REAL-TIME DYNAMIC O-D MATRIX ADJUSTMENT USING SIMULATED AND ACTUAL LINK FLOWS IN URBAN NETWORKS

by

Theodore Tsekeris

Department of Transportation Planning and Engineering, Faculty of Civil Engineering, National Technical University of Athens, 5 Iroon Polytechniou, 157 73 Athens, Greece.
Tel. No. (+3010) 7785610. E-mail Address: fabtse@central.ntua.gr

and

Antony Stathopoulos

Department of Transportation Planning and Engineering, Faculty of Civil Engineering, National Technical University of Athens, 5 Iroon Polytechniou, 157 73 Athens, Greece.
Tel. No. (+3010) 7721288. E-mail Address: astath@transport.ntua.gr

Word Count: 7450
ABSTRACT

The present paper investigates the efficiency and robustness of different real-time dynamic origin-destination (O-D) matrix adjustment algorithms when implemented in large-scale transportation networks. The proposed algorithms produce time-dependent O-D trip matrices based on the maximum-entropy trip departure times, as they are calculated with the use of simulated and actual observed link flows. The implementation of the algorithms, which are coupled with a quasi-dynamic traffic assignment (DTA) model, showed their convergent behavior and their potential for handling realistic urban-scale network problems, in terms of both accuracy and computational time. The algorithm performance was found to be affected by the assumptions underlying the structure of the prior matrix information and the quasi-DTA model, the time interval duration, the nature of observed link flows, the network scale and the link count availability. The relative efficiency of the algorithms was found to depend on the level at which the assigned flows approximate the observed link flows. These results may provide insights on the suitability of each algorithm for diverse application domains, including freeways, small networks and large-scale urban networks, where different quality of O-D information is usually available. Furthermore, other sources of traffic information, in addition to the link counts, could possibly enrich the evaluation of the performance of dynamic matrix adjustment procedures for the case of real-time urban networks.
1. INTRODUCTION

The origin-destination (O-D) demand matrix estimation is a long-standing problem in the areas of transportation planning and engineering, with applications ranging from urban transportation planning analysis (1) to real-time dynamic traffic estimation and prediction, traveler information provision and control operations (2, 3). In the context of dynamic transportation networks, the estimation of time-dependent O-D matrices involves a series of additional complexities, such as departure time uncertainty and fixed arrival rates. Such complexities are usually addressed through the use of some dynamic traffic assignment (DTA) model. On the other hand, the approaches which do not incorporate a traffic assignment component, usually referred to as non-DTA-based approaches, can basically be applied to simple networks such as interchanges/intersections and freeway segments, where the entry and exit flows cover major link flow information and are readily available (4, 5). The DTA-based approach can generally be regarded as more behaviorally sound and robust for addressing the various time-dependent aspects involved in the operation of large-scale transportation networks, in comparison to the non-DTA-based approach. Nonetheless, its performance can entail a significant computational burden (6).

The present paper describes and compares several algorithms for the time-dependent O-D matrix estimation problem using a quasi-DTA model. This model provides a plausible, in terms of mathematical complexity, as well as an efficient and tractable, in terms of computational cost, procedure for the calculation of the dynamic traffic loading conditions over the network. It is based on a simulation procedure developed in (7) and the PARCMAN project (8) for the time-dependent estimation of link-use proportions, as described in the following section. On the other hand, the time-dependent O-D matrix estimation process seeks a trip distribution that produces a link flow pattern sufficiently ‘close’ to a set of dynamic flow measurements at selected links over the network. These measurements are usually assumed to represent the dynamic user optimal (DUO) flow pattern. Given that in real network conditions there are infinitely many matrices satisfying the constraints imposed by the assignment procedure, the estimation problem can be typically stated as one seeking to find the maximum-entropy O-D trip distribution subject to constraints at the level of origins and destinations, the existence of a prior O-D matrix and the link traffic counts. Many methods have been proposed in the literature for the given problem, which is usually referred to as the matrix adjustment or balancing problem. However, a relatively little attention has been given on assessing their efficiency and robustness for on-line dynamic and realistic-scale urban network applications.

Lamond and Stewart (9) unified most of these methods by means of the theoretical properties of the balancing method of Bregman (10). Probably the most well-known of these methods is the multiproportional procedure (MPP) of Murchland (11), which was applied by Van Zuylen and Willumsen (12) for the O-D matrix estimation from traffic counts and was extended dynamically by Janson and Southworth (13) for the calculation of departure times from each origin zone. Another balancing method, which was also referenced in (9), is the multiplicative algebraic reconstruction technique (MART), developed by Gordon et al. (14). The convergence properties of MART were shown in (15). In general, the aforementioned balancing methods exhibit a theoretically slow convergence rate, although they perform well in practice (16, 17). In addition, Wu (18) provided a modified version of the latter method, known as R-MART, in order to enhance its numerical performance and convergence behavior, without providing any analytical proof on its convergence.

The present paper provides appropriate dynamic extensions to the MPP and MART algorithms, as well as a hybrid (doubly iterative) matrix adjustment procedure, to be referred here as DIMAP, which are coupled with the quasi-DTA model of PARCMAN in order to estimate dynamic trip departure rates and, subsequently, O-D matrices over a series of successive time intervals. Prompted by the need to emphasize on real-life large-scale urban networks, the above algorithms are tested on the central part of the Greater Athens Area network, using both simulated and actual traffic flow data obtained from a real-time traffic counting system. Furthermore, a sensitivity analysis is undertaken to evaluate the efficiency and robustness of the algorithms with respect to different assumptions underlying the loading conditions and the DTA procedure. As far as the organization of the paper is concerned, Section 2 describes the structure of the quasi-DTA model. Section 3 describes the various dynamic O-D matrix adjustment algorithms. Section 4 presents the simulation tests undertaken and the numerical results obtained by implementing the algorithms, together with some comments on their comparative performance. Section 5 includes the conclusions drawn from the study results.
2. THE QUASI-DTA PROCEDURE

The present section describes the quasi-DTA model to be coupled with the time-dependent O-D matrix estimation model. Several approaches have been proposed to formulate and solve the DTA problem. These can generally be categorized into two broad categories (see 6). First, the analytical DTA approaches, including models on the basis of mathematical programming, optimal control theory and variational inequalities. The second category includes the simulated-based dynamic assignment, wherein the vehicles are divided into a number of packets (e.g. ten vehicles in each packet) and then are loaded incrementally onto the paths of each O-D pair during the current time interval. The simulation-based approach allows the detailed investigation of behavioral aspects, such as those concerning the response of drivers to descriptive or prescriptive information conveyed to them in real time by traffic information systems (e.g. VMS panels) and it can handle large-scale network applications. The present simulation quasi-DTA model is based on the reactive or instantaneous link travel cost definition. This definition implies that each traveler departing from a particular location at any instant chooses the shortest path to his/her destination, based on the travel time perceived according to the currently prevailing traffic conditions and which is estimated at the time the traveler enters the path. Given that each instant may be approximated by a sufficiently short time interval (of, e.g. \( \tau = 15 \) minutes), the model, whose mathematical formulation is following, calculates at the beginning of each interval the minimum-cost path from each origin and, then, assigns each ‘packet’ of vehicles departing from the corresponding origin onto the ‘best’ route in order to reach the intended destination.

