Proceedings of the 8th International
IEEE Conference on Intelligent Transportation Systems
Vienna, Austria, September 13-16, 2005

Dynamic Equilibrium Assignment with Microscopic Traffic Simulation

Henry X. Liu, Wenteng Ma, Jeff X. Ban, Pitu Mirchandani

Abstract—**This paper presents a hybrid dynamic traffic assignment model by integrating the analytical dynamic user equilibrium technique and the micro-simulator Paramics. The time-dependent path flows are generated by the analytical dynamic user equilibrium model, while the network attributes such as link and path travel times are simulated by the micro-simulator. In particular, a path-based assignment method is developed in this paper to replace the off-the-shelf link-based assignment method in Paramics. To assure the convergence of the proposed model, the method of successive averages is applied to update dynamic path flows from iteration to iteration. Numerical examples are also provided in this paper to test the performance of the proposed hybrid model and the solution algorithm.**

I. INTRODUCTION

The deployment of intelligent transportation systems (ITS) must be assisted by suitable tools to conduct feasibility studies and evaluate expected impacts. In this context, traffic assignment, which is the last stage of the conventional four-step transportation planning process, can be used to help public authorities to test the impacts of various transportation policies and ITS deployments. In traffic assignment models, a set of rules and principles towards certain objectives (such as user equilibrium) is used to load a fixed or varied trip matrix onto the network, thus generate a set of link or path flows [1]. In particular, the dynamic traffic assignment (DTA), especially the dynamic user equilibrium (DUE), is an important tool in terms of modeling various applications related to ITS [2, 3, 4].

There are two categories of methods that arise in the study of DTA: analytical modeling methods and simulation modeling methods. The analytical-based models derive the optimality conditions from preset driver behavior principles such as utility maximization. So the sensitivity analysis for different scenarios is easier to perform since the procedure is usually less time consuming. Extensive research has been done in this area with various functional forms for cost-flow relationships. However, there are still some drawbacks to most of the analytical models. For example, most analytical models assume that the vehicle has zero length. So the spill-back of congestion can not be represented correctly. Moreover, the variety of the vehicle types and stochasticity of driver behaviors can not be captured properly in most of the analytical models. Therefore simulation methods have been suggested for years to overcome these problems.

The simulation-based models can represent the traffic dynamics in more detail and allow studies on the influence of the information system in the context of ITS, such as testing different traffic control strategies. Especially in recent years, the improvements in software, hardware and other techniques (like the distributed simulation [5]) have made possible to simulate microscopically real world networks with relatively large size and acceptable running time. The increasing of the simulation speed also insures that the multiple-run technique could be applied to decrease the uncertainty of the final simulation results. Thus simulation modeling becomes an increasingly popular and effective tool for analyzing transportation problems, which are not amendable to study by other means. In the transportation simulation field, there is some general agreement that micro-simulation, i.e., a computational resolution down to the level of individual travelers, is now a viable alternative and might be the only answer to a wide variety of problems. However, the drawbacks of simulation models are that the number of attributes involved is normally large and the properties of the solutions still remain uncertain. In particular, most of the simulation packages can not perform the convergence analysis and the solutions obtained could be far from the desired optimal ones.

Therefore, there is a need to combine the analytical and simulation-based approaches to integrate the merits of both methodologies. The goal is to generate a hybrid modeling technique so that both the mathematical soundness and the detailed traffic dynamics can be captured properly. Some previous investigations have been done in this area. Nagel et al. reported the TRANSIMS computational experiments for large scale DTA problem in the context of a Dallas scenario in 2000 [6]. The authors also investigated issues of uniqueness, variability, robustness and validation. Regarding uniqueness, despite some cautionary notes from a theoretical point of view, there was no indication of “meta-stable" states for the iterations. The variation of the simulation of a given plan by just changing the random seed was considerable. The results from three different micro-simulations under the same iteration scenario were tested for the robustness of the

Manuscript received March 1, 2005.

Henry Liu is with the Utah State University, Logan, UT 84322, USA (phone: 435-797-8289; fax: 435-797-1185; email: xliu@cc.usu.edu).

Wenteng Ma is with the Utah State University, Logan, UT 84322, USA (email: wenteng@cc.usu.edu).

