global positioning system measurements. J. Transp. Eng. 139 (12), 1235–1243.
Ben-Akiva, M.E., Bierlaire, M., Koutsopoulos, H.N., Mishalani, R.G., 1998. DynaMIT: a simulation-based system for traffic prediction. In: Proceedings of the DACCORD Short-Term Forecasting Workshop.
Beni, G., Wang, J., 1989. Swarm intelligence in cellular robotic systems, In: Proceedings of the NATO Advanced Workshop on Robots and Biological Systems. Tuscany, Italy, pp. 26–30.
Brockfeld, E., Lorkowski, S., Mieth, P., Wagner, P., 2007. Benefits and limits of recent floating car technology – an evaluation study. In: Proceedings of the 11th WCTR Conference. Berkeley, USA.
Burghout, W., Koutsopoulos, H., Andreasson, I., 2005. Hybrid mesoscopic–microscopic traffic simulation. Transp. Res. Rec. 1934, 218–225.
Cascetta, E., Cantarella, G.E., Di Gangi, M., 1991. Evaluation of control strategies through a doubly dynamic assignment model. Transp. Res. Rec., 1306
Celikoglu, H.B., 2007. A dynamic network loading process with explicit delay modelling. Transp. Res. Part C 15 (5), 279–299.
Celikoglu, H.B., Dell'Orco, M., 2007. Mesoscopic simulation of a dynamic link loading process. Transp. Res. Part C 15 (5), 329–344.
Celikoglu, H.B., Dell'Orco, M., 2008. A dynamic model for acceleration behaviour description in congested traffic. In: Proceedings of IEEE ITSC 2008. Beijing, pp. 986–991.
Celikoglu, H.B., Gedizlioglu, E., Dell'Orco, M., 2009. A node based modelling approach for the continuous dynamic network loading problem. IEEE Trans. Intell. Transp. Syst. 10 (1), 165–174.
Coifman, B., 2002. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. Transp. Res. Part A 36 (4), 351–364.
Cortes, C.E., Lavanya, R., Oh, J.S., Jayakrishnan, R., 2001. A general purpose methodology for link travel time estimation using multiple point detection of traffic. In: Institute of Transportation Studies, University of California, Irvine, Report no. UCI-ITS-TS-WP-01-5.
Corthout, R., Flötteröd, G., Viti, F., Tampère, C.M.J., 2012. Non-unique flows in macroscopic first-order intersection models. Transp. Res. Part B 46, 343–359.
Davidović, T., Selmić, M., Teodorović, D., 2009. Scheduling independent tasks: bee colony optimization approach. In: Proceedings of the 17th Mediterranean Conference on Control and Automation. Thessaloniki, Greece, pp. 1020–1025.
Dell'Orco, M., 2006. A dynamic network loading model for mesosimulation in transportation systems. Euro. J. Operat. Res. 175 (3), 1447–1454.
Edara, P., Selmić, M., Teodorović, D., 2008. Heuristic solution algorithms for a traffic sensor optimization problem. In: Proceedings of the INFORMS 2008. Washington, DC.
Friesz, T.L., Bernstein, D.E., Smith, T.E., Tobin, R.L., Wie, B.W., 1993. A variational inequality formulation of the dynamic network user equilibrium problem. Operat. Res. 41 (1), 179–191.
Greenshields, B.D., 1935. A study in highway capacity. In: Proceedings of Highway Research Board, vol. 14, pp. 468–477.
Haghani, A., Hamedi, M., Sadabadi, K.F., Young, S., Tarnoff, P., 2010. Freeway travel time ground truth data collection using Bluetooth sensors. Transp. Res. Rec. 2160, 60–68.
Hoogendoorn, S.P., 2000. Model-based multiclass travel time estimation. In: WCTRS (Ed.), World Conference on Transport Research. Seoul, Korea, pp. 1–20.
Jayakrishnan, R., Mahmassani, H.S., Hu, T.Y., 1994. An evaluation tool for advanced traffic information and management systems in urban networks. Transp. Res. C 3, 129–147.