Let consider a network represented as a graph \( \Gamma = (D, M) \), with \( D \) and \( M \) being the set of nodes and directed links (arcs) respectively, with a total number of \( N \) O-D pairs. Let \( X \) be the O-D matrix whose elements \( x_{ij}^* \in X^* \) denote the number of vehicular trips between the \( n \)th O-D pair, departing from origin \( i \in I \) during time interval \( \tau \in T_d \) directed towards destination \( j \in Z \), where \( X^* \) is the set of feasible (i.e. positive) O-D trip flows and \( T_d \) is a wider time period including all intervals in which trip departures have occurred. The \( x_{ij}^* \) trips may use different network paths connecting O-D pair \( (i, j) \). Hence, trip demand \( x_{ij}^* \) will give rise to path flows \( h_{ip}^* \in H \), with path \( p \) being one of the paths belonging to the set \( P_{ij} \) of all feasible paths connecting the O-D pair \( (i, j) \). Let also consider \( M_o \subseteq M \) the total number of the observed links that contain a traffic counter. \( y_{im}^* \) the observed traffic count on the \( m \)th link and \( \hat{y}_{im} \) the assigned (estimated) link volume on that link during time interval \( \tau \in T \). This interval \( \tau \) refers to the interval in which the link flow \( y_{im}^* \) has been counted and the time period \( T \) is the given study period, including all intervals in which link flows have been counted. In the case of static or steady-state traffic assignment, the length of \( T \) is typically ranging from 1 hour to 3 hours. The intervals \( \tau \) and \( \tau \), and, hence, time periods \( T_d \) and \( T \), do not necessarily coincide as \( \tau \) may indicate an interval prior to \( \tau \).

Then, the dynamic user optimal (DUO) conditions, through our quasi-DTA model formulation, can be stated as follows:

\[
\min \mathcal{J} (\hat{y}) = \sum_{m \in M} \sum_{\tau \in T} \int_0^\tau c_{im}^\tau (s) ds
\]

subject to \( \hat{y} \in Y^* \)

The objective function \( \mathcal{J} (\hat{y}) \) represents the generalized travel cost of users that occurs at time intervals \( \tau \). The variable \( c_{im}^\tau \) corresponds to the interval-specific link cost on the \( m \)th link. The subset \( Y^* \) is defined as the space of feasible flows \( \hat{y} \) contained in (1) and for each \( \hat{y} \in Y^* \), the following relationships should be satisfied:

\[
\sum_{p \in P_{ij}} h_{ip}^* = x_{ij}^*, \quad \forall (i, j), \forall \tau \in T_d
\]
\[ h^\tau_p \geq 0, \quad \forall \ p \in P_{ij}, \forall (i, j) \quad (3) \]

\[
\sum_{p \in P_{ij}} \sum_{\tau \in T_d} \delta^{\tau}_{pm} h^\tau_p = \hat{y}^\tau_m, \quad \forall \ m \in M, \forall \ \tau \in T \quad (4)
\]

where
\[
\delta^{\tau}_{pm} = \begin{cases} 
1, & \text{if path } p \in P_{ij} \text{ uses link } m, \\
0, & \text{otherwise}, 
\end{cases} \quad \forall \ m \in M, \forall \ p \in P_{ij}, \forall \ \tau \in T_d, \forall \ \tau \in T \quad (5)
\]

The binary integers \( \delta^{\tau}_{pm} \) constitute the link–route incidence matrix \( \Delta \) and take values equal to 1 if the link \( m \) belongs to the path \( p \), and 0 otherwise. Let denote \( u^\tau_{pm} \) the inflow rate into and \( v^\tau_{pm} \) the exit rate from link \( m \) of path \( p \in P_{ij} \) during time interval \( \tau \). Then, the number of vehicles on a link \( m \) at any interval \( \tau \in T \) can be expressed as:
\[
\hat{y}^\tau_m = \sum_{p \in P_{ij}} (u^\tau_{pm} - v^\tau_{pm}) \quad \forall \ m \in M, \forall \ \tau \in T \quad (6)
\]

In addition, let \( R \) be the route choice matrix, with elements \( r^\tau_{p} \) being the fraction of trip demand between O-D pair \((i, j)\) that is assigned during interval \( \tau \) and uses path \( p \in P_{ij} \). Then, it holds that:
\[
\sum_{p \in P_{ij}} r^\tau_{p} = 1, \quad \forall (i, j), \forall \ \tau \in T_d \quad (7)
\]

and
\[
r^\tau_{p} \geq 0, \quad \forall \ p \in P_{ij}, \forall \ \tau \in T_d \quad (8)
\]

Then, the path flow variable can be expressed as product between the O-D matrix and the route choice matrix \( R \) variables as follows:
\[
h^\tau_p = \sum_{ij} \sum_{\tau \in T_d} x^\tau_{ij} r^\tau_{ij}, \quad \forall \ \tau \in T_d \quad (9)
\]

Combining equations (4) and (9), the following linear-form relationship between the link flow for each link \( m \in M_o \) and the corresponding O-D trip flows is obtained:
\[
\hat{y}^\tau_m = \sum_{ij} \sum_{\tau \in T_d} x^\tau_{ij} \sum_{p \in P_{ij}} \delta^{\tau}_{pm} r^\tau_{ij} = \sum_{ij} \sum_{\tau \in T_d} \alpha^{\tau}_{ijm} x^\tau_{ij} \quad \forall \ \tau \in T \quad (10)
\]

where \( \alpha^{\tau}_{ijm} \) denotes the relevant link-use proportion, also known as assignment proportion, which is an expression of the probability that a trip departing from origin \( i \) during time interval \( \tau \) directed towards destination \( j \) will use the \( m^{th} \) link during the time interval \( \tau \). These \( \alpha^{\tau}_{ijm} \) proportions constitute the elements of the \((N \times M_o)\) assignment matrix \( A \). The above formulation provides the O-D-specific definition of assignment proportions. Alternatively, the assignment proportions may be expressed in different forms, including the origin- or departure-specific link-use proportions and the destination- or arrival-specific link-use proportions. In the present quasi-DTA model, the origin-specific link-use proportions \( \alpha^{\tau}_{im} \) are estimated for each time interval \( \tau \). Hence, equation (10) can be reduced as follows:
\[ \hat{y}_m^\tau = \sum_{i \in I} x_i^\tau \sum_{p \in p} \delta_{\tau_{pi}}^\tau \cdot y_{pi}^\tau = \sum_{i \in I} \sum_{\tau_{pi} \in T_p} \alpha_{\tau_{pi}}^\tau \cdot x_i^\tau, \]  

where \( \alpha_{\tau_{pi}}^\tau \) are the elements of the reduced \((I \times M)\) assignment matrix \(A\). The \( \alpha_{\tau_{pi}}^\tau \) proportions denote the probability that a trip departing from origin \(i \in I\) during time interval \(\tau\) will use the \(m^{th}\) link during the (observation) time interval \(\tau\). Similarly, \(x_i^\tau\) denote the number of vehicular trips departing from origin \(i\) during time interval \(\tau\). A suitable normalization technique for transforming the trip departures \(x_i^\tau\) to the corresponding O-D trip flows \(x_{mj}^\tau\) at each departure time interval \(\tau\) is presented in the next section.

