Comparison of path flow reassignment methods for dynamic user equilibrium traffic assignment based on mesoscopic simulation

M.P. Linares[1] and J. Barceló[1]

[1]Department of Statistics and Operations Research, Technical University of Catalonia Barcelona Tech
mari.paz.linares@upc.edu, jaume.barcelo@upc.edu

ABSTRACT

The iterative process proposed to solve the dynamic user equilibrium traffic assignment has two fundamental components: the dynamic network loading and the flow reassignment method. The reassignment component takes much less time and computational resources than the dynamic network loading; however, it has more direct influence to achieve the dynamic user equilibrium conditions. For this reason this paper is mainly concerned with path flow reassignment and how this affects solving dynamic user equilibrium. To investigate the performance of the reassignment, limitations of the method of successive averages are analyzed, paying special attention to the indiscriminately diversion of flow from used paths to the new found best path, and how this affects the convergence of later departure time intervals. Consequently, a new method of successive averages is developed including the improvements suggested. A method for paths flow reassignment that has been used in the literature, is compared with the new proposal embedding them in a dynamic traffic assignment scheme. For this purpose it is essential to incorporate a dynamic network loading model in order to complete the scheme: we have used the mesoscopic traffic simulation model included in Mezzo. The evaluation of these alternative flow reassignment algorithms has been computationally tested in the network of Södermalm (Stockholm). The results obtained show that, the proposed modification of the method of successive averages improves the observed bad convergence for latter departure time intervals.

Keywords: dynamic traffic assignment, dynamic user equilibrium, method of successive averages, mesoscopic traffic simulation.

INTRODUCTION

Dynamic traffic assignment (DTA) models for predicting dynamic user equilibrium (DUE) flows on urban traffic networks, are often solved by an algorithm that iterates between two main components until a convergence criterion is satisfied. These two components are the dynamic network loading and the path flow reassignment.

This paper works with the specific proposed DTA iterative scheme shown in FIGURE 1, which includes a time-dependent shortest path component that adds new paths, when it becomes necessary. Thus, the proposed path flow reassignment process must take into account this new path in the next path flow computation.

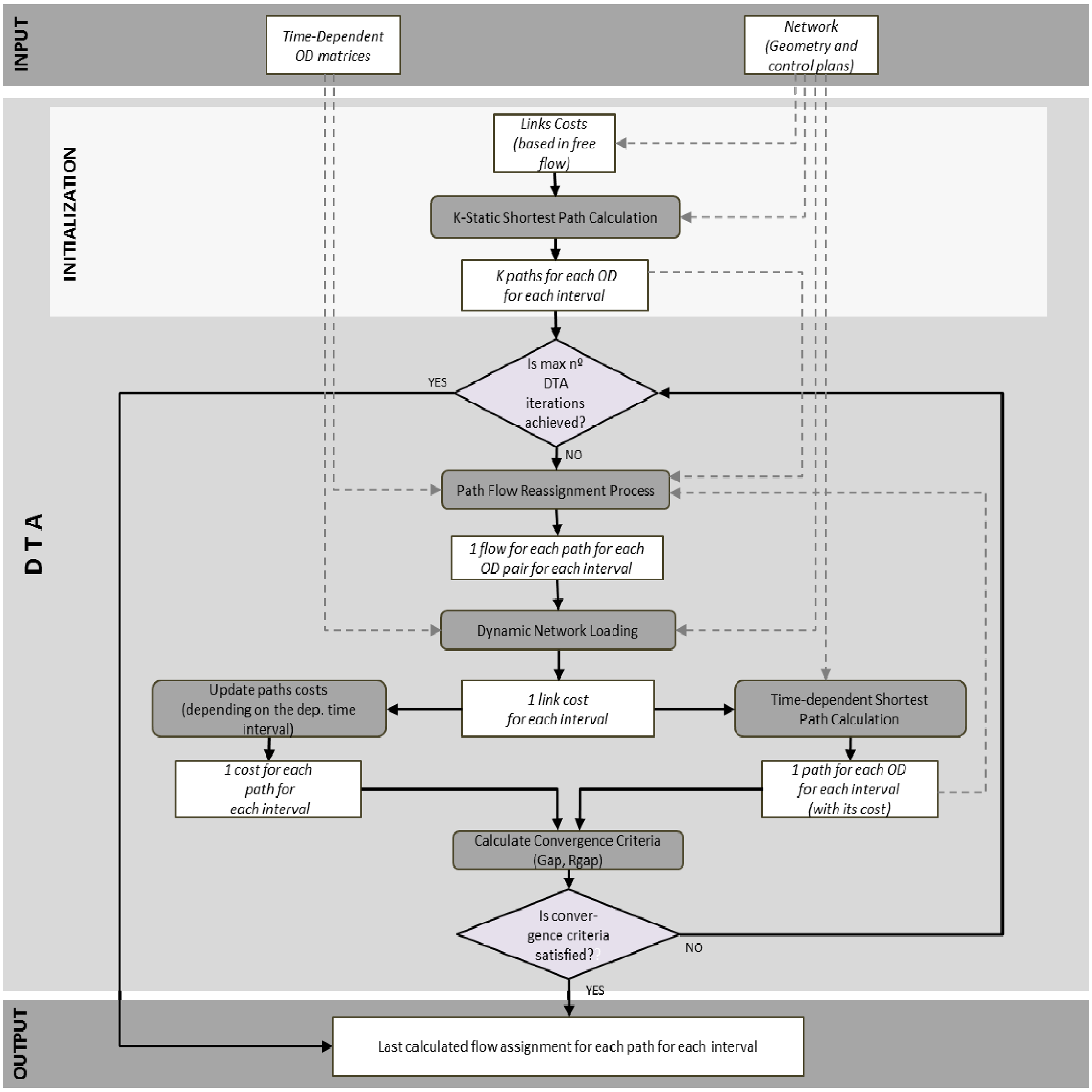

FIGURE 1 DTA scheme used.