Karaboğa, D., Akay, B., 2009. A survey: algorithms simulating bee swarm intelligence. Artif. Intell. Rev. 31, 61–85.
Leonard, D.R., Gower, P., Taylor, N.B., 1989. CONTRAM: structure of the model. In: TRRL Research Report 178. Crowtorne.
Lučić, P., Teodorović, D., 2001. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, pp. 441–445.
Marković, G.Z., Teodorović, D.B., Aćimović-Raspopović, V.S., 2007. Routing and wavelength assignment in all-optical networks based on the bee colony optimization. J. AI Commun. 20 (4), 273–285.
Merchant, D.K., Nemhauser, G.L., 1978. A model and an algorithm for the dynamic traffic assignment problems. Transp. Sci. 12 (3), 183–199.
Nagel, K., Schreckenberg, M., 1992. A cellular automaton model for freeway traffic. J. Phys. I 2, 2221–2229.
Nam, D.H., Drew, D.R., 1998. Analyzing freeway traffic under congestion: traffic dynamics approach. J. Transp. Eng. 124 (3), 208–212.
Oh, J., Jayakrishnan, R., Recker, W., 2003. Section travel time estimation from point detection data. In: Center for Traffic Simulation Studies, Paper VCI-ITS-TS-WP-02-15.
Pathirana, P.N., Savkin, A.V., Bulusu, N., Plunkett, T., 2006. Speed control and policing in a cellular mobile network: SpeedNet. Comp. Commun. 29 (17), 3633–3646.

Peeta, S., 1994. System Optimal Dynamic Traffic Assignment in Congested Networks with Advanced Information Systems, Ph.D. Dissertation. The University of Texas at Austin.
Peeta, S., Mahmassani, H.S., 1995. System optimal and user equilibrium time-dependent traffic assignment in congested networks. Ann. Operat. Res. 60, 81–113.
Rubio-Ardanaz, J.M., Wu, J.H., Florian, M., 2003. Two improved numerical algorithms for the dynamic network loading problems. Transp. Res. Part B 37 (2), 171–190.
Teodorović, D., 2008. Swarm intelligence systems for transportation engineering: principles and applications. Transp. Res. C 16, 651–782.
Teodorović, D., 2009. Bee colony optimization (BCO). In: Innovations in Swarm Intelligence. Springer-Verlag, Berlin, Heidelberg, pp. 39–60.
Teodorović, D., Dell'Orco, M., 2008. Mitigating traffic congestion: solving the ride-matching problem by bee colony optimization. Transp. Plan. Technol. 31 (2), 135–152.
Teodorović, D., Lučić, P., Marković, G., Dell'Orco, M., 2006. Bee colony optimization: principles and applications. In: 8th Seminar on Neural Network Applications in Electrical Engineering NEUREL 2006, pp. 151–156.
Treiber, M., Helbing, D., 2002. Reconstructing the spatio-temporal traffic dynamics from stationary detector data. Cooperat. Transp. Dynam. 1, 3.1–3.24.
Turner, S., Stockton, Wm.R., James, S. Rother, T., Walton, C.M., 1998. Its Benefits: Review of Evaluation Methods and Reported Benefits. Report 1790–1, Project Number 0–1790. Texas Transportation Institute, College Station, Texas.
Van Lint, J.W.C., Van der Zijpp, N.J., 2003. Improving a travel-time estimation algorithm by using dual loop detectors. Transp. Res. Rec. 1855, 41–48.
Van Lint, J.W.C., Hoogendoorn, S.P., van Zuylen, H.J., 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. Transp. Res. Part C 13 (5–6), 347–369.
Wang, Y., Zheng, Y., Xue, Y., 2014. Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 25–34.
Wie, B.W., Tobin, R.L., Friesz, T.L., Bernstein, D., 1995. A discrete time, nested cost operator approach to the dynamic network user equilibrium problem. Transp. Sci. 29 (1), 79–92.