Jeff X. Ban is with the Utah State University, Logan, UT 84322, USA (email: xban@cc.usu.edu).

Pitu Mirchandani is with the University of Arizona, Tucson, AZ 85721, USA (e-mail: pitu@sie.arizona.edu).

results. Some encouraging results were shown, especially when compared to those by the traditional assignment methods.

In 2002, Barcelo and Casas described a heuristic approach to DTA in which two alternative analytical components are used to determine the path flow rates, one based on a stochastic route choice method, and another one based on an approximation to DUE conditions; while the network loading was done by a microscopic simulation model, AIMSUN. The case study showed a fairly good agreement between the real world scenario and the simulation results [7]. In 2003, Mirchandani et al. proposed an iterated route-based CORSIM simulation model [8], where drivers' experience in one period was used as the input to the method of successive averages (MSA) to provide traffic assignment for the next period. To test the validity of the approach, traffic assignment was based on the assumption that travelers choose routes that will minimize their experienced travel times in the simulation. Furthermore, the network loading was examined using a path-based CORSIM simulation package. Analysis of the results from the assignment-simulation model showed that the used routes for each OD pair have nearly equal travel times and the assignment-simulation process tends to converge to the traffic equilibrium as the number of periods increases [8].

In this paper, a hybrid DTA model is developed by integrating the analytical DUE method and the micro-simulator Paramics. The analytical DUE determines the time-dependent route flows and ensures the overall convergence of the hybrid model. The micro-simulator, on the other hand, generates the attributes of the network such as link travel times and flows, based on the route flows from the analytical model – the so-called network loading procedure. Since off-the-shelf Paramics can only provide link-based assignments, no path information during the traffic assignment process can be stored or evaluated. This will bring difficulty for our hybrid framework since the input to Paramics is the time-dependent path flow based on which Paramics should be able to perform the network loading. Moreover, in some ATMS applications, particularly those related to the route diversion, partial or full path information is usually needed, which makes the path-based assignment in Paramics "a must" in order to apply Paramics for the analysis of these applications. To overcome this difficulty, a path-based assignment model is developed in this paper using Paramics application programming interface (API) functions.

The resulting hybrid model integrates the advantages of the analytical traffic assignment method and the good properties of the simulation tools. Some key techniques, e.g. route choice plug-ins, the turn penalty consideration, and the method of successive averages, are also developed in the model. The Paramics V4 is selected as the demonstration simulation tool. A real grid network in Tucson, Arizona, is designed to test the performance of the model. The results show that the model converges to the DUE state as expected.

This paper is organized as follows. In section II, a hybrid DTA model is presented by integrating an analytical DUE model and Paramics. Particularly, the time-dependent shortest path search is described for solving the analytical DUE model. The path-based assignment method is developed in Paramics for the network loading purpose. MSA is also introduced in this section to update path flows and guarantee the convergence of the algorithm. Section III provides numerical examples, based on the campus network of the University of Arizona, to test the proposed hybrid model. Finally, concluding remarks and future research directions are given in section IV.

II. PROPOSED METHODOLOGY

A framework for the DTA model can be depicted in Figure 1. In this study, since we assume the dynamic OD matrix is given, two major components in this framework needs to be resolved, namely, the dynamic route choice and the dynamic network loading.

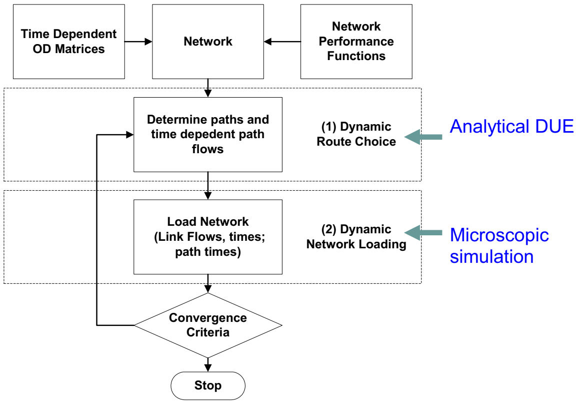

Figure 1 Prototype of the DUE model.