Also, by considering the costs \(c_{mj}^\tau\) along the \(m\) links forming a path \(p\) as completely additive, one can set the path travel cost \(C_p^\tau\) equal to the sum of all costs \(c_{mj}^\tau\). Hence, for each origin-specific path travel cost, it should hold that:

\[ C_p^\tau = \sum_{m \in M} \delta_{\tau_{mj}}^\tau \cdot c_{mj}^\tau, \quad \forall \ p \in P_j, \forall \ \tau_{pi} \in T_d, \forall \ \tau \in T \]  

The flow circulation over the network is governed by a strictly positive, flow-dependent delay function \(D_{mj}\) corresponding to each link \(m\). This function actually relates the exit time \(t_m(\tau)\) from link \(m\) to the time of entry \(t(\tau)\) during a specific time interval \(\tau\) as follows:

\[ t_m(\tau) = t(\tau) + D_{mj}(y_m^\tau) \]  

The formulation of \(D_{mj}\) function is dependent on the level of queuing and the capacity \(y_m^\tau\) of each link \(m\). The link travel cost \(c_{mj}^\tau\) is expressed by the function of travel time \(t_m(\tau)\), whose specification is based on the corresponding travel time functions of SATURN (19) as follows:

If \(\hat{y}_m^\tau \leq y_m^\tau\), then

\[ t_m = t_o + \left(\frac{\hat{y}_m^\tau}{y_m^\tau}\right)^\mu (t_c - t), \quad \forall \ \tau \in T \]  

If \(\hat{y}_m^\tau > y_m^\tau\), then

\[ t_m = t_o + \left(\frac{\hat{y}_m^\tau}{y_m^\tau}\right)^\mu (t_c - t) + \frac{1}{2} \left(\frac{\hat{y}_m^\tau}{y_m^\tau} - 1\right) \tau, \quad \forall \ \tau \in T \]  

where \(t_o\) denotes the travel time in free traffic flow conditions, \(t_c\) is the travel time at capacity and \(\mu\) is a calibrated exponent factor. The second terms of the sums in (14) and (15) represent the effect of congestion on the travel time along link \(m\). The third term in (15) represents the additional (queuing) delay time in case where the assigned traffic flow \(\hat{y}_m^\tau\) exceeds the capacity bound \(y_m^\tau\). These last terms comprise the value of the time-dependent delay function \(D_{mj}\).

The definition of initial network loading conditions requires an equilibrated network loading process to estimate the link-use proportions for the first observation time interval \(\tau_0\). This process was provided here through implementing a quantal loading assignment. This method has been found to be quite efficient for the case of large-scale network (20). The O-D matrix used for the quantal loading of the network was assumed to be equal to a prior O-D matrix estimate corresponding at interval \(\tau_0\) (see next section). For the calculation of the minimum-cost path between each O-D pair \((i, j)\) at each departure time interval \(\tau_j\), a time-dependent minimum-cost tree building process was employed based on D’Esopo’s algorithm (21). The purpose of using the present quasi-DTA model is to produce link volumes \(\hat{y}_m^\tau\) that are sufficiently close to the observed ones \(y_m^\tau\). Hence, a series of successive adjustment iterations (assignment mappings) between the set of path and link flows, \(h_{pj}^\tau\) and \(\hat{y}_m^\tau\)
respectively, and the link-use proportions $\alpha^\tau_{im}$ at each specific time interval $\tau$ may be required (e.g. 10 iterations per interval). This iterative process aims to correct (or best ‘fit’) the flow-dependent $\alpha^\tau_{im}$ values according to the DUO-defined path and link travel costs, $C^v_{\tau}$ and $c^v_{\tau}$ respectively.

The present simulation-based model, though no rigorous proof on its convergence properties exists, can provide an efficient tool for the development of a computationally tractable and mathematically plausible framework for the quasi-dynamic mapping of the O-D demand to the link flow pattern and the calculation of time-dependent trip departure rates. It does not explicitly represent several microscopic features of urban traffic, like the structural evolution of queuing and spillback phenomena, albeit it enables the consideration of deterministic control delays and variable travel time effects. Due to its intrinsic simplicity, the proposed model constitutes a framework which is realistic from the software point of view: it enables the simulation of realistically large-scale networks (i.e. spanning several thousands of links servicing several hundred thousands of travelers) using simple PC facilities (typically with 128 MB of RAM) in a period of a few seconds. Furthermore, it requires a relatively low level of data requirements. It basically uses the same data as a standard static assignment model, whilst it enables the updating of link-use proportions endogenously, with the single requirement of on-line storing the link volumes and travel costs of the previous time interval $\tau-1$.

3. THE DYNAMIC O-D MATRIX ADJUSTMENT PROBLEM

3.1 Problem formulation

Based on Willumsen (22), the generalized entropy maximization formulation of the time-dependent O-D matrix estimation problem, in terms of estimating the statistically optimal (most probable) trip departure time distribution can be given as follows:

$$\text{minimize} \quad \sum_{i \in I} \sum_{\tau \in T} \alpha^\tau_{i\tau} \left[ \log \left( \frac{x^\tau_{i\tau}}{\hat{x}^\tau_{i\tau}} \right) - 1 \right], \quad \forall \ \tau \in T_d$$

subject to

$$y^\tau_m = \sum_{i \in I} \sum_{\tau \in T} \alpha^\tau_{i\tau} x^\tau_{i\tau}, \quad \forall \ m \in M_o, \ \forall \ \tau \in T$$

and

$$x^\tau_{i\tau} \geq 0, \quad \forall \ i \in I, \ \forall \ \tau \in T_d$$

where $\hat{x}^\tau_{i\tau}$ correspond to the row (origin) sums of the $(I \times Z)$ matrix $\hat{x} = \{\hat{x}^\tau_{i\tau}\}$. This matrix, known as prior O-D matrix, may be a preliminary estimate based on e.g. partial surveys, or a time-of-day historical matrix corresponding to the departure time interval $\tau$, or even an O-D matrix estimate derived from previous departure intervals. The factor $\lambda^\tau_{i\tau}$ denotes relevant weights that allow for some prior O-D matrix elements to be assumed more reliable than others. For the case of real-time traffic networks, the reliability of prior information on O-D pattern and trip departure rates is unknown in practice, particularly since there is no clear process available for accounting the differences in knowledge or confidence between prior O-D matrix elements. Thus, the $\lambda^\tau_{i\tau}$ values can be reasonably assumed as equal (to unity) for all O-D pairs.

In the present study, the use of time-of-day historical estimates is made (see 4.1). In this way, the different algorithms are allowed to be evaluated and compared given the same initial demand pattern. On the other hand, several recursive algorithms could be theoretically implemented in order to use demand estimates derived from previous departure intervals for the estimation of dynamic O-D matrices. These approaches, principally based on Bayesian updating methods (5, 23), showed experimentally to produce a considerable computational load when applied to the given study area. This is particularly due to the large number of (thousands) O-D pairs, the spatial distribution of the network and the high levels of congestion. These factors essentially render the on-line application of such approaches in real-world urban network problems with so many dimensions, as the current one, practically impossible.

The present dynamic O-D matrix balancing problem, as stated above, can be considered as one seeking those trip departure rates at each interval $\tau_{ij}$ being as ‘close’ as possible to the prior rates $\hat{x}^\tau_{i\tau}$.
while satisfying the quasi-DTA problem constraints. The normalization technique used to transform the trip departures $x'_{ij}$ into the equivalent O-D trip flows $x''_{ij}$ is based on the assumption of fixed arrival rates from each origin, according to the prior O-D demand pattern. Thus, the balanced or adjusted O-D matrices will be given as follows (13):

$$x''_{ij} = x'(\frac{\sum z'_{ij}}{\sum_{\tau \in T} z'_{ij}}), \quad \forall \ i \in I, \ \forall \ j \in Z, \ \forall \ \tau \in T_d$$

(19)

The transformation process above does not constrain the total trip departures over the whole time period $T_d$. Hence, the resulting demand is elastic, in the sense that a smaller or larger percentage of the trips originally departing from the origin zones at each interval $\tau_i$ may be estimated in order to satisfy the constraints of the quasi-DTA problem.

