Paper No. 03-2186

Real-Time Dynamic Origin–Destination Matrix Adjustment with Simulated and Actual Link Flows in Urban Networks

Theodore Tsekeris and Antony Stathopoulos

The efficiency and robustness of different real-time dynamic origin–destination (O-D) matrix adjustment algorithms were investigated 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 with simulated and actual observed link flows. Implementation of the algorithms, which are coupled with a quasi-dynamic traffic assignment model, indicated their convergent behavior and their potential for handling realistic urban-scale network problems in terms of both accuracy and computational time. The main factors influencing the numerical performance of each algorithm were identified and analyzed. Their relative efficiency was found to be particularly dependent on the level at which the assigned flows approximate the observed link flows. These results may provide insights into the suitability of each algorithm for diverse application domains, including freeways, small networks, and large-scale urban networks, where a different quality of O-D information is usually available.

The origin–destination (O-D) trip matrix estimation is a longstanding 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 usually are addressed through the use of some dynamic traffic assignment (DTA) model. On the other hand, the approaches that do not incorporate a traffic assignment component, usually referred to as non-DTA-based approaches, basically can 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 generally can be regarded as more behaviorally sound and robust for addressing the various time-dependent aspects involved in the operation of large-scale networks than the non-DTA-based approach. Nonetheless, its performance can entail a significant computational burden (*6*).

This paper describes and compares several algorithms for the time-dependent O-D matrix estimation problem with 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 calculating the dynamic traffic loading conditions over the network. It is based on a simulation procedure developed by Stathopoulos et al. (*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 usually are 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 typically can be 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 usually is referred to as the matrix adjustment or balancing problem. However, relatively little attention has been paid to assessing their efficiency and robustness for online dynamic and realistic-scale urban network applications.

Department of Transportation Planning and Engineering, School of Civil Engineering, National Technical University of Athens, 5 Iroon Polytechniou, 157 73 Athens, Greece.

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 also was referenced elsewhere (*9*), is the multiplicative algebraic reconstruction technique (MART), developed by Gordon et al. (*14*). The convergence properties of MART are presented elsewhere (*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 the Revised Multiplicative Algebraic Reconstruction Technique (RMART), to enhance its numerical performance and convergence behavior, without submitting any analytical proof for its convergence.

This paper provides appropriate dynamic extensions to the MPP and MART algorithms, together with a hybrid (doubly iterative) matrix adjustment procedure. These algorithms are coupled with the quasi-DTA model of PARCMAN to estimate dynamic trip departure rates and, subsequently, O-D matrices over a series of successive time intervals. Prompted by the need to emphasize real-life large-scale urban networks, the algorithms are tested on the central part of the greater Athens area network with 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 the assumptions underlying them. This paper is organized as follows: the next section describes the structure of the quasi-DTA model, the third section describes the various dynamic O-D matrix adjustment algorithms, the fourth section presents the simulation tests and analyses of the results, and the fifth section provides conclusions and contributions of the paper.

QUASI-DTA PROCEDURE

This section describes the quasi-DTA model to be coupled with the time-dependent O-D matrix estimation algorithms. The various approaches that have been proposed to formulate and solve the DTA problem generally can be classified in two broad categories, as discussed elsewhere (*6*). The first category refers to the analytical DTA approaches, including models on the basis of mathematical programming, optimal control theory, and variational inequalities. The second category refers to simulation-based DTA, in which the vehicles are divided into a number of packets (e.g., 10 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 detailed investigation of a traveler's behavior, while 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 or her destination, based on the travel time perceived according to the currently prevailing traffic conditions, 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., $\tau = 15$ min), the model, whose mathematical formulation follows, calculates at the beginning of each interval the minimum-cost path from each origin. Then, it assigns each packet of vehicles departing from the corresponding origin onto the best route to reach the intended destination.

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 $\mathbf{x}$ be the O-D matrix whose elements $x_{ij}^{\tau_d} \in X^*$ denote the number of vehicular trips between the nth O-D pair, departing from origin $i \in I$ during time interval $\tau_d \in T_d$ directed toward 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 $x_{ij}^{\tau_d}$ trips may use different network paths connecting O-D pair (i, j). Hence, trip demand $x_{ij}^{\tau_d}$ will give rise to path flows $h_p^{\tau_d} \in \mathrm{H}$, 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). Also consider $M_O \subseteq M$ the total number of the observed links that contain a traffic counter, y_m^τ the observed traffic count on the mth link, and $\hat{y}_m^\tau$ the assigned (estimated) link volume on that link during time interval $\tau \in T$. This interval τ refers to the interval in which the link flow y_m^τ has been counted and 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 typically ranges from 1 to 3 h. The intervals τ_d and τ, and, hence, time periods T_d and T, do not necessarily coincide as τ_d may indicate an interval before τ.

Then, the DUO conditions, through the quasi-DTA model formulation, can be stated as follows:

$$\text{minimize } \Im(\hat{y}) = \sum_{m \in M} \sum_{\tau \in T} \int_0^{\hat{y}_m^\tau} c_m^\tau(s)\,ds \qquad \text{subject to } \hat{y} \in Y^* \tag{1}$$

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

$$\sum_{p \in P_{ij}} h_p^{\tau_d} = x_{ij}^{\tau_d} \qquad \forall (i, j),\ \forall \tau_d \in T_d \tag{2}$$

$$h_p^{\tau_d} \geq 0 \qquad \forall p \in P_{ij}, \forall (i, j) \tag{3}$$

$$\sum_{p \in P_{ij}} \sum_{\tau_d \in T_d} \delta_{pm}^{\tau_d \tau} h_p^{\tau_d} = \hat{y}_m \qquad \forall m \in M, \forall \tau \in T \tag{4}$$

where

$$\delta_{pm}^{\tau_d \tau} = \begin{cases} 1 & \text{if path } p \in P_{ij} \text{ uses link } m, \forall m \in M, \\ & \forall p \in P_{ij}, \forall \tau_d \in T_d, \forall \tau \in T \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The binary integers $\delta_{pm}^{\tau_d \tau}$ constitute the link–route incidence matrix Δ and take values equal to 1 if the link m belongs to the path p, and 0 otherwise. Let u_{pm}^τ denote the inflow rate into and v_{pm}^τ the exit rate from link m of path $p \in P_{ij}$ during time interval τ. Then, the number of vehicles on link m at any interval $\tau \in T$ can be expressed as follows:

$$\hat{y}_m^\tau = \sum_{p \in P_{ij}} \left(u_{pm}^\tau - v_{pm}^\tau\right) \qquad \forall m \in M, \forall \tau \in T \tag{6}$$