In our proposed hybrid model, the dynamic route choice is modeled using the analytical DUE method based on the link travel time information obtained from the network loading model. Meanwhile, the dynamic network loading is performed by Paramics, a microscopic simulator, using the time-dependent path flows generated by the analytical DUE. In actual implementations, these two models will run alternatively and iteratively until certain convergence criterion is satisfied.

A. *Analytical DUE Model*

As an extension to the static user equilibrium by Wardrop [9], the DUE condition can be expressed as follows [2]:

If, for each OD pair at each instant of time, the actual travel times experienced by travelers departing at the same time are equal and minimal, the dynamic traffic flow over the network is in a travel-time-based dynamic user equilibrium (DUE) state.

Then analytically, DUE can be formulated as a variational inequality (VI) as follows.

$$\sum_{t}\sum_{rs}\sum_{p}\eta_p^{rs*}(t)\left[f_p^{rs}(t)-f_p^{rs*}(t)\right]dt \ge 0 \qquad (1)$$

$s.t.$ dynamic network constraints

where $\eta_p^{rs}(t)$ and $f_p^{rs}(t)$ denote, respectively, the path travel time and path flow of path p between origin-destination (OD) pair rs at time interval t, and $\eta_p^{rs*}(t)$ and $f_p^{rs*}(t)$ the optimal solutions in terms of time-dependent path travel times and path flows, respectively. Further, t represents the discrete time interval during the entire study period. Note that $\eta_p^{rs}(t)$ is the actual experienced path travel time, implying that we are studying the ideal case DUE in this paper instead of the instantaneous case [2]. Note also that the dynamic network constraints in equation (1) are well-defined for the VI formulation in (1), as shown in [2].

Traditionally, a pure analytical DUE model will normally compute the path travel time $\eta_p^{rs}(t)$ as a summation of corresponding link travel times, where the later are usually represented by certain volume-delay functions. Although these functions provide nice convergence properties to the DUE model, they have certain limitations. Especially, they take a simplistic view of congestion and do not incorporate network or traffic characteristics such as intersection control strategies, start-up loss times and vehicle headways, pedestrian traffic, and different driver types and modes of transport, etc. In order to overcome these drawbacks, the analytical DUE model in this paper utilizes the link and path travel times evaluated from the latest run of the network loading. In other words, while solving the analytical DUE, we temporarily fix the link and path travel times using the results from Paramics. The rationality behind this is that drivers can experience the current prevailing link and path travel times and then try to adjust their route choice based on their newest experience.

Consequently, the VI based DUE formulation in (1) will reduce to a time-dependent shortest path search for each OD pair at each time instant [8]. In this paper, we implement the time-dependent shortest path algorithm based on the Dijkstra's method [10]. The original Dijkstra's algorithm is a static one and revisions have been made to accommodate the time-dependent search. Another major modification is the associated turning penalty for the link cost. At intersections or junctions of urban networks, different turning movements have different travel costs. For example, the un-protected left turns at a signalized intersection usually need to yield to the through movements. As a result, left turn vehicles normally have higher travel costs than through vehicles. This additional cost associated with the turning movement is the so-called turning penalty.

In Paramics, the turning penalties are associated with the downstream links, which will cause problems when applying traditional shortest path algorithms, like the Dijkstra's method. The reason is in every loop, the minimum cost from the origin node to the current labeled node is not fixed due to the fact that the next turn is unknown at this stage. As shown in Figure 2a, C_{ij} denotes the link travel cost (including the turning penalty) for link *i->j*, C_{ij}^0 the average vehicle travel time for *i->j*, and C_{ij}^T the turning penalty for the same link. Apparently, during the shortest path search, when link *i->j* is evaluated, it is unknown that which link is going to be chosen next at node *j*. This is why the turning penalty C_{ij}^T can not be determined when link *i->j* is processed.