### 3.2 The multi-proportional procedure (MPP)

The MPP seeks to optimize a set of balancing factors $\beta^*_m$ corresponding to the flow constraint at each observed link $m \in M_o$ and time interval $\tau \in T$. According to the traditional practice of resolving the MPP (e.g., see 24), the link-use proportions $\alpha^*_{m\tau}$ are held constant throughout the iteration loops of the O-D matrix adjustment process during each interval $\tau$. In the present study, these loops for which $\alpha^*_{m\tau}$ values are held constant are referred to as inner-loop iterations $\ell_{IN}$. On the contrary, the suggested approach periodically updates the $\alpha^*_{m\tau}$ values after a fixed number of inner-loop iterations $\ell_{IN}$. The total number of these updates is represented here through the outer-loop iterations $\ell_{OUT}$. In this way, the solution procedure takes endogenously into account the fact that users may reconsider their route choice decisions (by re-calculating the minimum-cost path) from each origin $i \in I$ at each departure interval $\tau_i$ based on the prevailing congestion conditions at interval $\tau$. The modified dynamic version of the MPP, whose solution is based on the optimality conditions of the Lagrange multiplier method, can be described into the following steps:

**Step 1:** Assign the $\hat{x}$ matrix for departure interval $\tau_i$ to calculate the $\alpha^*_{m\tau}$ proportions. Set the $\beta^*_m$ values equal to unity for each $m \in M_o$ and $\tau \in T$. Set the number of outer-loop iterations $\ell_{OUT} = 0$.

**Step 2:** Set the number of inner-loop iterations $\ell_{IN} = 0$.

**Step 3:** Increment the number of inner-loop iterations $\ell_{IN} = \ell_{IN} + 1$.

**Step 4:** Select a new combination $(m, \tau)$ of observed link and time interval and calculate the link volumes as follows:

$$\hat{y}_m(\ell_{IN}) = \sum_{i \in I} \sum_{\tau_i \in T_d} \alpha^*_{m\tau} \cdot \bar{x}_i \cdot \prod_{m' \tau'} \left[\beta^*_m(\ell_{IN}) \phi^*_m(\ell_{IN+1})\right],$$

(20)

**Step 5:** Update the $\beta^*_m$ values as follows:

$$\beta^*_m(\ell_{IN+1}) = \beta^*_m(\ell_{IN}) \phi^*_m,$$

(21)

where $\phi^*_m$ is an adjustment factor obtained for the given combination $(m, \tau)$ by solving the following equation using a modified Newton-Raphson method:
\[ y_{ij}^{\ell} (\ell_{IN}) = \sum_{l=1}^{\infty} \sum_{\tau_{d} \in \mathcal{T}} \alpha_{m}^{\omega_{d} \tau_{d}} x_{ij}^{\ell_{IN}} \prod_{m} \left( \beta_{m}^{\omega_{d} \tau_{d}} (\ell_{IN}) \right)^{s_{ij}^{\omega_{d} \tau_{d}}} (\phi_{m}^{\omega_{d} \tau_{d}})^{s_{ij}^{\omega_{d} \tau_{d}}} , \quad (22) \]

**Step 6:** Update the trip departure flows as follows:

\[ x_{ij}^{\omega_{d} \tau_{d}} = \hat{x}_{ij}^{\omega_{d} \tau_{d}} \prod_{m \in T} \left( \beta_{m}^{\omega_{d} \tau_{d}} (\ell_{IN} + 1) \right)^{s_{ij}^{\omega_{d} \tau_{d}}} \quad (23) \]

Subsequently, estimate the adjusted O-D matrix based on (19).

**Step 7:** If not all \((m, \tau)\) combinations have been processed, then go to **Step 4**. Else go to **Step 8**.

**Step 8:** If the distance between the estimated and observed link volumes for each \((m, \tau)\) combination is greater than a pre-specified minimum value \(\delta\), or a maximum number of inner-loop iterations \(\max \ell_{IN}\) has not been reached, then return to **Step 3**. Else continue to **Step 9**.

**Step 9:** Increment the number of outer-loop iterations \(\ell_{OUT} = \ell_{OUT} + 1\). If a maximum number of outer-loop iterations \(\max \ell_{OUT}\) has been reached, then STOP. Else, assign the updated O-D matrix to obtain a new set of \(\alpha_{m}^{\omega_{d} \tau_{d}}\) proportions and go to **Step 2**.

### 3.3 The multiplicative algebraic reconstruction technique (MART)

The MART algorithm provides a convergent, generalized iterative matrix scaling procedure for the recursive adjustment (reconstruction) of the prior O-D trip flows to each of the constraints imposed by the set of link counts. Its dynamic extension, in terms of estimating time-dependent trip departures, can be described as follows:

**Step 1:** Set the number of iterations \(\ell = 1\). Initialize \(x_{ij}^{\omega_{d} \tau_{d}} (\ell) = \hat{x}_{ij}^{\omega_{d} \tau_{d}} , \forall i \in I , \forall \tau_{d} \in T_{d} \).

**Step 2:** Increment the number of iterations \(\ell = \ell + 1\).

**Step 3:** Update the trip departure flows as follows:

\[ x_{ij}^{\omega_{d} \tau_{d}} (\ell+1) = \prod_{m} \left( \sum_{l=1}^{\infty} \sum_{\tau_{d} \in \mathcal{T}_{d}} \alpha_{m}^{\omega_{d} \tau_{d}} x_{ij}^{\ell_{IN}} \prod_{m} \left( \beta_{m}^{\omega_{d} \tau_{d}} (\ell_{IN}) \right)^{s_{ij}^{\omega_{d} \tau_{d}}} (\phi_{m}^{\omega_{d} \tau_{d}})^{s_{ij}^{\omega_{d} \tau_{d}}} \right) \quad (24) \]

where \(s_{ij}^{\omega_{d} \tau_{d}}\) is a normalizing exponent factor introduced by Lent (1976) to enhance the numerical stability of the algorithm convergence. This is given as follows:

\[ s_{ij}^{\omega_{d} \tau_{d}} = \frac{1}{\sum_{m \in M_{d}} \alpha_{m}^{\omega_{d} \tau_{d}}} , \forall i \in I , \forall \tau_{d} \in T_{d} , \forall \tau \in T . \quad (25) \]

**Step 4:** Estimate the updated O-D matrix based on (19) and the corresponding new set of link volumes based on (11).

**Step 5:** If the distance between the observed and estimated link volumes is greater than a pre-specified minimum value \(\delta\), or a maximum number of iterations \(\max \ell\) has not been reached, then return to **Step 2**. Else STOP; the matrix with elements \(x_{ij}^{\omega_{d} \tau_{d}} (\ell)\) is the final adjusted O-D matrix.
The MART algorithm has a theoretically slow convergence behavior due to the orthogonal step directions established every two successive iterations. In contrast, the RMART algorithm (18) provides a diagonal search between two successive iterations to improve its convergence speed. The modified form of RMART to estimate time-dependent O-D matrices, in terms of trip departure rates, can be described into the following steps:

**Step 1, Step 2, Step 3 and Step 4** as in MART above.

**Step 5:** If the distance between the observed and estimated link volumes is greater than a pre-specified minimum value $\delta$, or a maximum number of iterations $\max \ell$ has not been reached, then proceed to **Step 6**. Else STOP; the matrix with elements $x_{ij}^\ell$ is the final adjusted O-D matrix.