In addition, let $\mathbf{R}$ be the route choice matrix, with elements $r_p^{\tau_d}$ being the fraction of trip demand between O-D pair (i, j) that is assigned during interval τ_d and uses path $p \in P_{ij}$. Then, it holds that

$$\sum_{p \in P_{ij}} r_p^{\tau_d} = 1 \qquad \forall (i, j), \forall \tau_d \in T_d \tag{7}$$

and

$$r_p^{\tau_d} \geq 0 \qquad \forall p \in P_{ij}, \forall \tau_d \in T_d \tag{8}$$

Then, the path flow variable can be expressed as a product between the O-D matrix and the route choice matrix $\mathbf{R}$ variables as follows:

$$h_p^{\tau_d} = \sum_{ij} \sum_{p \in P_{ij}} x_{ij}^{\tau_d} r_p^{\tau_d} \qquad \forall \tau_d \in T_d \tag{9}$$

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:

$$\hat{y}_m^\tau = \sum_{ij} \sum_{\tau_d \in T_d} x_{ij}^{\tau_d} \sum_{p \in P_{ij}} \delta_{pm}^{\tau_d \tau} r_p^{\tau_d} = \sum_{ij} \sum_{\tau_d \in T_d} \alpha_{ijm}^{\tau_d \tau} x_{ij}^{\tau_d} \qquad \forall \tau \in T \tag{10}$$

where $\alpha_{ijm}^{\tau_d \tau}$ 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 τ_d directed toward destination j will use the mth link during the time interval τ. These $\alpha_{ijm}^{\tau_d \tau}$ proportions constitute the elements of the $(N \times M_O)$ assignment matrix $\mathbf{A}$. The preceding formulation provides the O-D-specific definition of assignment proportions. To enhance computational efficiency, the present model endogenously calculates origin- (or departure-) specific link-use proportions $\alpha_{im}^{\tau_d\tau}$ for each interval τ. Hence, Equation 10 can be reduced as follows:

$$\hat{y}_m^{\tau} = \sum_{i\in I} x_i^{\tau_d} \sum_{p\in P_i} \delta_{pm}^{\tau_d\tau}\, r_p^{\tau_d} = \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{im}^{\tau_d\tau}\, x_i^{\tau_d} \tag{11}$$

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

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

$$C_p^{\tau} = \sum_{m\in M} \delta_{pm}^{\tau_d\tau}\, c_m^{\tau} \qquad \forall p \in P_i, \forall \tau_d \in T_d, \forall \tau \in T \tag{12}$$

The flow circulation over the network is governed by a strictly positive, flow-dependent delay function D_m 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 τ as follows:

$$t_m(\tau) = t(\tau) + D_m(\hat{y}_m^{\tau}) \tag{13}$$

The formulation of D_m function depends on the level of queuing and the capacity y_m^c of each link m. The link travel cost c_m^{τ} is expressed by the function of travel time t_m, whose specification is based on the corresponding travel time functions of SATURN (*19*) as follows:

If $\hat{y}_m^{\tau} \le y_m^c$, then

$$t_m = t_0 + \left(\frac{\hat{y}_m^{\tau}}{y_m^c}\right)^{\mu}(t_c - t) \qquad \forall \tau \in T \tag{14}$$

If $\hat{y}_m^{\tau} > y_m^c$, then

$$t_m = t_0 + \left(\frac{\hat{y}_m^{\tau}}{y_m^c}\right)^{\mu}(t_c - t) + \frac{1}{2}\left(\frac{\hat{y}_m^{\tau}}{y_m^c} - 1\right)\tau \qquad \forall \tau \in T \tag{15}$$

where

- t_0 = travel time in free-flow conditions,
- t_c = travel time at capacity, and
- μ = calibrated exponent factor.

The second terms of the sums in Equations 14 and 15 represent the effect of congestion on the travel time along link m. The third term in Equation 15 represents the additional (queuing) delay time when the assigned flow $\hat{y}_m^{\tau}$ exceeds the capacity bound y_m^c. These last terms comprise the value of the time-dependent delay function D_m.

For calculation of the minimum-cost path between each O-D pair (i, j) at each departure time interval τ_d, a time-dependent minimum-cost tree building process was used based on D'Esopo's algorithm (*20*). To produce link volumes ($\hat{y}_m^{\tau}$) that are sufficiently close to the observed ones (y_m^{τ}), a series of successive adjustment iterations (assignment mappings) between the set of path and link flows, $h_p^{\tau_d}$ and $\hat{y}_m^{\tau}$, respectively, and the link-use proportions $\alpha_{im}^{\tau_d\tau}$ may be required (e.g., 10 iterations) per interval τ.

Although no rigorous proof of its convergence properties exists, the present simulation-based model can provide a computationally efficient tool for quasi-dynamic mapping of the O-D demand to the link flow pattern and the calculation of time-dependent trip departure rates. Even though it does not explicitly represent several microscopic traffic phenomena, like the structural evolution of queuing and spillbacks, it enables the macroscopic consideration of deterministic control delays and variable travel time effects. Because of its intrinsic simplicity, the proposed model constitutes a framework that is realistic from the software point of view: it allows simulation of realistically large-scale networks (i.e., spanning several thousand links servicing several hundred thousand travelers) with simple personal computer facilities [typically with 128 megabytes of random-access memory (MB RAM)] in a period of a few seconds. Furthermore, it has a relatively low level of data requirements, because it uses the same data as a standard static assignment model.

DYNAMIC O-D MATRIX ADJUSTMENT PROBLEM

Problem Formulation

Based on Willumsen (*21*), 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:

$$\text{minimize } \lambda_i^{\tau_d} \sum_{i\in I} x_i^{\tau_d}\left[\log\left(\frac{x_i^{\tau_d}}{\hat{x}_i^{\tau_d}}\right) - 1\right] \qquad \forall \tau_d \in T_d \tag{16}$$

subject to

$$y_m^{\tau} = \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{im}^{\tau_d\tau}\, x_i^{\tau_d} \qquad \forall m \in M_O, \forall \tau \in T \tag{17}$$

and

$$x_i^{\tau_d} \ge 0 \qquad \forall i \in I, \forall \tau_d \in T_d \tag{18}$$

where $\hat{x}_i^{\tau_d}$ corresponds to the row (origin) sums of the $(I \times Z)$ matrix $\bar{\mathbf{x}} = \{\hat{x}_{ij}^{\tau_d}\}$. This matrix, known as a prior O-D matrix, may be a preliminary estimate based on, for example, partial surveys, time-of-day historical information, or results from previous departure intervals of the same day. The factor $\lambda_i^{\tau_d}$ denotes relevant weights that allow for some prior O-D matrix elements to be assumed more reliable than others. In real-time traffic networks, the reliability of prior information on O-D pattern and trip departure rates is unknown in practice, particularly because there is no clear process available for accounting for the differences in knowledge or confidence between prior matrix elements. Thus, the $\lambda_i^{\tau_d}$ values can be reasonably assumed as equal (to unity) for all O-D pairs.