It should be mentioned that although in general the flow reassignment takes much less time and computing resources than the dynamic network loading (over 95% of the total computational time, Carey and Ge [1]), it has more direct influence on the convergence speed of the solution (number of iterations), and also on whether the global process converges to a dynamic user equilibrium or not. Hence, this paper is mainly concerned with path flow reassignment and how this affects solving dynamic user equilibrium. In particular, among all the reassignment possibilities, it is focused on the method of successive averages.

This paper is structured as follows. The first section presents a brief literature review regarding the method of successive averages and some of its adaptations. Next section is dedicated to the presentation of a new flow reassignment method based on a MSA scheme. Finally, the computational tests performed are described beginning with the specifications about: the network loading method used to complete the DTA model, the study real network and the used demand data. The paper is completed with the presentation and discussion of the comparisons of the new algorithm with an alternative assignment method.

LITERATURE REVIEW OF THE METHOD OF SUCCESSIVE AVERAGES

The most widely used method for path flow reassignment component in a DTA scheme is the method of successive averages (MSA) that is the most simple an efficient. MSA was introduced by Robbins and Monro [2] for a quite different type of problem. The method was later used in transportation modeling (e.g. in Powell and Sheffi [3]) and has been widely used since then. In the present context MSA consists of removing a fraction of the flow (step size) from each of the currently used paths and adding this amount to the flow for the current shortest path for each OD pair. With the proper choice of the step size at each iteration, the method converges to the Wardrop's equilibrium solution in static traffic assignment (Sheffi [4]).

When applied to the dynamic assignment, the MSA needs a slight modification of the algorithm. Since in static assignment the path flows are time independent, the application of the averaging process on the path flows is equivalent to that on the link flows. However, in dynamic assignment the application of the averaging process on link flows will lead to erroneous results. Therefore, the averaging process should be applied on the path flows rather than the link flows as for the static assignment (Tong and Wong [5]).

When the assignment is dynamic in nature, the new path generated at any time interval of the current iteration (having no flow during previous iteration) raises the problem of flow distribution, as soon as it is no longer a shortest path in MSA. Tong and Wong [5] proposed a MSA scheme in order to overcome this problem.

In this MSA method and in other methods considered in this paper, the step size (λ_k) varies with the iteration number (k). Usually, the step size is considered the reciprocal of the iteration number $\left(\lambda_k = \frac{1}{k}\right)$. Other possibility would be choosing a predetermined fixed λ_k, and let it remain constant over all iterations. Carey and Ge [1] showed that the results obtained for MSA with constant step sizes $(1 > \lambda > 0)$ are in all cases worse than for MSA with a variable step size $\frac{1}{k}$.

The different iterative algorithms presented iterate till some selected convergence criterion is satisfied. The convergence criterion often used is the duality gap for any feasible solution of DUE, proposed by Janson [6]. The gap can be determined as:

$$\mathbf{Gap^k} = \frac{\sum_t \sum_{o,d,p \in P_{odt}^{k-1}} \mathbf{f_{odpt}^{k-1}} \cdot \left| \mathbf{c_{odpt}^k} - \mathbf{c_{ody_{odt}t}^k} \right|}{\sum_t \sum_{o,d} \mathbf{q_{odt}} \cdot \mathbf{c_{ody_{odt}t}^k}} \quad (1)$$

Where:

k is the iteration count.

t	is the departure time interval.
P_{odt}^{k}	is the set of paths from origin o to destination d entering to the network during the time interval t at iteration k.
f_{odpt}^{k}	is the flow assigned to the path p departing at time interval t at iteration k.
c_{odpt}	is the cost of the path p based on the actual travel times (tt_{odpt}) obtained at the dynamic network loading performed at the previous iteration $k-1$.
q_{odt}	is the demand from origin o to destination d entering to the network during the time interval t.
y_{odt}	is the time-dependent shortest path from origin o to destination d entering to the network during the time interval t.

Generally, the successive averages algorithm ends when:

- It gets an acceptable error $Gap^{k} < \varepsilon$, because this gap measures the deviation of the MSA solution from a true equilibrium solution. A 5% gap can be considered acceptable (Tong and Wong 2000).
- It reaches a maximum number of iterations: (K_{max}).

Mahut [7], [8] proposed another gap measure inspired from that used in static network equilibrium models that may be used for qualifying a given MSA solution. This is called the Relative Gap ($RGap$), that is the difference between the total travel cost experienced and the total travel cost that would have been experienced if all vehicles had the travel cost (over each interval) equal to that of the current shortest path. Hence,

$$RGap_{t}^{k} = \frac{\sum_{o,d,p\in P_{odt}^{k-1}} f_{odpt}^{k-1} \cdot \left| c_{odpt}^{k} - c_{ody_{odt}t}^{k} \right|}{\sum_{o,d} q_{odt} \cdot c_{ody_{odt}t}^{k}} \tag{2}$$

An observation detected on the behavior of the MSA algorithm is that the assignment for later departure time intervals is further away from the dynamic user equilibrium conditions than earlier departure time intervals. This may be due to vehicles entering later on the network may incur a high congestion, and the convergence is more difficult to achieve if the network is congested.

Mahut et al. [9] proposed a time-varying step-size heuristic, which gradually modifies the step-sizes applied to latter intervals, in order to amend this limitation. In the various tests performed on real networks the algorithm significantly accelerated convergence results.

A NEW FLOW REASSIGNMENT METHOD BASED ON MSA

Justification of the proposed flow reassignment method

The proposed iterative flow reassignment algorithm is based on a modification of the method of successive averages. The main objective of the proposed modification is try to overcome some of the limitations observed during the study of the state-of-the art about dynamic traffic assignment with MSA for the reassignment component. Among the main drawbacks of the method, this proposal pays attention to two of them which are summarized in the following.