**Step 6:** Perform another MART iteration to estimate a new set of trip departures $z_{ij}^\ell$ based on $x_{ij}^\ell (\ell+1)$.

**Step 7:** Perform a diagonal search as follows:

$$x_{ij}^\ell (\ell+2) = z_{ij}^\ell + b \left[ z_{ij}^\ell - x_{ij}^\ell (\ell+1) \right], \quad \forall \ i \in I, \ \forall \ \tau_d \in T_d$$

where

$$b = \max \{0, \min\{b_1, b_2\}\}$$

$$b_1 = \min \left\{ b \left[ \frac{z_{ij}^\ell}{x_{ij}^\ell (\ell+1)-z_{ij}^\ell} \right] x_{ij}^\ell (\ell+1)-z_{ij}^\ell \right\} > 0, \quad \forall \ i \in I, \ \forall \ \tau_d \in T_d$$

If

$$\sum_{i \in I} \sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d} z_{ij}^\ell - \sum_{i \in I} \sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d} x_{ij}^\ell (\ell) \neq 0,$$

then

$$b_2 = \frac{\sum_{i \in I} \sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d} z_{ij}^\ell - \sum_{i \in I} \sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d} x_{ij}^\ell (\ell)}{\sum_{i \in I} \sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d} z_{ij}^\ell - \sum_{i \in I} \sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d} x_{ij}^\ell (\ell)}, \quad \forall \ \mu$$

where $\mu$ is such that:

$$\max \left\{ y_m^\ell - \sum_{i \in I} \sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d} z_{ij}^\ell, \quad \forall \ \mu \right\}$$

Else, $b_2 = 1$;

and return to **Step 2**.

### 3.4 A doubly iterative matrix adjustment procedure (DIMAP)

Another possible dynamic extension of the aforementioned algorithms is the suitable combination of them in order to improve their performance characteristics. Such a procedure is referred here as *doubly iterative matrix adjustment procedure* or *DIMAP*, in the sense that the multi-proportionally adjusted matrix in each iteration is one that has already been corrected (reconstructed) on the basis of MART procedure. The DIMAP aims to further enhance the consistency between the trip departure rates from each origin zone and the observed link flows. In addition, the doubly iterative nature of the algorithm results in the reduction of the required amount of total iterations, with respect to MPP and MART. The algorithm convergence can also be ensured given that, as mentioned earlier, both MPP and MART have a proven convergent behavior. The steps of the DIMAP can be described as follows:
Step I. Set the number of MART iterations $\ell_1 = 1$. Initialize $x_{ij}^{\tau d}(\ell) = \hat{x}_{ij}^{\tau d}$, $\forall i \in I$, $\forall \tau_d \in T_d$. Increment the number of iterations $\ell_1 = \ell_1 + 1$.

Step II. Implement Step 3 described in 3.3 to perform a MART iteration and update the trip departure rates.

Step III. Set the number of inner-loop iterations for the MPP equal to $\ell_2 = 0$.

Step IV. Perform an inner-loop iteration to multi-proportionally adjust the updated trip departure rates following Step 2 - Step 7 as described in 3.2.

Step V. If the distance between the estimated and observed link volumes for each $(m, \tau)$ combination is greater than a pre-specified minimum value $\delta$, or a maximum number of inner-loop iterations $\text{max} \ell_2$ has not been reached, then return to Step IV. Else continue to Step VI.

Step VI. Increment the number of MART iterations $\ell_1 = \ell_1 + 1$. If a maximum number of iterations $\text{max} \ell_1$ has been reached, then STOP. Else, return to Step II.

4. NUMERICAL TESTS AND RESULTS

4.1 Presentation of the simulation experiments

All time-dependent O-D matrix estimation algorithms were implemented in FORTRAN and run on a Pentium III processor with 128 MB RAM. They were tested in a real urban-scale road network, that of the central part of the Greater Athens Area network. This network was modeled using 151 assignment nodes, 256 assignment links and a total number of 1936 O-D pairs, i.e. the assigned O-D matrix is $4444 \times 44$. Figure 1 provides the graphical representation of the given network. In the first set of simulation tests (see 4.2), the time-dependent O-D matrix estimation is undertaken using simulated link flows for about 7% of the total network links. Namely, the simulated flows on the selected links resulting from DUO-assigning the prior O-D flows $\hat{x}_{ij}^\tau$ corresponding at each interval $\tau_j$ are assumed as the ground-truth link counts. Different assumptions were made on the variability and time partition of the prior O-D matrix. This matrix is based on extensive O-D travel surveys in the Greater Athens Area and refers to the morning peak hour period, including a total number of 17340 vehicles per hour to be loaded onto the network.

In the second set of simulation tests (see 4.3), the estimation is carried out using real traffic count data for the first Tuesday of February 2000. These data are automatically collected at 22 key locations of the network and stored at the end of every 90-seconds cycle. The real-time counting system identifies and excludes data from malfunctioning detectors, whilst it uses smoothed flow values to avoid short-term local fluctuations, attributed to significant random and non-recurring traffic episodes. In this case, the simulation period spans the whole morning period (i.e. 6:00 am – 9:00 am), which is partitioned into 12 equal intervals of 15-minutes length, while different assumptions were also made on the distribution of the prior O-D matrix.

4.2 Estimation using simulated link flows

In the present paper, the percentage or relative root mean square error ($\text{RRMSE}_{\text{LINK}}$) was used to measure the distance between the estimated and observed link flows in order to control the convergence of the algorithms. The $\text{RRMSE}_{\text{LINK}}$ measure is given as follows:
\[ RRMSE_{\text{LINK}} = \sqrt{MSE_{\text{LINK}}} \left( \sum_{\text{mm} \in M_o} \frac{y_{\text{m}}^i - y_{\text{m}}^i}{M_o} \right)^{-1}, \quad \forall \tau \in T \]  

(30)

where \( MSE_{\text{LINK}} \) is the corresponding mean square error measure, given as follows:

\[ MSE_{\text{LINK}} = \sum_{\text{mm} \in M_o} \frac{(y_{\text{m}}^i - y_{\text{m}}^i)^2}{M_o}, \quad \forall \tau \in T \]  

(31)

Similarly, the performance of the various algorithms was evaluated on the basis of the corresponding O-D trip flow measures, as follows:

\[ RRMSE_{\text{OD}} = \sqrt{MSE_{\text{OD}}} \left( \sum_{\text{ij} \in N^+} \frac{x_{\text{i}j}^\tau - \hat{x}_{\text{i}j}^\tau}{N^+} \right)^{-1}, \quad \forall \tau \in T_d \]  

(32)

and

\[ MSE_{\text{OD}} = \sum_{\text{ij} \in N^+} \frac{(x_{\text{i}j}^\tau - \hat{x}_{\text{i}j}^\tau)^2}{N^+}, \quad \forall \tau \in T_d \]  

(33)

where \( N^+ \) is the number of feasible (i.e., positive) O-D trip flows. First, the various algorithms were tested for the basic case of uniform (symmetric) distribution of the prior O-D demand across 4 time intervals of 15 minutes length each. The initially assigned O-D matrix was constructed to deviate at a level of 10% from the corresponding elements of the prior O-D matrix.