In this paper, time-of-day historical estimates are used (see section Presentation of Simulation Experiments). 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 theoretically could be implemented to utilize estimates derived from previous departure intervals. These approaches, principally based on Bayesian updating methods (*2, 5*), were found experimentally to produce a considerable computational load when applied to the given study area. This is particularly due to the large number (thousands) of O-D pairs, the spatial distribution of the network, and the high levels of congestion. These factors essentially render online application of such approaches in real-world urban network problems with so many dimensions, like the current one, practically impossible.

The normalization technique used to transform the trip departures $x_i^{\tau_d}$ into the equivalent O-D trip flows $x_{ij}^{\tau_d}$ is based on the assumption of fixed arrival rates from each origin, according to the prior O-D demand pattern. Thus, the adjusted O-D matrices are given as follows (*13*):

$$x_i^{\tau_d} = x_i^{\tau_d}\left(\frac{\sum_{\tau_d \in T_d} \hat{x}_{ij}^{\tau_d}}{\sum_{\tau_d \in T_d} \hat{x}_i^{\tau_d}}\right) \qquad \forall i \in I, \forall j \in Z, \forall \tau_d \in T_d \tag{19}$$

where $\hat{x}_{ij}^{\tau_d}$ denotes the elements of the prior O-D matrix.

The preceding transformation process 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 τ_d may be estimated to satisfy the constraints of the quasi-DTA problem.

MPP

The MPP seeks to optimize a set of balancing factors β_m^τ 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 Taylor (*22*)], the link-use proportions $\alpha_{im}^{\tau_d\tau}$ are held constant throughout the iteration loops of the O-D matrix adjustment process during each interval τ. The loops for which $\alpha_{im}^{\tau_d\tau}$ values are held constant are referred to here as inner-loop iterations λ_{IN}. On the contrary, the suggested approach periodically updates the $\alpha_{im}^{\tau_d\tau}$ values after a fixed number of inner-loop iterations λ_{IN}. The total number of these updates is represented through the outer-loop iterations λ_{OUT}. In this way, the solution procedure endogenously takes into account the fact that users may reconsider their route choice decisions (by recalculating the minimum-cost path) from each origin $i \in I$ at each departure interval τ_d based on the prevailing congestion conditions at interval τ. The modified dynamic version of MPP, whose solution is based on the optimality conditions of the Lagrange multiplier method, can be described in the following steps:

- Step 1: Assign the $\hat{\mathbf{x}}$ matrix for departure interval τ_d to calculate the $\alpha_{im}^{\tau_d\tau}$ proportions. Set the β_m^τ values equal to unity for each $m \in M_O$ and $\tau \in T$. Set the number of outer-loop iterations $\lambda_{OUT} = 0$.
- Step 2: Set the number of inner-loop iterations $\lambda_{IN} = 0$.
- Step 3: Increment the number of inner-loop iterations $\lambda_{IN} = \lambda_{IN} + 1$.
- Step 4: Select a new combination (m, τ) of observed link and time interval and calculate the link volumes as follows:

$$\hat{y}_m^\tau(\lambda_{IN}) = \sum_{i\in I}\sum_{\tau_d \in T_d} \alpha_{im}^{\tau_d\tau}\, x_i^{\tau_d} \prod_{m\tau}\left[\beta_m^\tau(\lambda_{IN})\right]^{a_{im}^{\tau_d\tau}} \tag{20}$$

- Step 5: Update the β_m^τ values as follows:

$$\beta_m^\tau(\lambda_{IN}+1) = \beta_m^\tau(\lambda_{IN})\phi_m^\tau \tag{21}$$

where ϕ_m^τ is an adjustment factor obtained for the given combination (m, τ) by solving the following equation by a modified Newton–Raphson method:

$$y_m^\tau = \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{im}^{\tau_d\tau} x_i^{\tau_d} \prod_{m\tau}\left[\beta_m^\tau(\lambda_{IN})\right]^{\alpha_{im}^{\tau_d\tau}} (\phi_m^\tau)\alpha_{im}^{\tau_d\tau} \tag{22}$$

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

$$x_i^{\tau_d} = \hat{x}_i^{\tau_d} \prod_{m\tau}\left[\beta_m^\tau(\lambda_{IN}+1)\right]^{a_{im}^{\tau_d\tau}} \tag{23}$$

Then, estimate the adjusted O-D matrix based on Equation 19.

- Step 7: If not all (m, τ) 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, τ) combination is greater than a prespecified minimum value δ, or a maximum number of inner-loop iterations max λ_{IN} has not been reached, then return to Step 3. Else continue to Step 9.
- Step 9: Increment the number of outer-loop iterations $\lambda_{OUT} = \lambda_{OUT} + 1$. If a maximum number of outer-loop iterations max λ_{OUT} has been reached, then stop. Else, assign the updated O-D matrix to obtain a new set of $\alpha_{im}^{\tau_d\tau}$ proportions and go to Step 2.