The first observed limitation is that MSA requires the storage of all paths and its flow assignments at each iteration of the process. This can require a lot of memory space in order to perform the usual implementation of the algorithm. Furthermore, this drawback of the MSA tends to worse when the size of the network grows, or when the level of congestion is high. It is for this reason that most of the early works on this subject avoided the use of the MSA for large or congested networks.

The second problem is the standard flow reassignment used to divert traffic from the paths used in the last iteration to the current optimum paths. At each iteration of the process (or during some iterations), for each OD pair, for each time interval, the MSA method adds to the set of the used paths, a new one route, which is the path with a lower cost taking into account the cost of the links based on the travel times obtained in the last dynamic network loading. So, the objective of the method is to divert some flow from each used path to this new best path. The original MSA does it indiscriminately, and it extracts the same amount of flow from each used path, regardless of the paths costs, i.e. without taking notice if the path is the worst or only slightly worse than the optimal. This way of flow reassignment is not intuitive because the paths that have an appropriate costs should not suffer the same flow discharge that the paths that have high costs, from which it would intuitively removes flow to reduce congestion and consequently to reduce costs.

Attending the mentioned drawbacks of the MSA, the developed flow reassignment algorithm tries to improve the currently available options proposed in the literature to address these limitations, which sometimes solve one of the problems but not both. Taking into account the proposed solutions, a new MSA that combines some of these modifications with the addition of new ones is specified below.

New Flow Reassignment Modified MSA

As is raised in the previous section, one of the common problems that these methods present is the heavy computational load associated with their implementation as these, MSA needs to store at each iteration, the new best paths found for each OD, for each time interval. Mahut et al. [7],[8] proposed a solution to alleviate this problem which is based on the idea of limiting the number of alternative paths for each OD pair for each possible departure time interval.

This algorithm performs the reassignment of the flow in a different way depending on whether not yet reached the maximum number of paths (defined previously), or conversely, it is reached. In the first case, it must recalculate the minimum path using the links costs obtained from the performance of the last dynamic network loading. In this case, Mahut assigns the flow among all the possible paths (including the new one), equally. In the case of having reached the minimum number of paths, it is not need to recalculate any further shortest path, so that the set of paths remains stable until the end of the procedure. From this time, the flow is distributed among the possible paths using a classic MSA scheme.

In this case the problem about the paths storages is solved. However this solution does not take into account the different costs of each path when it reassigns the flow, i.e., the second MSA drawback still holds. This is why the method proposed in this paper uses the same idea of dividing the process into two parts, according to if the maximum number of paths for each OD pair for each time interval is reached or not; but, it incorporates a modification in the flow reassignment of each iteration, which is developed to eliminate the second MSA proposed problem. Therefore, a new algorithm that solves both conflicts simultaneously is proposed below.

In the classical version of the MSA method, the MSA parameter used to distribute the flow among the possible paths depends on the current iteration of the assignment process, and usually is equal to the inverse of the number of this iteration. In that case, as already mentioned, the diversion of flow to the best path from the rest of paths is done indiscriminately, without taking into account the cost of all the possible paths.

To overcome this limitation, Varia and Dhingra [10] proposed a modification in the procedure, using a new factor based on a logit distribution of demand flow according to instantaneous travel time on the corresponding paths. So, the reassignment explicitly takes into account the cost of the alternative paths when diverts the flow. Thus, the main part of the flow assigned to the new shortest path comes from the worst paths, i.e. from the paths with higher costs.

It is important to note here that the improvements in terms of performances allowed by the proposed algorithm are negligible with respect to those that could have been induced by a classical approach. This is because the logit factor, that Varia and Dhingra proposed, is based on instantaneous travel times rather than on the actual travel times.

The main idea of the proposed MSA modification is to complement the usual parameter of MSA in some specific parts of the proposed scheme trying to take advantage of the information of the previous dynamic network loading (the other main component of the DTA model). Thereby, when at certain iteration of the process a new shortest path is found, the proposed factor improves the method taking into account the cost of the alternative paths (all the paths belong to the set of paths used in the previous iteration) in the flow reassignment.

Looking at the commented proposal of Varia and Dhingra [10], this new factor (called in the following as diversion factor) is based on a logit distribution, but in this case according to the actual costs of alternative paths. The method considers the costs based on the links actual travel times obtained in the dynamic network loading performed in the previous iteration of the procedure. Thus, the expected improvements will be significant by comparing the results with those obtained in a classic MSA procedure.

The diversion factor $\left(\delta_{odpt}\right)$ is defined for each path p from origin o to destination d departing at time interval t as follows:

$$\delta_{odtp} = \frac{exp\left(-c_{odpt}\left(tt_{odpt}\right)\right)}{\sum_{p \in P_{odt}} exp\left(-c_{ocpt}\left(tt_{odpt}\right)\right)} \qquad (3)$$

where c_{odpt} is the cost of the path p based on the actual travel time (tt_{odpt}) of the path.

Finally, some ideas about the first iteration of the flow reassignment proposed method are commented here. At the initialization step, a static shortest path for each OD pair is calculated. Here, it is not necessary calculate time-dependent shortest path for each OD pair for each departure time interval, because the method bases the calculation on the free flow links travel times, and obviously all the travel times are the same for all the time intervals. Therefore, in the initialization a static shortest path algorithm is used and the same set of paths for each departure time interval is generated for each OD pair.