Figure 2 presents the results obtained from the implementation of the MPP. As it is shown in Figure 2(a), the MPP converges rapidly within the first four inner-loop iterations \( \ell_{\text{IN}} \), in case where the number of outer-loop iterations is set equal to \( \ell_{\text{OUT}} = 1 \). The \( RRMSE_{\text{LINK}} \) value falls below 5% for each interval \( \tau \) after \( \ell_{\text{OUT}} = 13 \) (see Figure 2(b)), using a number of \( \ell_{\text{IN}} = 10 \) iterations for each outer-loop iteration, at about 75 seconds of CPU time. The MPP converges at a stationary point for each \( \tau \) after \( \ell_{\text{OUT}} = 22 \) (not shown in Figure 2) at about 320 seconds, without reaching the strict convergence criterion of the minimum distance value \( \delta = 1\% \). For the results of the MPP, the average level of non-convergence (LNC) for each interval \( \tau \), which is defined as follows:

\[ LNC (\%) = 100 \times \frac{\delta - RRMSE_{\text{LINK}}}{\delta} \]  

(34)

was found equal to \( LNC = 180 \% \). On the contrary, all the other algorithms of the present study, for the case of using simulated observed flows, were found to converge or, in other words, they reached a level of non-convergence equal to \( LNC = 0 \). Figure 2(c) indicates that the reduction of \( RRMSE_{\text{LINK}} \) across the outer-loop iterations is associated with an increase of \( RRMSE_{\text{OD}} \) for each interval \( \tau \), as it is calculated in (32), while the same behavior was observed for \( MSE_{\text{OD}} \), based on (33). Similar trends between the O-D trip and link flow performance measures across iterations were also observed in the other algorithms. This finding essentially indicates a trade-off between optimal link flow pattern and trip departure rates. Moreover, all algorithms showed a linear convergence behavior, except of the DIMAP case, which performed small disturbances decreasing with the number of iterations.

As mentioned above, in contrast to the MPP, the other algorithms satisfied the criterion of minimum distance \( \delta = 1\% \) for each interval \( \tau \), while they required a much less computational time for convergence, using the same prior trip distribution assumptions as above (see case (a) in Table 1).

The significantly greater computational time associated with the MPP may be attributed to the use of
the modified Newton-Raphson method as well as the need to periodically update the link-use proportions. As it can be derived from the results shown in Table 1, the DIMAP performed most efficiently, in terms of the $MSE_{OD}$ and $RRMSE_{OD}$, and displayed a robust behavior, with respect to both MART and RMART, using a range of different assumptions. These assumptions refer to different (lower and higher) levels of variability with respect to the prior trip distribution (see case (a) and case (b) respectively), different directions of change in the prior demand conditions (see case (c) and case (d)), the size of O-D trip flows (case (e)), different (shorter and longer) time interval durations (see case (f) and case (g) respectively), and the variability of observed link flows from the DUO link flow pattern (case (h)). The DIMAP also requires a considerably less number of iterations to convergence, in comparison to MART, although it is associated with higher CPU times.

In all algorithms, the increase in demand variability and the size of O-D trip flows appear to adversely affect the reliability of O-D matrix estimates as well as the CPU times required for convergence. This outcome may be interpreted by the fact that trip makers, in case of volatile and growing demand conditions, cannot easily determine which paths correspond to lower travel costs, so they finally select a less optimal travel path. Similar are the effects of the variability of the (simulated) observed flows from the DUO link flow pattern. In contrast, the skewed (non-symmetric) loading of the assigned O-D matrix appears to enhance the performance of the tested algorithms. Nevertheless, the direction of change in demand conditions does significantly influence their convergence behavior. Furthermore, shorter time interval durations (10-minutes long) showed an increase in the reliability of O-D matrix estimates, in spite of the increase in required CPU times due to the greater number of the quasi-DTA runs, in comparison to the longer interval durations (30-minutes long). This improvement may be associated with a faster adjustment of users’ decisions, related to the departure time and route choice, in regard to the prevailing congestion conditions. Thus, possible changes in the demand structure should preferably be considered to obtain a deeper understanding on the efficiency and robustness of different dynamic O-D matrix adjustment algorithms. The next subsection focuses on the influence of considering observed flows of a different nature, i.e. actual link counts.

4.3 Estimation using real traffic counts

In the case of using real traffic count data, the algorithm convergence is far from guaranteed, mainly due to the intrinsic complexities of the DTA procedure as well as the departure of real-life link flows from the DUO pattern. The DTA should be considered as a prior assumption, which cannot be actually observed in practice. Hence, a maximum number of 200 iterations was specified as the termination criterion for each algorithm. In all cases, the algorithms showed to asymptotically converge at a stationary $RRMSE_{LINK}$ value. This value was basically considered to evaluate the performance of each algorithm. In addition, a number of iterative adjustments of the link-use proportions to the DUO-defined path and link travel costs was considered by carrying out 10 successive quasi-DTA runs per interval. These re-assignments may address errors in the estimation of link-use proportions, which can be attributed to a variety of factors, including path choice modeling, network representation and performance functions (25). Also, two different loading conditions were considered to encounter possible effects of bias imposed by the prior matrix estimates. First, the network was loaded using a prior O-D matrix of moderate size (about 50000 vehicles per hour). Next, the loading was performed assuming a 50 % growth in total demand.

All cases showed a significant reduction of the initial $RRMSE_{LINK}$ value (which was calculated by assigning the prior O-D matrix estimate), ranging on average between 40% and 60% for all intervals. As it is shown in Table 2(a) and Table 2(b), the greatest improvements were attained by the consecutive dynamic re-assignments of the trip matrix, particularly for the case of increased demand, in expense of the computational time. This outcome denotes the high sensitivity of the dynamic O-D matrix adjustment to the accuracy of link-use proportions. This sensitivity appears to increase with the growth in the size of O-D trip flows. The RMART algorithm provided the best results, in terms of the $RRMSE_{LINK}$ improvement, with the MART algorithm and the DIMAP to follow in sequence. As regards the computational effort, the MART algorithm displayed the best behavior, while the DIMAP required the highest CPU times. Thus, the use of actual time-dependent link counts may lead to different inferences about the relative efficiency of real-time matrix adjustment algorithms, with respect to the results typically obtained on the basis of simulated observed link flows. The implications of this finding are discussed in the following section.

A deeper understanding of the relative performance of the time-dependent O-D matrix estimation algorithms can be obtained by calculating a suitable statistical measure of the differences of the
dynamic trip departure rates produced in each case. This measure can be derived by the Relative Average Error of departures, as follows:

\[ RAE_d = \frac{\sum_{\tau \in T_d} (x_{\tau} - \hat{x}_{\tau})}{\sum_{\tau \in T_d} \hat{x}_{\tau}}, \quad \forall \tau \in T_d \] (35)

In addition, the sensitivity of the efficiency and robustness of the algorithms was examined with respect to the network scale and the link count availability. In the former case, the scale of the network area was increased to include 168 assignment nodes and 300 assignment links, while, in the latter case, the number of counted links was reduced by 30%. In all cases, the quasi-DTA model was used carrying out 10 successive assignment runs per interval. As it is shown in Table 3(a) and Table 3(b), the produced trip departure rates demonstrate an overall increase, particularly for the moderately congested conditions. This increase can be attributed to the trip understatement underlying the survey-based O-D estimates as well as the traffic growth in the study area. The RMART algorithm was found to better capture the suppressed effects of congestion, in the sense of producing a smaller increase in departure rates, in comparison to the MART and DIMAP. On the other hand, in the experiments assuming an increased network scale and reduced link count availability, referred to as case (ii) and case (iii) respectively, the differences in the departure rates resulted from the moderate and increased O-D trip sizes are diminished, in comparison to the initial case (i). This outcome can be explained by the greater spread of trip flows in the expanded network, and the fact that the reduced link flow information cannot fully capture the congestion effects.