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 $\lambda = 1$. Initialize $x_i^{\tau_d}(\lambda) = \hat{x}_i^{\tau_d}$, $\forall i \in I, \forall \tau_d \in T_d$.
- Step 2: Increment the number of iterations $\lambda = \lambda + 1$.
- Step 3: Update the trip departure flows as follows:

$$x_i^{\tau_d}(\lambda+1) = \left\{\prod_m \left[\frac{y_m^\tau}{\sum_{l\in I}\sum_{\tau_d\in T_d} \alpha_{im}^{\tau_d\tau} x_i^{\tau_d}(\lambda)}\right]^{s_i^{\tau_d\tau}\alpha_{im}^{\tau_d\tau}}\right\} x_i^{\tau_d}(\lambda) \qquad \forall i \in I, \forall \tau_d \in T_d \tag{24}$$

where $s_i^{\tau_d\tau}$ is a normalizing exponent factor used to enhance the numerical stability of the algorithm convergence. This is given as follows:

$$s_i^{\tau_d\tau} = \frac{1}{\sum_{m\in M_O} \alpha_{im}^{\tau_d\tau}} \qquad \forall i \in I, \forall \tau_d \in T_d, \forall \tau \in T \tag{25}$$

- Step 4: Estimate the updated O-D matrix based on Equation 19 and the corresponding new set of link volumes based on Equation 11.
- Step 5: If the distance between the observed and estimated link volumes is greater than a prespecified minimum value δ, or a maximum number of iterations max λ has not been reached, then return to Step 2. Else Stop; the matrix with elements $x_{ij}^{\tau_d}(\lambda)$ 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 in the following steps:

- Step 1, Step 2, Step 3, and Step 4: as in MART earlier.
- Step 5: If the distance between the observed and estimated link volumes is greater than a prespecified minimum value δ, or a maximum number of iterations max λ has not been reached, then proceed to Step 6. Else stop; the matrix with elements $x_{ij}^{\tau_d}(\lambda)$ is the final adjusted O-D matrix.
- Step 6: Perform another MART iteration to estimate a new set of trip departures $z_i^{\tau_d}$ based on $x_i^{\tau_d}(\lambda + 1)$.
- Step 7: Perform a diagonal search as follows:

$$x_i^{\tau_d}(\lambda + 2) = z_i^{\tau_d} + b\left[z_i^{\tau_d} - x_i^{\tau_d}(\lambda + 1)\right] \qquad \forall i \in I, \forall \tau_d \in T_d \tag{26}$$

where

$$b = \max\{0, \min[b_1, b_2]\} \tag{27}$$

$$b_1 = \min\left\{ \frac{z_i^{\tau_d}}{x_i^{\tau_d}(\lambda + 1) - z_i^{\tau_d}} \,\middle|\, x_i^{\tau_d}(\lambda - 1) - z_i^{\tau_d} > 0, \quad \forall i \in I, \forall \tau_d \in T_d \right\} \tag{28}$$

If $\sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{i\mu}^{\tau_d\tau} z_i^{\tau_d} - \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{i\mu}^{\tau_d\tau} x_i^{\tau_d}(\lambda) \neq 0$

$$b_2 = \frac{y_\mu^\tau - \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{i\mu}^{\tau_d\tau} z_i^{\tau_d}}{\sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{i\mu}^{\tau_d\tau} z_i^{\tau_d} - \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{i\mu}^{\tau_d\tau} x_i^{\tau_d}(\lambda)} \tag{29}$$

where μ is such that

$$\left| y_\mu^\tau - \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{i\mu}^{\tau_d\tau} z_i^{\tau_d} \right| = \max\left\{ \left| y_m^\tau - \sum_{i\in I}\sum_{\tau_d\in T_d} \alpha_{i\mu}^{\tau_d\tau} z_i^{\tau_d} \right|, \forall m \in M_O \right\}$$

Else, $b_2 = 1$;

and return to Step 2.

Doubly Iterative Matrix Adjustment Procedure

Another possible dynamic extension of the aforementioned algorithms is the suitable combination of them to improve their performance characteristics. Such a procedure is referred to here as doubly iterative matrix adjustment procedure (DIMAP), in the sense that the multiproportionally adjusted matrix in each iteration is one that has already been corrected (reconstructed) on the basis of the MART procedure. DIMAP aims to further enhance the consistency between the trip departure rates from each origin zone and the observed link flows. The algorithm convergence can be ensured given that, as mentioned earlier, both MPP and MART have a proven convergent behavior. The steps of DIMAP can be described as follows:

- Step I: Set the number of MART iterations $\lambda_1 = 1$. Initialize $x_i^{\tau_d}(\lambda) = \hat{x}_i^{\tau_d}$, $\forall i \in I$, $\forall \tau_d \in T_d$. Increment the number of iterations $\lambda_1 = \lambda_1 + 1$.
- Step II: Implement Step 3 described in the section MART to perform a MART iteration and update the trip departure rates.
- Step III: Set the number of inner-loop iterations for MPP equal to $\lambda_2 = 0$.
- Step IV: Perform an inner-loop iteration to multiproportionally adjust the updated trip departure rates following Step 2 to Step 7 as described in the section MPP.
- Step V: If the distance between the estimated and observed link volumes for each (m, τ) combination is greater than a prespecified minimum value δ, or a maximum number of inner-loop iterations max λ_2 has not been reached, then return to Step IV. Else continue to Step VI.
- Step VI: Increment the number of MART iterations $\lambda_1 = \lambda_1 + 1$. If a maximum number of iterations max λ_1 has been reached, then stop. Else, return to Step II.

NUMERICAL TESTS AND RESULTS

Presentation of 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 on a real urban-scale road network, that of the central part of the greater Athens area network. This network was modeled with 151 assignment nodes, 256 assignment links, and a total number of 1,936 O-D pairs—that is, the assigned O-D matrix is 44×44. Figure 1 presents a graphical representation of the network. In the first set of simulation tests (see the next section), the time-dependent O-D matrix estimation is undertaken with 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 $\hat{x}_{ij}^{\tau_d}$ corresponding at each interval τ_d, are assumed as the ground-truth link counts. The prior matrix is based on extensive O-D travel surveys in the greater Athens area and refers to the morning peak period, including a total of 17,340 vehicles per hour to be loaded onto the network.

In the second set of simulation tests (see section Estimation with Real Traffic Counts), the estimation is carried out with 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-s cycle. The real-time counting system identifies and excludes data from malfunctioning detectors, while it uses smoothed flow values to avoid short-term local fluctuations, attributed to significant random and nonrecurring traffic episodes. In this case, the simulation period spans the whole morning period (i.e., 6:00 to 9:00 a.m.), which is partitioned into 12 equal intervals of 15 min.