It is important to note that in this first step, more than one path is calculated for each OD pair. This is because if the algorithm starts with only one possible path for assign all the flow of each OD pair, the possibility of generate false congestion is very high. If it occurs, all the main links of the network can present congestion, and consequently very high costs. In the next step of the process, the algorithm finds a new shortest path that not use these congested links, so it can be going very far from the equilibrium solution, and it will need more iterations in order to converge. However, if the algorithm starts with a little set of paths for each OD pair and it assigns the flow proportional to each paths cost, then it needs less iterations to achieve the equilibrium. The number of initial paths (M) depends on the networks characteristics. The most convenient would be to test different options during the calibration network process.

In summary, an adaptation of the MSA method that combines the following two specified proposed solutions is presented:

1. Limit the maximum number of available paths for each OD pair for each departure time interval t, in order to reduce the computational storage needed in the original MSA.
2. Use a diversion factor based on actual travel times in the reassignment process in order to do more realistic reassignment flow among the alternative paths.

Global scheme of the proposed flow reassignment algorithm

In the following scheme, the specific proposed algorithm is shown:

1. Initialization $(k = 1)$

    a. Static shortest paths calculation based on costs according to free flow link travel times. All path sets P_{odt} are generated with the same number of path defined previously (M).
    b. Initial flow assignment (proportional to the paths costs):
        i. For all OD pair (o, d) for all departure time interval t for all paths $p \in P_{odt}$:

$$f_{odpt}^{k} = \frac{c_{odpt}}{\sum_{p \in P_{odt}} c_{odpt}} \cdot q_{odt} \qquad (4)$$

    c. Dynamic network loading to obtain an initial solution.
    d. Update iteration count $(k \coloneqq k + 1)$

2. Path Flow Reassignment

   For all OD pair (o, d) for all departure time interval t:

   - If the maximum number of paths is not reached $\left(\left|P_{odt}^{k-1}\right| < N_{odt}\right)$:

     a. Time-dependent shortest path (y_{odt}) calculation based on the links costs according to the actual link travel times obtained in the last dynamic network loading.
     b. Path Flow Reassignment:

        - If the shortest path is new $\left(y_{odt} \notin P_{odt}^{k-1}\right)$:
          i. Assign the flow $f_{odpt}^{k} \forall p \in P_{odt}^{k-1}$ and on y_{odt} following:

$$f_{odpt}^{k} = \begin{cases} \lambda_k \cdot q_{odt} & if\ p = y_{odt} \\ (1-\lambda_k) \cdot q_{odt} \cdot \delta_{odpt} & if\ p \neq y_{odt} \end{cases} \quad (5)$$

          ii. Update paths set: $P_{odt}^{k} = P_{odt}^{k-1} \cup y_{odt}$
          iii. Update number of paths.
        - Else $\left(y_{odt} \in P_{odt}^{k-1}\right)$
          i. Assign the flow $f_{odpt}^{k}\ \forall p \in P_{odt}^{k-1}$ following:

$$f_{odpt}^{k} = \begin{cases} \lambda_k \cdot q_{odt} + (1-\lambda_k) \cdot f_{odtp}^{k-1} & if\ p = y_{odt} \\ (1-\lambda_k) \cdot f_{odpt}^{k-1} & if\ p \neq y_{odt} \end{cases} \quad (6)$$

          ii. Update paths set: $P_{odt}^{k} = P_{odt}^{k-1}$

   - Else, the maximum number of paths is achieved $\left(\left|P_{odt}^{k-1}\right| \geq N_{odt}\right)$:

     a. Identify the shortest path y_{odt} among those already used P_{odt}^{k-1}.
     b. Path Flow Reassignment:

          i. Assign the flow $f_{odpt}^{k}\ \forall p \in P_{odt}^{k-1}$ following:

$$f_{odpt}^{k} = \begin{cases} \lambda_k \cdot q_{odt} + (1-\lambda_k) \cdot f_{odtp}^{k-1} & if\ p = y_{odt} \\ (1-\lambda_k) \cdot f_{odpt}^{k-1} & if\ p \neq y_{odt} \end{cases} \quad (7)$$

          ii. Update path set: $P_{odt}^{k} = P_{odt}^{k-1}$

3. Dynamic Network Loading

   a. Flow propagation using mesoscopic simulation.
   b. Update link travel times, and consequently its costs.

4. Convergence criteria
   - If the maximum number of iterations is reached $(k \geq K_{max})$ or the Rgap is satisfied ⇒ STOP
   - Else:

a. Update iteration count $(k := k + 1)$.
b. Go to Step 1.

Where:

M is the initial number of paths considered for each OD pair for each departure time interval t at initizalization step $(k = 1)$.

N_{odt} is the maximum number of paths considered from origin o to destination d entering to the network during the time interval t.

λ_k is the MSA parameter according to the corresponding iteration k.

δ_{odpt} is the diversion factor previously defined.

K_{max} is the maximum number of iterations considered in the procedure.

Depending of the different proposed values for the MSA parameter (λ_k) the method obtains some different results. In this paper the proposal is the standard $\lambda_k = \frac{1}{k}$, other possibilities have also been tested.

COMPUTATIONAL RESULTS FOR THE PROPOSED REASSIGNMENT FLOW METHOD

In the previous section, an iterative scheme for the resolution of the dynamic traffic assignment problem, which includes a new flow reassignment process based on a modification of the MSA algorithm, is presented. The objective now is to use a real example to carry out a primary performance test of the proposed approach.

Therefore, the results obtained with the new proposed algorithm are compared with the results of the path flow reassignment proposed by Mahut [7],[8]. With the aim of doing this, the two proposals are embedded in the DTA scheme presented above and executed for the study real network.

Moreover, the results obtained with the new proposed algorithm which limits the number of paths for each OD pair for each departure time interval are compared with the same proposal without this limitation. The main idea of this experiment is to show that this restraint not influences on the results, so, the method can improve its computational charge without the results are affected.