As it is shown in Table 3(c) and Table 3(d), RMART produces the lowest \( RRMSE_{\text{LINK}} \) values in the case (ii), in comparison to the MART and DIMAP. The higher \( RRMSE_{\text{LINK}} \) values, in comparison to the case (i), indicate that the increased network scale is associated with a greater uncertainty when calculating the link use proportions from each origin. On the contrary, the MART and DIMAP algorithms produced lower \( RRMSE_{\text{LINK}} \) values for the case (iii), in comparison to the RMART as well as the corresponding values in the initial case (i). The small relative changes in \( RRMSE_{\text{LINK}} \) between case (i) and case (iii) suggest that the omitting link counts probably do not carry such significant information in order to considerably affect the resultant link flow solution. Finally, the increase in network scale, in case (ii), is accompanied with a slight increase of the CPU time, due to the additional effort required by the quasi-DTA model in route processing. On the contrary, the reduction of the link count availability, in case (iii), results in a decrease of the CPU time, because of the reduced dimension of the assignment matrix.

5. CONCLUSIONS

In the present paper, several quasi-DTA-based approaches were developed and tested for the estimation of the maximum-entropy trip departure times from each origin zone and the production of the most probable time-dependent O-D matrix. In general, the performance of the proposed algorithms appeared to be promising for the case of large-scale urban networks, while potential for further improvements was also recognized. All algorithms displayed a convergent behavior at an asymptotical stationary value and showed significant superiority over the link flow pattern produced by the prior matrix estimate. The algorithm performance was found to be affected by the assumptions underlying the structure of the prior O-D matrix, the time interval duration, the nature of observed link flows, the DTA procedure, the network scale and the link count availability. In particular, the relative performance of the various algorithms for the cases of using simulated and actual observed flows suggests that the fit to link counts only cannot provide a sufficient indication to evaluate the effectiveness of different matrix adjustment procedures on the application of real-time transportation management and control operations.

For the case of simulated link flows, wherein a satisfactory convergence level can be achieved, the DIMAP resulted in the most reliable matrix and performed with greater robustness than the other algorithms. Thus, it may provide a satisfactory alternative for the assessment of users’ responses and network performance under a set of different traffic management strategies, when the assigned flows do sufficiently approximate the observed link flows. Such cases may be practically recognized in freeways segments or in small and simplified networks. On the other hand, the MART algorithm and,
particularly, its heuristic improvement, i.e. the RMART algorithm, show a better potential to solve the real-time matrix adjustment problem in the case of using actual observed flows in a real urban-scale network. The implementation of other available, dynamically extended balancing algorithms, or possible improvements of them, would provide more insights on resolving the problem of optimal trip departure times and time-dependent O-D matrix estimation. These efforts could also benefit from using as testbed a greater variety of real networks with different characteristics as well as from the advent of methodologies enabling the combination of link counts with other real-time obtainable traffic measures, including traffic densities and speeds.

REFERENCES


LIST OF TABLES AND FIGURES

TABLES:

TABLE 1  Sensitivity analysis of algorithm performances for the case of simulated link flows:

Case (a): 10% variation of the assigned O-D flows from the prior O-D matrix
Case (b): 20% variation of the assigned O-D flows from the prior O-D matrix
Case (c): Skewed prior trip departure time profile as 10% - 20% - 30% - 40%
Case (d): Skewed prior trip departure time profile as 40% - 30% - 20% - 10%
Case (e): 50% increase of the prior O-D trip demand
Case (g): 6 trip departure intervals of 10-minutes length
Case (g): 2 trip departure intervals of 30-minutes length
Case (h): 5% variation of observed link flows from the DUO flow pattern

TABLE 2  Sensitivity analysis of algorithm performances for the case of actual link counts:

(a) Moderate size of O-D trip flows
(b) 50% growth in O-D trip flows

TABLE 3  Sensitivity analysis of algorithm performances for the case of actual link counts using different network scale and link count availability for moderate (a), (c) and increasing (b), (d) size of O-D trip flows.

case (i) Initial network scale and link count availability
case (ii) Increasing network scale
case (iii) Reducing link count availability

FIGURES:

FIGURE 1  Graphical illustration of the central part of the Greater Athens Area network

FIGURE 2  The performance of the MPP vs. the number of inner- and outer-loop iterations
TABLE 1  
Sensitivity analysis of algorithm performances for the case of simulated link flows:

Case (a): 10% variation of the assigned O-D flows from the prior O-D matrix
Case (b): 20% variation of the assigned O-D flows from the prior O-D matrix
Case (c): Skewed prior trip departure time profile as 10% - 20% - 30% - 40%
Case (d): Skewed prior trip departure time profile as 40% - 30% - 20% - 10%
Case (e): 50% increase of the prior O-D trip demand
Case (f): 6 trip departure intervals of 10-minutes length
Case (g): 2 trip departure intervals of 30-minutes length
Case (h): 5% variation of observed link flows from the DUO flow pattern