Estimation with Simulated Link Flows

In this paper, the percentage or relative root mean square error ($RRMSE_{LINK}$) was used to measure the distance between the estimated and observed link flows to control the convergence of the algorithms. The $RRMSE_{LINK}$ is calculated as follows:

$$RRMSE_{LINK} = \sqrt{MSE_{LINK}} \left[\frac{\sum_{m\in M_O} y_m^\tau}{M_O} \right]^{-1} \qquad \forall \tau \in T \tag{30}$$

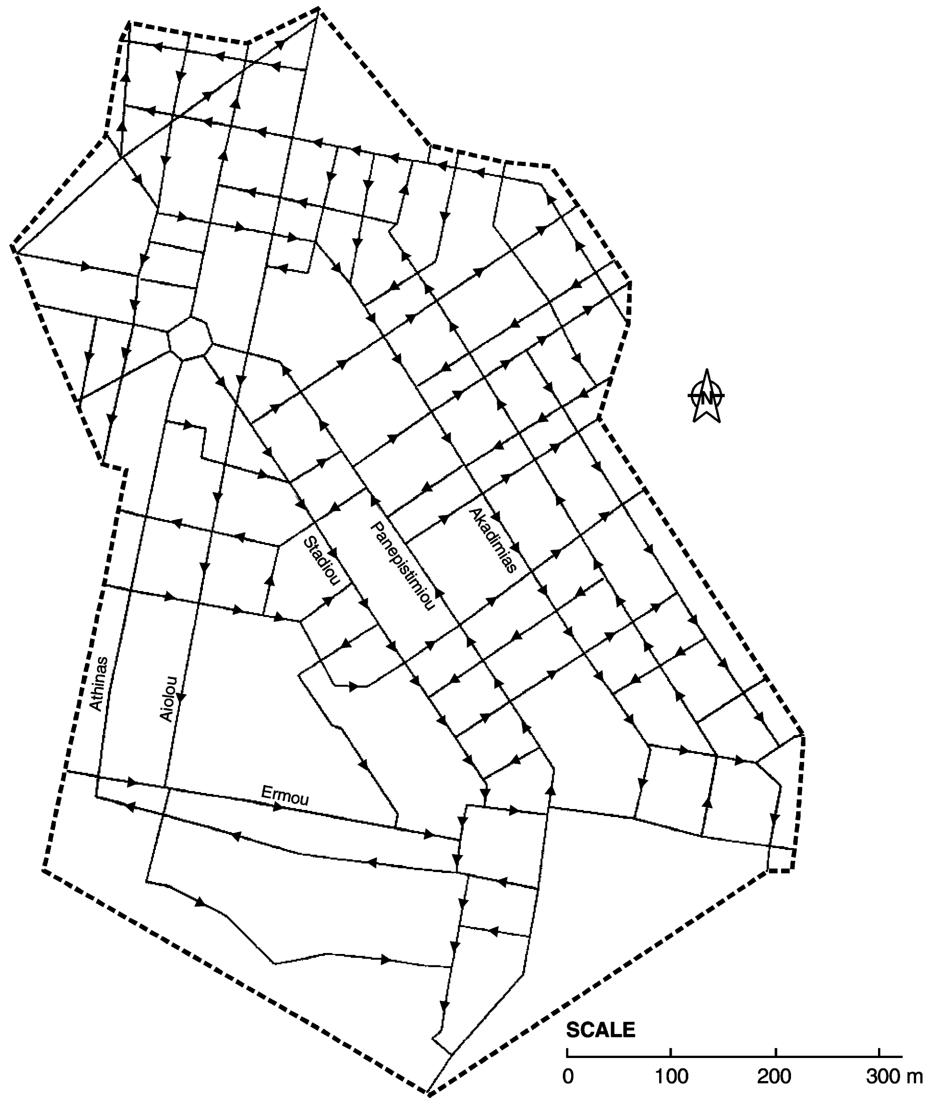

FIGURE 1 Graphical illustration of central part of the greater Athens area network.

where MSE_{LINK} is the corresponding mean square error, given as follows:

$$MSE_{LINK} = \frac{\sum_{m \in M_O} (\hat{y}_m^\tau - y_m^\tau)^2}{M_O} \qquad \forall \tau \in T \tag{31}$$

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

$$RRMSE_{OD} = \sqrt{MSE_{OD}} \left[\frac{\sum_{ij} \hat{x}_{ij}^{\tau_d}}{N^+} \right]^{-1} \qquad \forall \tau_d \in T_d \tag{32}$$

and

$$MSE_{OD} = \frac{\sum_{ij} \left(x_{ij}^{\tau_d} - \hat{x}_{ij}^{\tau_d} \right)^2}{N^+} \qquad \forall \tau_d \in T_d \tag{33}$$

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

Figure 2 presents the results obtained from implementation of MPP. Figure 2*a* indicates that MPP converges rapidly within the first four inner-loop iterations λ_{IN}, when the number of outer-loop iterations is set equal to $\lambda_{OUT} = 1$. $RRMSE_{LINK}$ falls below 5% for each interval τ after $\lambda_{OUT} = 13$ (see Figure 2*b*), with a number of $\lambda_{IN} = 10$ iterations for each outer-loop iteration, at about 75 s of central processing unit (CPU) time. MPP converges at a stationary point for each τ after $\lambda_{OUT} = 22$ (not indicated in Figure 2) at about 320 s, without reaching the strict convergence criterion of the minimum distance $\delta = 1\%$. For the results of MPP, the average level of nonconvergence (LNC) for each τ, which is given as