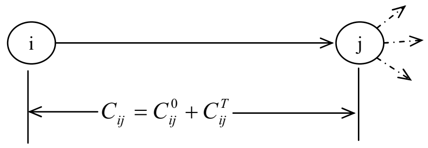

Figure 2a Illustration of the turning penalty for link i->j

To solve this problem, we introduce a dummy node for each link, as shown in Figure 2b. Then the cost of each link consists of two parts: vehicle travel time and the turning penalty. Notice that intuitively, as shown in Figure 2b, C_{ij} should consist of its travel time C_{ij}^0 and the turning penalty from its own dummy node $\hat{j}$ to its ending node j $C_{\hat{j}j}^T$. However, due to the aforementioned reason, we define the travel cost for link *i->j* in a slightly different way as follows.

$$C_{ij} = C_{\hat{i}i}^T + C_{ij}^0 \cdot \qquad (2)$$

In other words, the link cost is the summation of its free flow travel time and the turning penalty from the processor node to the starting node of the link ($C_{\hat{i}i}^T$ as shown in Figure 2b). Since we know the route from the origin node to the ending node *j* of current link, the link cost defined in equation (2) is known when link *i->j* is processed. Note that since the link costs for all links in the network are defined in a consistent way as in equation (2), the shortest path search results should be the same as those by the traditional definition (e.g., the one in Figure 2a).

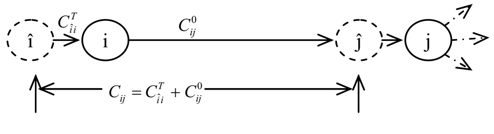

Figure 2b Redefinition of the link cost

The time-dependent shortest path search generates the path with the minimum travel cost between each OD pair at each time interval. Then an all or nothing (AON) assignment is performed to generate the auxiliary dynamic path flows.

At each time interval, the AON assignment will basically assign all trips of a given OD pair to a single path with the minimum travel cost. The all-or-nothing path flows will be used later in the MSA update.

B. Method of Successive Average (MSA)

The dynamic path flows generated by the analytical DUE model are actually auxiliary path flows by temporarily fixing the link cost for each link at each time interval. In order to insure the convergence, we need certain averaging scheme to combine the auxiliary path flows and the current assigned path flows. For this purpose, MSA has been proved to be effective in many cases. Assume the assigned dynamic path flow is $f_{p,rs}^{n-1}(t), \forall p, rs, t$ at iteration *n-1* and $\hat{f}_{p,rs}(t), \forall p, rs, t$ is the auxiliary dynamic path flow generated by the analytical DUE model for current iteration. Then for next iteration, i.e. iteration *n*, the updated dynamic path flow can be computed as follows.

$$f_{p,rs}^{n}(t) = (1-\theta) f_{p,rs}^{n-1}(t) + \theta \hat{f}_{p,rs}(t), \tag{3}$$

where $\theta = 1/n$ is the MSA step size.

Notice that since flow conservation holds for both current path flow $f_{p,rs}^{n-1}(t), \forall p, rs, t$ and the auxiliary path flow $\hat{f}_{p,rs}(t), \forall p, rs, t$, equation (3) guarantees that the flow conservation also holds for the newly updated path flow $f_{p,rs}^{n}(t), \forall p, rs, t$ which is a convex combination of current flow and the auxiliary flow. Obviously, the MSA update is operated directly on the path flow space, which requires a set of used paths to be stored for each OD pair at each time interval. For large scale problems, the number of used paths could be many.

C. Path-based Dynamic Network Loading in Paramics

The dynamic network loading takes the updated dynamic path flows from the MSA step to evaluate and generate the dynamic attributes of the network such as dynamic link flows, link travel times, and path travel times. In this paper, the micro-simulator Paramics is adopted for performing the dynamic network loading.

Paramics is a suite of microscopic simulation tools used to model the movements and behaviors of individual vehicles on urban and highway road networks [11]. It simulates the components of traffic flows and congestion, and presents its graphical animation output simultaneously for the traffic management and road network design. Due to its detailed and relatively accurate descriptions of traffic dynamics and network characteristics, Paramics has been widely applied in various traffic related modeling and analysis tasks [12,13]. However, off-the-shelf Paramics can only provide link-based assignments and thus can not be used directly as a network loading tool which needs to operate on the path flows. Fortunately, Paramics provides APIs to allow users to customize many features of the underlying simulation. In general, API functions can be used to override and extend the default functionalities or logic, and obtain and set the simulation parameters of Paramics. Therefore, in this study, we develop a set of plug-in modules using API functions to interface with Paramics. In particular, we replace the default link-based simulation in Paramics by the path-based simulation to develop the dynamic network loading procedure needed in our proposed hybrid model.