<table>
<thead>
<tr>
<th>Algorithm:</th>
<th>MART</th>
<th>RMART</th>
<th>DIMAP</th>
<th>Algorithm:</th>
<th>MART</th>
<th>RMART</th>
<th>DIMAP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case (a)</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>Case (e)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$MSE_{OD}$ (%)</td>
<td>17.269</td>
<td>18.140</td>
<td>14.861</td>
<td>$MSE_{OD}$ (%)</td>
<td>49,924</td>
<td>49,822</td>
<td>47,066</td>
</tr>
<tr>
<td>$RRMSE_{OD}$ (%)</td>
<td>80.216</td>
<td>78.857</td>
<td>68.935</td>
<td>$RRMSE_{OD}$ (%)</td>
<td>89,912</td>
<td>89,892</td>
<td>86,549</td>
</tr>
<tr>
<td>CPU time (sec.)</td>
<td>5.88</td>
<td>5.71</td>
<td>6.86</td>
<td>CPU time (sec.)</td>
<td>6.08</td>
<td>5.83</td>
<td>10.08</td>
</tr>
<tr>
<td>No. of Iterations</td>
<td>161</td>
<td>44</td>
<td>68</td>
<td>No. of Iterations</td>
<td>180</td>
<td>86</td>
<td>74</td>
</tr>
<tr>
<td><strong>Case (b)</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>Case (f)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$MSE_{OD}$ (%)</td>
<td>24.132</td>
<td>19.498</td>
<td>15.977</td>
<td>$MSE_{OD}$ (%)</td>
<td>0.227</td>
<td>1.731</td>
<td>0.222</td>
</tr>
<tr>
<td>$RRMSE_{OD}$ (%)</td>
<td>84.481</td>
<td>82.061</td>
<td>77.039</td>
<td>$RRMSE_{OD}$ (%)</td>
<td>14.518</td>
<td>31.258</td>
<td>14.312</td>
</tr>
<tr>
<td>CPU time (sec.)</td>
<td>6.53</td>
<td>5.77</td>
<td>9.49</td>
<td>CPU time (sec.)</td>
<td>7.17</td>
<td>7.30</td>
<td>7.90</td>
</tr>
<tr>
<td>No. of Iterations</td>
<td>299</td>
<td>97</td>
<td>104</td>
<td>No. of Iterations</td>
<td>34</td>
<td>59</td>
<td>21</td>
</tr>
<tr>
<td><strong>Case (c)</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>Case (g)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$MSE_{OD}$ (%)</td>
<td>1.015</td>
<td>11.907</td>
<td>0.913</td>
<td>$MSE_{OD}$ (%)</td>
<td>63.030</td>
<td>149.958</td>
<td>58.321</td>
</tr>
<tr>
<td>$RRMSE_{OD}$ (%)</td>
<td>17.323</td>
<td>63.249</td>
<td>17.285</td>
<td>$RRMSE_{OD}$ (%)</td>
<td>81.302</td>
<td>123.144</td>
<td>78.491</td>
</tr>
<tr>
<td>CPU time (sec.)</td>
<td>5.27</td>
<td>5.44</td>
<td>5.83</td>
<td>CPU time (sec.)</td>
<td>3.16</td>
<td>3.11</td>
<td>5.00</td>
</tr>
<tr>
<td>No. of Iterations</td>
<td>23</td>
<td>37</td>
<td>13</td>
<td>No. of Iterations</td>
<td>32</td>
<td>28</td>
<td>17</td>
</tr>
<tr>
<td><strong>Case (d)</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>Case (h)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$MSE_{OD}$ (%)</td>
<td>5.966</td>
<td>10.979</td>
<td>4.929</td>
<td>$MSE_{OD}$ (%)</td>
<td>22,419</td>
<td>20,378</td>
<td>15,319</td>
</tr>
<tr>
<td>$RRMSE_{OD}$ (%)</td>
<td>46.778</td>
<td>85.018</td>
<td>43.167</td>
<td>$RRMSE_{OD}$ (%)</td>
<td>82,704</td>
<td>80,873</td>
<td>70,315</td>
</tr>
<tr>
<td>CPU time (sec.)</td>
<td>5.49</td>
<td>5.45</td>
<td>8.58</td>
<td>CPU time (sec.)</td>
<td>5.98</td>
<td>5.39</td>
<td>9.94</td>
</tr>
<tr>
<td>No. of Iterations</td>
<td>97</td>
<td>91</td>
<td>47</td>
<td>No. of Iterations</td>
<td>151</td>
<td>54</td>
<td>68</td>
</tr>
</tbody>
</table>
TABLE 2  
Sensitivity analysis of algorithm performances for the case of actual link counts:

(a) Moderate size of O-D trip flows
(b) 50% growth in O-D trip flows

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Initial $RRMSE_{LINK}$ (%)</th>
<th>Final $RRMSE_{LINK}$ (%)</th>
<th>Improvement (%)</th>
<th>CPU time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>74.516</td>
<td>40.018</td>
<td>46.225</td>
<td>22.94</td>
</tr>
<tr>
<td>RMART</td>
<td>74.516</td>
<td>36.479</td>
<td>50.980</td>
<td>30.10</td>
</tr>
<tr>
<td>DIMAP</td>
<td>74.516</td>
<td>42.963</td>
<td>42.266</td>
<td>92.67</td>
</tr>
</tbody>
</table>

Results obtained after 10 re-assignments per interval

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Initial $RRMSE_{LINK}$ (%)</th>
<th>Final $RRMSE_{LINK}$ (%)</th>
<th>Improvement (%)</th>
<th>CPU time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>73.603</td>
<td>36.456</td>
<td>50.470</td>
<td>113.96</td>
</tr>
<tr>
<td>RMART</td>
<td>73.603</td>
<td>35.554</td>
<td>51.700</td>
<td>142.57</td>
</tr>
<tr>
<td>DIMAP</td>
<td>73.603</td>
<td>37.680</td>
<td>48.807</td>
<td>185.60</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Initial $RRMSE_{LINK}$ (%)</th>
<th>Final $RRMSE_{LINK}$ (%)</th>
<th>Improvement (%)</th>
<th>CPU time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>82.261</td>
<td>45.855</td>
<td>44.256</td>
<td>26.28</td>
</tr>
<tr>
<td>RMART</td>
<td>82.261</td>
<td>39.227</td>
<td>52.314</td>
<td>32.50</td>
</tr>
<tr>
<td>DIMAP</td>
<td>82.261</td>
<td>47.716</td>
<td>41.995</td>
<td>96.01</td>
</tr>
</tbody>
</table>

Results obtained after 10 re-assignments per interval

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Initial $RRMSE_{LINK}$ (%)</th>
<th>Final $RRMSE_{LINK}$ (%)</th>
<th>Improvement (%)</th>
<th>CPU time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>75.714</td>
<td>32.624</td>
<td>56.912</td>
<td>115.73</td>
</tr>
<tr>
<td>RMART</td>
<td>75.714</td>
<td>31.523</td>
<td>58.366</td>
<td>143.10</td>
</tr>
<tr>
<td>DIMAP</td>
<td>75.714</td>
<td>34.671</td>
<td>54.208</td>
<td>191.82</td>
</tr>
</tbody>
</table>

(b)
**TABLE 3**  Sensitivity analysis of algorithm performances for the case of actual link counts using different network scale and link count availability for moderate (a), (c) and increasing (b), (d) size of O-D trip flows.

*Case (i)* Initial network scale and link count availability  
*Case (ii)* Increasing network scale  
*Case (iii)* Reducing link count availability

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Case (i)</th>
<th>Case (ii)</th>
<th>Case (iii)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>90.04</td>
<td>43.11</td>
<td>81.37</td>
</tr>
<tr>
<td>RMART</td>
<td>51.86</td>
<td>31.49</td>
<td>34.73</td>
</tr>
<tr>
<td>DIMAP</td>
<td>69.80</td>
<td>42.19</td>
<td>54.83</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Case (i)</th>
<th>Case (ii)</th>
<th>Case (iii)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>24.36</td>
<td>49.33</td>
<td>27.05</td>
</tr>
<tr>
<td>RMART</td>
<td>4.77</td>
<td>24.36</td>
<td>12.24</td>
</tr>
<tr>
<td>DIMAP</td>
<td>28.47</td>
<td>50.45</td>
<td>19.44</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Case (ii)</th>
<th>Case (iii)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>43.191</td>
<td>35.612</td>
</tr>
<tr>
<td>RMART</td>
<td>38.650</td>
<td>39.891</td>
</tr>
<tr>
<td>DIMAP</td>
<td>40.210</td>
<td>37.884</td>
</tr>
</tbody>
</table>

(c)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Case (ii)</th>
<th>Case (iii)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>37.390</td>
<td>30.534</td>
</tr>
<tr>
<td>RMART</td>
<td>32.974</td>
<td>32.432</td>
</tr>
<tr>
<td>DIMAP</td>
<td>36.982</td>
<td>31.434</td>
</tr>
</tbody>
</table>

(d)
FIGURE 1  Graphical illustration of the central part of the Greater Athens Area network
FIGURE 2 The performance of the MPP vs. the number of inner- and outer-loop iterations