$$LNC(\%) = 100 \times \frac{|\delta - RRMSE_{LINK}|}{\delta} \tag{34}$$

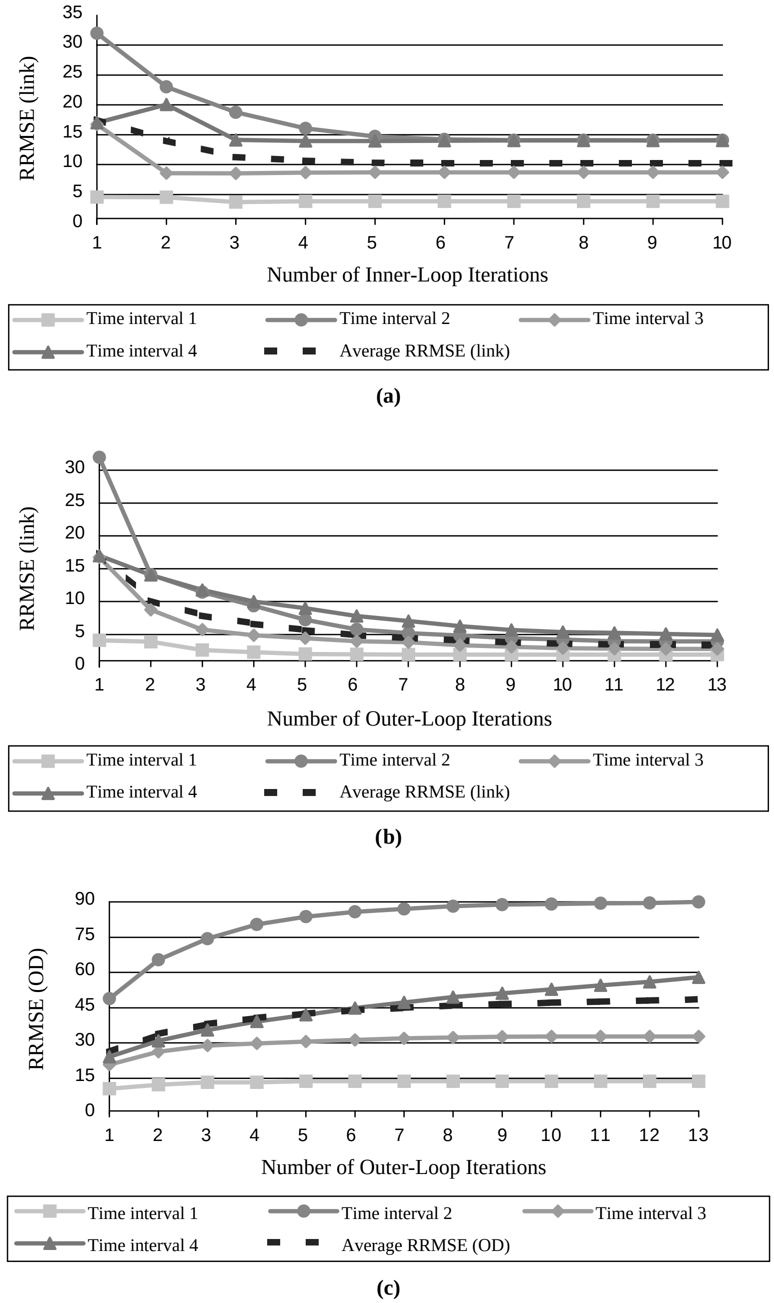

FIGURE 2 Performance of MPP versus number of inner- and outer-loop iterations.

was found equal to LNC = 180%. In contrast, all the other algorithms, when using simulated observed flows, were found to converge; in other words, they reached a level of nonconvergence equal to LNC = 0. Figure 2*c* indicates that the reduction of $RRMSE_{LINK}$ across the outer-loop iterations is associated with an increase of $RRMSE_{OD}$ for each interval τ. Similar trends between the O-D trip and link flow performance measures across iterations were also observed in the other algorithms. This finding essentially suggests a trade-off between optimal link flow pattern and trip departure rates. Moreover, all algorithms indicated a linear convergence behavior, except for the DIMAP case, which performed small disturbances decreasing with the number of iterations.

In contrast to MPP, the other algorithms satisfied the criterion of minimum distance $\delta = 1\%$ for each τ, while they required much less computational time for convergence, with the same prior trip distribution assumptions as before [see Case (*a*) in Table 1]. The significantly greater computational time associated with MPP may be attributed to use of the modified Newton–Raphson method and the need for periodically updating the link-use proportions. Based on Table 1, DIMAP showed the most efficient, in terms of MSE_{OD} and $RRMSE_{OD}$, and robust behavior, compared with both MART and RMART, with a range of different assumptions. These assumptions refer to different (lower and higher) levels of variability with respect to the prior trip distribution [Cases (*a*) and (*b*)], different directions

TABLE 1 Sensitivity Analysis of Algorithm Performances for the Case of Simulated Link Flows

Algorithm	MART	RMART	DIMAP	Algorithm	MART	RMART	DIMAP
Case (a)				***Case (e)***			
MSE_{OD} (%)	17.269	18.140	14.861	MSE_{OD} (%)	49.924	49.822	47.066
$RRMSE_{OD}$ (%)	80.216	78.587	68.935	$RRMSE_{OD}$ (%)	89.912	89.892	86.549
CPU time (s)	5.88	5.71	6.86	CPU time (s)	6.08	5.83	10.08
No. of Iterations	161	44	68	No. of Iterations	180	86	74
Case (b)				***Case (f)***			
MSE_{OD} (%)	24.132	19.498	15.977	MSE_{OD} (%)	0.227	1.731	0.222
$RRMSE_{OD}$ (%)	84.481	82.061	77.039	$RRMSE_{OD}$ (%)	14.518	31.258	14.312
CPU time (s)	6.53	5.77	9.49	CPU time (s)	7.17	7.30	7.90
No. of Iterations	299	97	104	No. of Iterations	34	59	21
Case (c)				***Case (g)***			
MSE_{OD} (%)	1.015	11.907	0.913	MSE_{OD} (%)	63.030	149.958	58.321
$RRMSE_{OD}$ (%)	17.323	63.249	17.285	$RRMSE_{OD}$ (%)	81.302	123.144	78.491
CPU time (s)	5.27	5.44	5.83	CPU time (s)	3.16	3.11	5.00
No. of Iterations	23	37	13	No. of Iterations	32	28	17
Case (d)				***Case (h)***			
MSE_{OD} (%)	5.966	10.979	4.929	MSE_{OD} (%)	22.419	20.378	15.319
$RRMSE_{OD}$ (%)	46.778	85.018	43.167	$RRMSE_{OD}$ (%)	82.704	80.873	70.315
CPU time (s)	5.49	5.45	8.58	CPU time (s)	5.98	5.39	9.94
No. of Iterations	97	91	47	No. of Iterations	151	54	68

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 in the prior O-D trip demand
Case *(f)*: 6 trip departure intervals of 10-minute length
Case *(g)*: 2 trip departure intervals of 30-minute length
Case *(h)*: 5% variation of observed link flows from the DUO flow pattern

of change in the prior demand [Cases (*c*) and (*d*)], the size of O-D trip flows, [Case (*e*)], different (shorter and longer) time interval durations [Cases (*f*) and (*g*)], and the variability of observed link flows from the DUO link flow pattern [Case (*h*)]. DIMAP requires a considerably smaller number of iterations to convergence compared with MART, although it is associated with higher CPU times.

In all algorithms, the increase in variability and size of O-D trip flows appears to adversely affect the reliability of O-D matrix estimates and the CPU times required for convergence. This outcome may be interpreted by the fact that trip makers, in the presence of volatile and growing demand conditions, cannot easily determine which paths correspond to lower travel costs, so they finally select a less optimal path. The effects of the variability of the (simulated) observed flows from the DUO link flow pattern are similar. In contrast, the skewed (nonsymmetric) loading of the assigned O-D matrix appears to enhance the performance of the tested algorithms. Nevertheless, the direction of change in demand conditions significantly influences their convergence behavior. Furthermore, shorter time interval durations (10 min) resulted in 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, compared with the longer interval durations (30 min). This improvement may be associated with a faster adjustment of users' decisions, related to departure time and route choice, with regard to the prevailing congestion conditions.