The mechanism for developing the path-based assignment in Paramics is as follows. Firstly, the dynamic path flows generated by the MSA step will be received by the route assignment plug-ins. Then every time step when there is a new vehicle released, the route assignment plug-ins will automatically associate a network path to this vehicle. Hence, the movement or link transfer of the newly released vehicle will be determined by Paramics simulation based on its associated path. Combined with other plug-ins, such as link performance data collection, Paramics can produce the link costs and further feedback them to the analytical DUE model.

D. Convergence Criterion

To check the convergence of the hybrid model, the relative gap defined in equation 4 is selected as the convergence criterion.

$$R_{gap} = \frac{\sum_{t} \sum_{\forall r,s} \sum_{p \in P^{rs}(t)} f_p^{rs}(t) \cdot [\eta_p^{rs}(t) - \pi^{rs}(t)]}{\sum_{t} \sum_{\forall r,s} d^{rs}(t) \cdot \pi^{rs}(t)}, \tag{4}$$

where $\pi^{rs}(t)$ is the actual and minimum travel cost between OD pair *rs* at time interval *t*, $d^{rs}(t)$ the demand from origin *r* to destination *s* departing *r* at time *t*, and $P^{rs}(t)$ the path set between O-D pair *rs* at time interval *t*. Obviously, if R_{gap} goes to zero, perfect equilibrium condition will be obtained. In practice, however, it is hard to get such exact equilibrium. Normally certain stopping threshold, e.g. 1.0e-4, will be pre-defined to assure that the algorithm can converge to certain level of accuracy.

E. Flowchart of the Solution Algorithm

Figure 3 shows the flowchart of the solution algorithm for the proposed hybrid model. It starts with loading the network information, O-D matrix, and the initial link costs into the network. Note that the initial link costs are set as the free flow travel time pre-generated by Paramics. Meanwhile, it sets the iteration number *n* = 1. Then the Dijkstra's algorithm is invoked to find the time-dependent shortest paths of all O-D pairs with the current link cost information generated from Paramics. After the shortest paths are found, an AON assignment is performed to obtain the auxiliary dynamic path flows. Next the MSA step uses the auxiliary path flows and averages them with current assigned path flows to generate the updated path flows. These updated path flows are then input to the path-based assignment model in Paramics to produce the dynamic network attributes, including the dynamic link costs, link flows, and path flows. The link costs will then be fed into the time-dependent shortest path routine. Such an iterative algorithm will run repeatedly until the convergence criterion is met, which is assured by the MSA for most of the times.

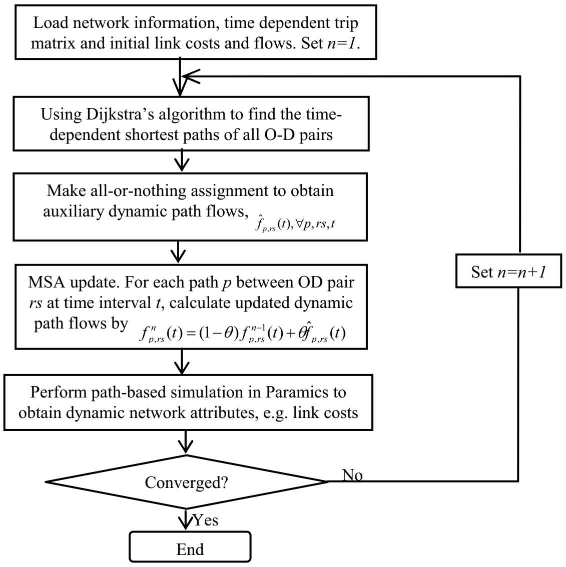

FIGURE 3 Flowchart of the proposed model.

III. COMPUTATIONAL RESULTS

In this section, an example network based on the University of Arizona campus at Tucson, Arizona, was built to test the performance of the proposed hybrid traffic assignment-microscopic simulation model. The micro-simulator we chose is the Paramics V4. This version of Paramics has more structured, easier to use, and inherently safer APIs which can facilitate significantly our model development.