Employed dynamic network loading model: Mezzo

Though this paper is mainly concerned with path flow reassignment, and how this affects solving DUE, it is essential to include a dynamic network loading (DNL) model in order to complete the DUE algorithm. Several approaches are available to model traffic dynamics. In this case a mesoscopic simulation-based DNL proposed in Mezzo is selected because its flexible structure allows the relatively easy incorporation of it within the DTA framework.

Mezzo was developed by Burghout [11] at Royal Institute of Technology as the mesoscopic component of a hybrid mesoscopic-microscopic simulation model. Its structure is similar to the link-node queue-server model of DYNASMART, in that it uses speed/density relations for the determination of link-travel times of vehicles, in addition to stochastic node-servers to reproduce delays caused by interactions at nodes. In contrast to DYNASMART, the simulation is event-based (as in DYNAMEQ), and explicit mechanisms ensure the correct modeling of start-up shockwaves in case of queue dissipation, resulting in more realistic behavior of queue formation and dissipation over both time and space.

Example Data Set

A set of computational experiments was conducted with example network corresponding to the Södermalm district in central Stockholm (**FIGURE 2**), with the infrastructure corresponding to 2007. This network depicted in **FIGURE 3** has 1101 sections, 409 intersections, 168 centroids and 462 OD pairs.

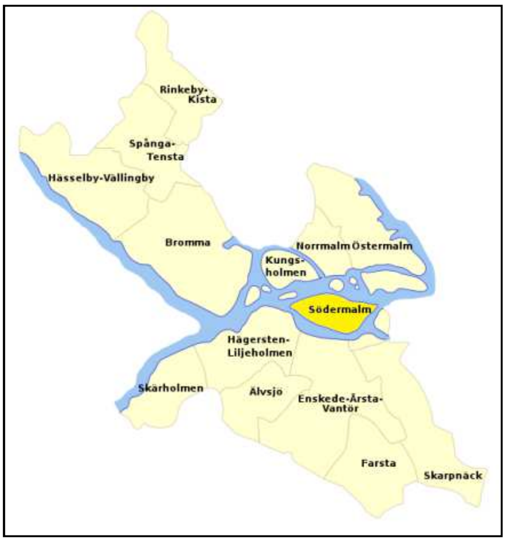

FIGURE 2 Södermalm district.

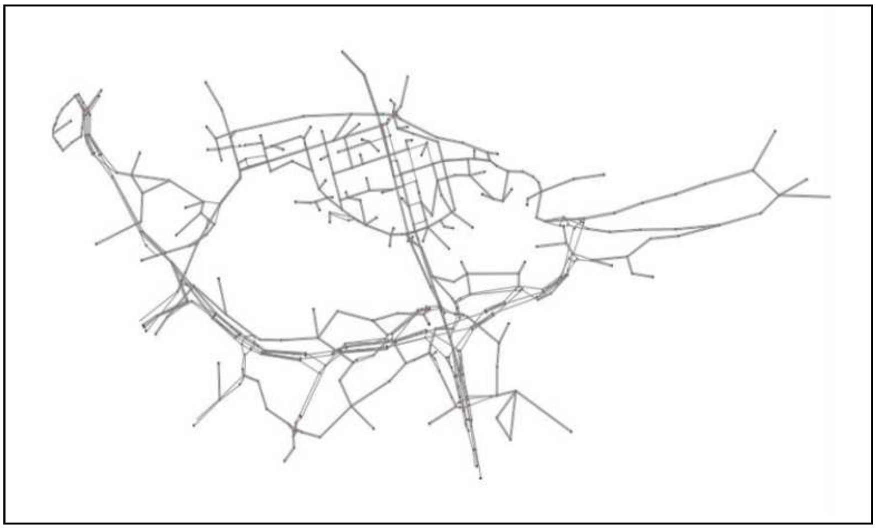

FIGURE 3 Södermalm Mezzo model.

The experiments executed used a synthetic demand that assigned to the Södermalm network the total flow of 34.451 trips. The demand was split into eight different time-dependent Origin-Destination matrices corresponding to eight different time slices of 15 min.

Experiments and results

The results for the DTA process that verify the performance and feasibility of the flow reassignment method proposed in the previous section are presented here. First, the Gap proposed by Janson [6] was calculated in order to have a global idea of the good performance of the process. Then, using the RGap proposed by Mahut [7],[8], the conclusions are tried to be refined taking into account the different departure time intervals in the analysis of the results.

Comparison of assignment algorithms

The method for paths flow reassignment proposed by Mahut [7],[8] was compared with the new proposed algorithm embedding them in the DUE scheme presented. Both algorithms have limitation about the maximum number of paths allowed for each OD pair for each departure time interval. In this case, the quantity of paths bound was considered five according to the network characteristics.

The two dynamic traffic assignment experiments proposed were run for a maximum number of iterations equal to 16.

TABLE 1 and **FIGURE 4** show the achieved gap after each iteration for both executed algorithms: the modified MSA presented in this paper and the Mahut proposal. Looking for these results, in terms of the global gap, it concludes that at the 16th iteration both reassignment methods have a similar convergence. But, the proposed method achieved the accepted as DUE gap (0.05) before than the literature method evaluated, so its rate of convergence is faster.