Estimation with Real Traffic Counts

In the case of using real traffic count data, the algorithm convergence is far from guaranteed, mainly because of the intrinsic complexities of the DTA procedure and the departure of real-life link flows from the DUO pattern. DTA should be considered as a prior assumption, which cannot actually be observed in practice. Hence, a maximum number of 200 iterations were specified as the termination criterion for each algorithm. In all cases, the algorithms were observed to asymptotically converge at a stationary $RRMSE_{LINK}$ value. This value was basically considered to evaluate the performance of each algorithm. Two different loading conditions were used to encounter possible effects of bias imposed by the prior demand information. First, the network was loaded with a trip matrix of moderate size (about 50,000 vehicles per hour). Next, the loading was performed assuming

TABLE 2 Sensitivity Analysis of Algorithm Performances for the Case of Actual Link Counts: Moderate Size of O-D Trip Flows

Algorithm	Initial $RRMSE_{LINK}$ (%)	Final $RRMSE_{LINK}$ (%)	Improvement (%)	CPU time (s)
MART	74.516	40.018	46.225	22.94
RMART	74.516	36.479	50.980	30.10
DIMAP	74.516	42.963	42.266	92.67
	Results obtained after 10 reassignments per interval			
	Initial $RRMSE_{LINK}$ (%)	Final $RRMSE_{LINK}$ (%)	Improvement (%)	CPU time (s)
MART	73.603	36.456	50.470	113.96
RMART	73.603	35.554	51.700	142.57
DIMAP	73.603	37.680	48.807	185.60

a 50% growth in total demand. Ten assignment mappings (successive quasi-DTA runs) were additionally considered per interval.

All cases indicated a significant reduction (between 40% and 60%) of the initial $RRMSE_{LINK}$ value, as obtained by assigning the prior matrix. Tables 2 and 3 indicate that the greatest improvements were attained by the consecutive dynamic reassignments of the trip matrix, particularly under increasing demand, at the expense of computational time. This outcome denotes the high sensitivity of the dynamic matrix adjustment to the accuracy of link-use proportions. RMART provided the greatest $RRMSE_{LINK}$ improvement, with MART and DIMAP following in sequence. MART displayed the fastest computational speed, while DIMAP required the longest 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 regard to the results typically obtained on the basis of simulated observed link flows.

A deeper understanding of algorithm performances 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 (RAE_d), as follows:

$$RAE_d = \frac{\sum_{i \in I} \left(x_i^{\tau_d} - \hat{x}_i^{\tau_d} \right)}{\sum_{i \in I} \hat{x}_i^{\tau_d}} \qquad \forall \tau_d \in T_d \tag{35}$$

In addition, the sensitivity of the efficiency and robustness of the algorithms was examined with respect to the network scale and 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 in carrying out 10 successive assignment runs per interval. Tables 4 and 5 indicate an overall increase in the produced trip departure rates, particularly for the moderately congested conditions. This increase can be attributed to the trip understatement underlying the survey-based O-D estimates and the traffic growth in the area. RMART was found to better capture the suppressed effects of congestion, in the sense of producing a smaller increase in departure rates, compared with MART and DIMAP. In the experiments assuming an increased network scale and reduced link count availability, referred to as Cases (*ii*) and (*iii*), respectively, the differences in the departure rates resulting from the moderate and increased O-D trip sizes are diminished, compared with 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.

Tables 6 and 7 indicate that RMART produces the lowest $RRMSE_{LINK}$ in Case (*ii*), compared with MART and DIMAP. The higher $RRMSE_{LINK}$, compared with Case (*i*), indicates that the increased network scale is associated with a greater uncertainty when calculating the link-use proportions from each origin. Conversely,

TABLE 3 Sensitivity Analysis of Algorithm Performances for the Case of Actual Link Counts: Increasing Size of O-D Trip Flows

Algorithm	Initial $RRMSE_{LINK}$ (%)	Final $RRMSE_{LINK}$ (%)	Improvement (%)	CPU time (s)
MART	82.261	45.855	44.256	26.28
RMART	82.261	39.227	52.314	32.50
DIMAP	82.261	47.716	41.995	96.01
	Results obtained after 10 reassignments per interval			
	Initial $RRMSE_{LINK}$ (%)	Final $RRMSE_{LINK}$ (%)	Improvement (%)	CPU time (s)
MART	75.714	32.624	56.912	115.73
RMART	75.714	31.523	58.366	143.10
DIMAP	75.714	34.671	54.208	191.82

TABLE 4 Performance of the Trip Departure Estimates of Each Algorithm for the Case of Actual Link Counts for Moderate 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.

Algorithm	*Case (i)* RAE$_d$ (%)	*Case (ii)* RAE$_d$ (%)	*Case (iii)* RAE$_d$ (%)
MART	90.04	43.11	81.37
RMART	51.86	31.49	34.73
DIMAP	69.80	42.19	54.83

TABLE 5 Performance of the Trip Departure Estimates of Each Algorithm for the Case of Actual Link Counts for Increasing 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.

Algorithm	*Case (i)* RAE$_d$ (%)	*Case (ii)* RAE$_d$ (%)	*Case (iii)* RAE$_d$ (%)
MART	24.36	49.33	27.05
RMART	4.77	24.36	12.24
DIMAP	28.47	50.45	19.44

MART and DIMAP produced lower RRMSE$_{LINK}$ for Case (*iii*) compared with RMART and the corresponding values in Case (*i*). The small relative changes in RRMSE$_{LINK}$ between Case (*i*) and Case (*iii*) suggest that the omitted link counts probably do not carry such significant information to affect considerably the resultant flow solution. Finally, Case (*ii*) is accompanied by a slight increase in CPU time because of the additional effort required by the quasi-DTA model in route processing. In contrast, Case (*iii*) results in a decrease in CPU time because of the reduced dimension of the assignment matrix.

CONCLUSIONS

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 and indicated significant superiority over the flow pattern produced by the prior matrix. The algorithm performance was found to be affected by the assumptions underlying the prior matrix structure, the time interval duration, the source of ground-truth flows, the DTA procedure, the network scale, and the link count availability.