TABLE 1 Gap Obtained For the Compared Flow reassignment methods at each iteration

Iteration	GAP	
counter	Proposal	Mahut
1	0,3054160	0,3101380
2	0,1528720	0,1484130
3	0,0657945	0,0993414
4	0,0391474	0,0672048
5	0,0283231	0,0470400
6	0,0227178	0,0332407
7	0,0207230	0,0453495
8	0,0198850	0,0395041
9	0,0167493	0,0329077
10	0,0224624	0,0324397
11	0,0160405	0,0230915
12	0,0187119	0,0225669
13	0,0187224	0,0200237
14	0,0153077	0,0187239
15	0,0144372	0,0171095
16	0,0136779	0,0163962

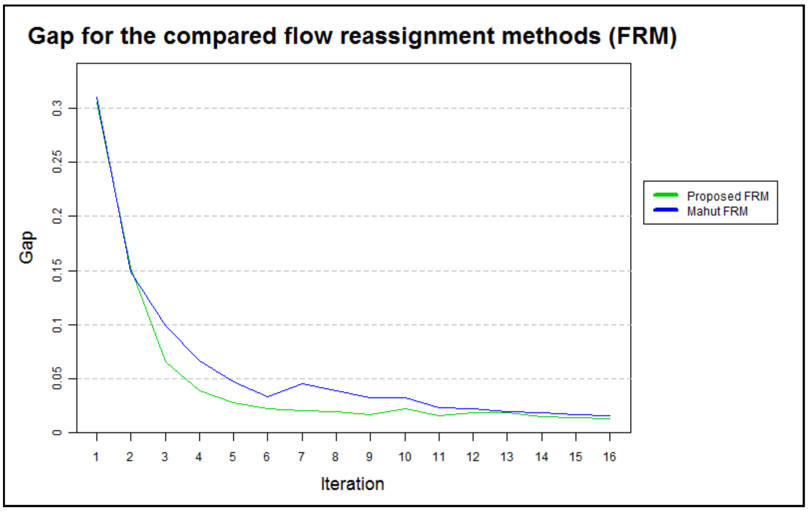

FIGURE 4 Gap achieved with the compared flow reassignment methods.

With regard to the Relative Gap,

TABLE 2 presents the results obtained with the experiment with limited number of paths for each OD pair for each time interval (5 paths) with the new modified MSA proposed. Also, it shows the results for the same experiment, changing the flow reassignment method to the literature proposed. **FIGURE 5** summarizes these results.

TABLE 2 Relative Gap Achieved at Each Iteration of the Proposed Flow Reassignment Method and of the Mahut Flow Reassignment Method, At Each Departure Time Interval

RELATIVE GAP - Proposed flow reassignment method								
Iteration	Departure Time Interval							
counter	1	2	3	4	5	6	7	8
1	0,0844837	0,0812983	0,1240780	0,2454320	0,2692640	0,4476490	0,4660990	0,4551260
2	0,0521314	0,0445898	0,0644136	0,1156120	0,1243090	0,1954270	0,2151100	0,2527650
3	0,0324426	0,0276975	0,0270475	0,0373833	0,0338631	0,0691494	0,0356514	0,1663880
4	0,0253015	0,0183146	0,0172331	0,0256336	0,0296968	0,0248377	0,0247900	0,0930225
5	0,0218558	0,0140509	0,0128190	0,0165564	0,0237415	0,0136726	0,0147007	0,0687174
6	0,0196410	0,0131893	0,0108767	0,0129996	0,0185174	0,0137777	0,0196190	0,0479542
7	0,0159828	0,0128305	0,0124464	0,0215115	0,0166212	0,0130779	0,0161906	0,0376912
8	0,0123801	0,0103181	0,0156504	0,0144584	0,0149094	0,0269740	0,0139225	0,0331674
9	0,0119414	0,0081769	0,0136452	0,0113601	0,0120479	0,0214657	0,0105725	0,0291701
10	0,0132278	0,0096469	0,0132686	0,0109323	0,0111448	0,0129372	0,0101660	0,0542691
11	0,0138756	0,0094142	0,0120026	0,0093153	0,0135797	0,0086774	0,0102966	0,0322857
12	0,0168406	0,0285791	0,0235303	0,0175544	0,0122738	0,0076347	0,0102541	0,0259762
13	0,0136321	0,0233804	0,0189639	0,0136860	0,0228927	0,0110117	0,0099898	0,0267182
14	0,0121259	0,0131847	0,0123031	0,0101216	0,0162933	0,0084944	0,0145335	0,0245239
15	0,0135455	0,0090775	0,0106272	0,0105591	0,0101026	0,0069397	0,0106729	0,0282554
16	0,0127287	0,0085243	0,0100627	0,0128847	0,0115704	0,0089182	0,0091582	0,0237124

RELATIVE GAP - Mahut flow reassignment method								
Iteration	Departure Time Interval							
counter	1	2	3	4	5	6	7	8
1	0,0850681	0,0796329	0,1155030	0,2396890	0,2572940	0,4105500	0,4787400	0,4968280
2	0,0447769	0,0484278	0,0547221	0,1065010	0,1293790	0,1603470	0,1730720	0,2761310
3	0,0289269	0,0279788	0,0319731	0,0716990	0,0852882	0,0972985	0,1153010	0,1941510
4	0,0259743	0,0274572	0,0223223	0,0431167	0,0523244	0,0643824	0,0721921	0,1331090
5	0,0266600	0,0332320	0,0258320	0,0281479	0,0314601	0,0358487	0,0386363	0,0934881
6	0,0252699	0,0146733	0,0199051	0,0231148	0,0239773	0,0185092	0,0255463	0,0677807
7	0,0217581	0,0224350	0,0166620	0,0232343	0,0388666	0,0389139	0,0365214	0,0999331
8	0,0187998	0,0200834	0,0237645	0,0275398	0,0353542	0,0351591	0,0254727	0,0801257
9	0,0175056	0,0181189	0,0217394	0,0230960	0,0240543	0,0266019	0,0320128	0,0633709
10	0,0162306	0,0150776	0,0194535	0,0181424	0,0191231	0,0207146	0,0250054	0,0743522
11	0,0139109	0,0118364	0,0149872	0,0159501	0,0139758	0,0150417	0,0219413	0,0473400
12	0,0117413	0,0081791	0,0112256	0,0161769	0,0131967	0,0132143	0,0208451	0,0510525
13	0,0116089	0,0069301	0,0099820	0,0210860	0,0103426	0,0183967	0,0132039	0,0412221
14	0,0118210	0,0070261	0,0093968	0,0185828	0,0083283	0,0134426	0,0144635	0,0399545
15	0,0108828	0,0091975	0,0100560	0,0149959	0,0073628	0,0126997	0,0196037	0,0325642
16	0,0084246	0,0076890	0,0095326	0,0156377	0,0107162	0,0194858	0,0182266	0,0268929