For the case of simulated link flows, in which a satisfactory convergence level can be achieved, DIMAP resulted in the most reliable matrix and performed with greater robustness than the other algorithms. Thus, it may provide a competitive solution, whereas the assigned flows do sufficiently approximate the observed link flows. Such conditions may be practically met in freeway segments or in small and simplified networks. Conversely, MART and, particularly, its heuristic improvement RMART show a better potential to solve the real-time matrix adjustment problem when using actual observed flows in a real urban-scale network. The implementation of other, dynamically extended balancing algorithms, or possible improvements of them, would provide more insights into the problem of estimating optimal trip departure times and time-dependent O-D matrices.

TABLE 6 Performance of the Link Flow Solution and CPU Time of Each Algorithm for the Case of Actual Link Counts for Moderate Size of O-D Trip Flows: Case (*i*), Increasing Network Scale; Case (*ii*), Reducing Link Count Availability.

Algorithm	*Case (ii)* RRMSE$_{LINK}$ (%)	*Case (ii)* CPU time (s)	*Case (iii)* RRMSE$_{LINK}$ (%)	*Case (iii)* CPU time (s)
MART	43.191	115.15	35.612	110.46
RMART	38.650	144.41	39.891	139.13
DIMAP	40.210	189.91	37.884	179.32

TABLE 7 Performance of the Link Flow Solution and CPU Time of Each Algorithm for the Case of Actual Link Counts for Increasing Size of O-D Trip Flows: Case (*i*), Increasing Network Scale; Case (*ii*), Reducing Link Count Availability.

Algorithm	*Case (ii)* RRMSE$_{LINK}$ (%)	*Case (ii)* CPU time (s)	*Case (iii)* RRMSE$_{LINK}$ (%)	*Case (iii)* CPU time (s)
MART	37.390	117.55	30.534	112.77
RMART	32.974	146.35	32.432	140.15
DIMAP	36.982	196.23	31.434	185.16

REFERENCES

1. Tatineni, M. R., M. R. Lupa, D. B. Englund, and D. E. Boyce. Transportation Policy Analysis Using a Combined Model of Travel Choice. In *Transportation Research Record 1452,* TRB, National Research Council, Washington, D.C., 1994, pp. 10–17.
2. Ben-Akiva, M., M. Bierlaire, H. N. Koutsopoulos, and R. Mishalani. DynaMIT: A Simulation-Based System for Traffic Prediction and Guidance Generation. *Proc., TRISTAN III.* San Juan, Puerto Rico, 1998.
3. Chang, T-H., and Z-Y. Li. Optimization of Mainline Traffic Via an Adaptive Coordinated Ramp-Metering Control Model with Dynamic OD Estimation. *Transportation Research C,* Vol. 10, No. 2, 2002, pp. 99–120.
4. Cremer, M., and H. Keller. A New Class of Dynamic Methods for the Identification of Origin-Destination Flows. *Transportation Research B,* Vol. 21, No. 2, 1987, pp. 117–132.
5. Chang, G. L., and J. Wu. Recursive Estimation of Time-Varying Origin-Destination Flows from Traffic Counts in Freeway Corridors. *Transportation Research B,* Vol. 28, 1994, pp. 141–160.
6. Ziliaskopoulos, A. K., and S. Peeta. Review of Dynamic Traffic Assignment Models. *Networks and Spatial Economics,* Vol. 1, 2001, pp. 233–267.
7. Stathopoulos, A., J. Polak, A. Tillis, S. Mitropoulos, and B. Ryan. Parking Management and Control. *Advanced Telematics in Road Transport, Vol. 1: Proceedings of the DRIVE Conference,* Brussels, Belgium, Feb. 4–6, 1991, Elsevier, Oxford, 1991.
8. PARCMAN. *Parking Management, Control and Information System.* Final Report, The PARCMAN Consortium, Athens, 1992.
9. Lamond, B., and N. F. Stewart. Bregman's Balancing Method. *Transportation Research B,* Vol. 15, 1981, pp. 239–248.
10. Bregman, L. The Relaxation Method of Finding the Common Point of Convex Sets and Its Application to the Solution of Problems in Convex Programming. *USSR Computational Mathematics and Mathematical Physics,* Vol. 7, 1967, pp. 200–217.
11. Murchland, J. D. *The Multi-Proportional Problem.* Research Note JDM-263, University College London, London, 1977.
12. Van Zuylen, H. J., and L. G. Willumsen. The Most Likely Trip Matrix Estimated from Traffic Counts. *Transportation Research B,* Vol. 14, No. 4, 1980, pp. 281–293.
13. Janson, B. N., and F. Southworth. Estimating Departure Times from Traffic Counts Using Dynamic Assignment. *Transportation Research B,* Vol. 26, No. 1, 1992, pp. 3–16.

14. Gordon, R., R. Bender, and G. T. Hermann. Algebraic Reconstruction Techniques (ART) for Three-Dimensional Electron Microscopy and X-Ray Photography. *Journal of Theoretical Biology,* Vol. 29, 1970, pp. 471–481.
15. Darroch, J. N., and D. Ratcliff. Generalised Iterative Scaling for Log-Linear Models. *Annals of Mathematical Statistics,* Vol. 43, 1972, pp. 1470–1480.
16. Schneider, M. H., and S. A. Zenios. A Comparative Study of Algorithms for Matrix Balancing. *Operations Research,* Vol. 38, No. 3, 1990, pp. 439–455.
17. Leurent, F. Dual Solution of the Multiproportional Problem. *Proc., 6th Meeting of the EURO Working Group on Transportation,* Gothenburg, Sweden, Sept. 9–11, 1998.
18. Wu, J. A Real-Time Origin-Destination Matrix Updating Algorithm for On-Line Applications. *Transportation Research B,* Vol. 31, No. 5, 1997, pp. 381–396.
19. Hall, M., D. Van Vliet, and L. Willumsen. SATURN—A Simulation-Assignment Model for the Evaluation of Traffic Management Schemes. *Traffic Engineering and Control,* No. 4, 1980.
20. Van Vliet, D. Improved Shortest Path Algorithm for Transport Networks. *Transportation Research,* Vol. 12, 1978, pp. 7–20.
21. Willumsen, L. G. Estimating Time-Dependent Trip Matrices from Traffic Counts. *Proc., 9th International Symposium on Transportation and Traffic Theory* (J. Volmüller and R. Hamerslag, eds.), VNU Science Press, Utrecht, 1984.
22. Taylor, N. B. CONTRAM5: An Enhanced Traffic Assignment Model. *TRRL Research Report 241.* Transport and Road Research Laboratory, Department of the Environment, Crowthorne, Berkshire, U.K., 1990.

Publication of this paper sponsored by Subcommittee on Network Equilibrium Modeling and the Committee on Transportation Network Modeling.