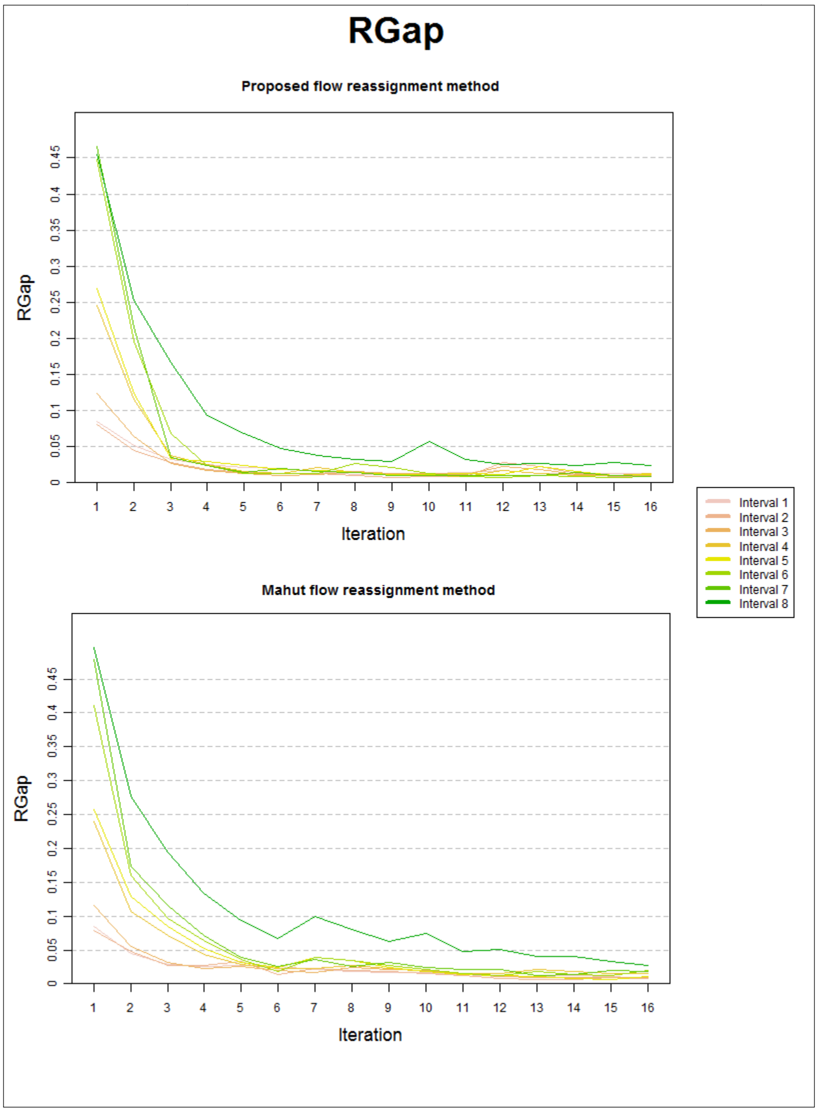

FIGURE 5 Rgap achieved at each departure time interval with the proposed flow reassignment method and with Mahut flow reassignment method.

The proposed method improves the results produced by the literature option. It can observe that the results for the last departure time interval are worse than the others departure intervals for both experiments. Anyway, it appears that the new reassignment flow method mitigates this effect achieving good RGap (less than 0.055) at 6th iteration while the literature option needs 11 iterations to achieve it.

Evaluation of the limitation about the maximum number of paths

In order to test the MSA adjustment about the limitation on the maximum number of available paths for each OD pair for each interval, the experimental design took into account two specific situations. While the first one did not consider this limitation, the second strictly followed the DTA scheme proposed with limited number of path approach. In this case, the quantity of paths bound was also considered five.

FIGURE 6 and **FIGURE 7** show Gap and Relative Gap (respectively) obtained in the experiments with limited or unlimited maximum number of paths.

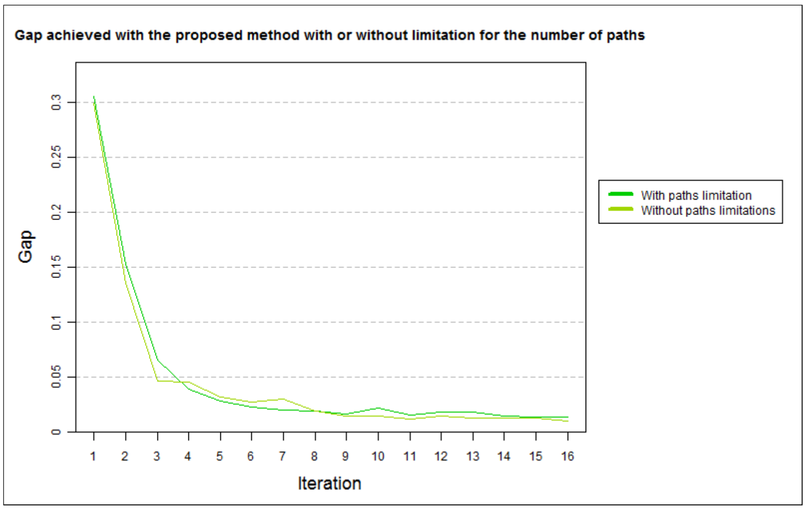

FIGURE 6 Gap achieved with the proposed flow reassignment method with or without the limitation about the maximum number of paths for each OD pair for each interval.

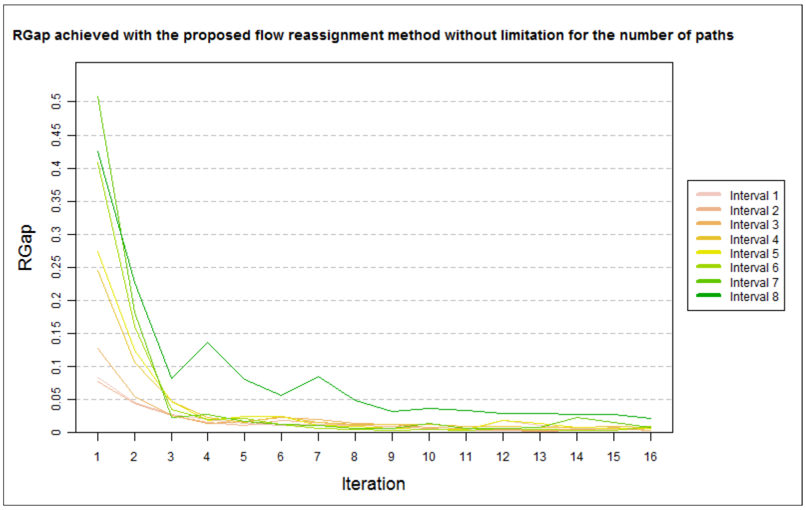

FIGURE 7 Relative Gap for each departure time interval achieved with the proposed flow reassignment method without the limitation about the maximum number of paths for each OD pair for each interval.

Simultaneously analyzing the presented results sets, it could say that the results of the experiments that have the limitation in the number of paths are a bit more stable than the results of the experiments that not restricted, but were not significant differences. So, the conclusion is that the limitation solution reduces the computational storage needed in the original MSA without reducing the good process performance.

CONCLUSIONS

A new modification of the method of the successive averages was developed to work around the improvement of the reassignment flow module of the DTA scheme proposed. The new method tried to overcome some detected MSA drawbacks combining solutions existing in the literature with new ideas. Specifically, the following modifications are done: the limitation of the maximum number of available paths for each OD pair for each time interval to reduce computational storage needed in the original MSA; and the addition of a new diversion factor (based on a logit distribution according to the actual travel times) in the reassignment process in order to do more realistic reassignment flow among the alternative paths.

In the global gap case, the quality of the new flow reassignment method solution was found to be independent of the use of limited number of paths for each OD pair for each departure time interval. Therefore, a reduction in the high computational storage, typical in the classical MSA, was achieved without losing the quality of the reassignment solution.

Comparing the new method with the literature option, in the case of global gap obtained for the experiments performed in the studied network, the new proposal was faster than the other evaluated option, and achieved the accepted gap of 0.05 some iterations before.

About the latter departure time intervals, their bad convergence was clearly observed in the results of all the experiments about the Relative Gap performed in Södermalm network. The proposed method mitigated this effect with the use of the diversion factor which enhances the flow reassignment among the alternative paths, without requiring other sophisticated solutions like the commented time-varying step-size adjustment. Besides, this new proposal achieved good RGap results (less than 0.055) in about half number of iterations than the other tested method.

ACKNOWLEDGEMENTS

The research reported in this paper has been funded by projects MITRA (TRA2009-14270 (subprogram MODAL, FEDER Co funded)) and In4Mo (TSI-020100-2010-690) of the Spanish R+D National Programs and has benefited from participation in EU COST Action TU0903 MULTITUDE. Especial thanks to Dr. Wilco Burghout for making available MEZZO and providing all kind of support.

REFERENCES

[1] Carey, M., Ge, Y.E. Comparison of methods for path flow reassignment for dynamic user equilibrium. *Networks and Spatial Economics*, 201l, pp. 1-40.
[2] Robbins, H., Monro, S., 1951. A stochastic approximation method. *The Annals of Mathematical Statistics* 22, 400-407
[3] Sheffi Y, Powell WB (1982) An algorithm for the equilibrium assignment problem with random link times. Networks 12(2):191–207
[4] Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall, Englewood Cliffs, NJ
[5] Tong, C.O., Wong, S.C. A predictive dynamic traffic assignment model in congested capacity-constrained road networks. Transportations Research Part B, Vol. 34, 2000, pp. 625-644.

[6] Janson, B.N. Dynamic traffic assignment for urban road networks. *Transportation Research Part B*, Vol.25, 1991, pp.143-161.
[7] Mahut, M., Florian, and M., Tremblay, N. Space-Time Queues and Dynamic Traffic Assignment: A Model, Algorithm and Applications. Presented at 82nd Annual Meeting of the Transportation Research Board, Washington, D.C., 2003.
[8] Mahut, M. et al. Calibration and Application of a Simulation-Based Dynamic Traffic Assignment Model. Presented at 83rd Annual Meeting of the Transportation Research Board, Washington, D.C., 2004. In *Transportation Research Record: Journal of the transportation Research Board*, No. 1876, Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 101-111.
[9] Mahut, M., Florian, M., Tremblay, N. Comparison of assignment methods for simulation-based dynamic-equilibrium traffic assignment. Presented at 87th Annual Meeting of the Transportation Research Board, Washington, D.C., 2008.
[10] Varia, H. R., Dhingra, S.L. Dynamic User Equilibrium Traffic Assignment on congested Multidestination Network. *Journal of Transportation Engineering*, Vol. 130, No 2, 2004, pp. 211-221.
[11] Burghout, W. Hybrid Microscopic-Mesoscopic Traffic Simulation. PhD thesis, Department of Infrastructure, Division of Transportation and Logistics, Royal Institute of Technology , Stockholm, Sweden, 2004.