The network shown in Figure 4 is a well calibrated gird network of the campus of the University of Arizona. It is coded by 29 O-D zones and 21 actuated signalized intersections in Paramics Modeler Version 4.21. The length of the east-west direction is about 4 miles, and that of the north-south direction is about 2 miles. Two-hour demands are assigned in the network to represent the demand pattern in rush hours. The demand pattern is shown as Figure 5. Testing is carried on a desktop with one 2.8 GHz CPU processor and 1 GB RAM, and the operation system is Windows XP Professional.

FIGURE 4 Tucson grid network.

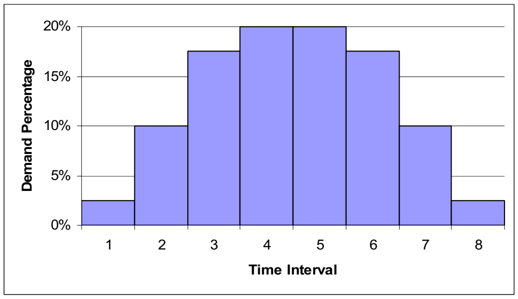

FIGURE 5 Demand pattern.

Since the total simulation time is 2 hours and each time interval in the DUE model is set to 1 minute, there are total 120 time intervals. The convergence criteria used in this paper is $R_{gap} = 10^{-4}$ and the maximum iteration is 60. Figure 6 shows the convergence curve of R_{gap} value vs. the iteration number. It is obviously that the relative gap drops very quickly in the first several iterations, from around 1.1% to near 0.1%, and after that the R_{gap} starts to oscillate. This oscillation is largely due to the stochastic nature of the microscopic simulation model. The algorithm converges after 50 iterations.

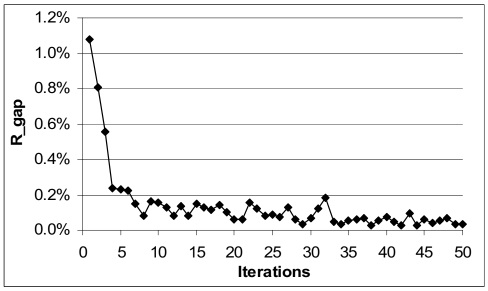

FIGURE 6 R_{ga} Convergence Curve.

A specific O-D pair from zone 5 to zone 27 is selected for analysis. There are totally 6 different used route between this OD pair. Figure 7 shows the route travel time pattern for all 6 used routes from zone 5 to zone 27 over the whole study period. We can see that it has the similar pattern as the demand pattern in Figure 5. Furthermore, the travel time differences among used paths are within 5%, which means the 6 used routes have nearly equal travel times. This indicates that the DUE state is approximately achieved.

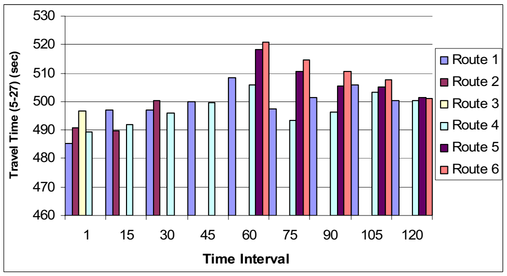

FIGURE 7 Route travel time pattern.

What shows in Figure 8 is the improvement in link travel time at equilibrium. One particularly selected link (24->11) is congested after 30 minutes of simulation in the initial iteration; while it is quite smooth and no congestion occurred during the last iteration.

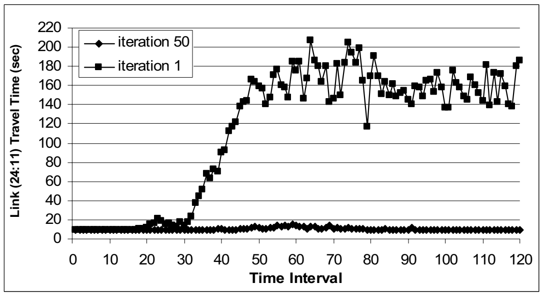

FIGURE 8 Link travel time improvement.

The detailed pattern of the travel times for link 24->11 for the last iteration is further depicted in Figure 9. Clearly, it is also similar to the demand pattern shown in Figure 5.

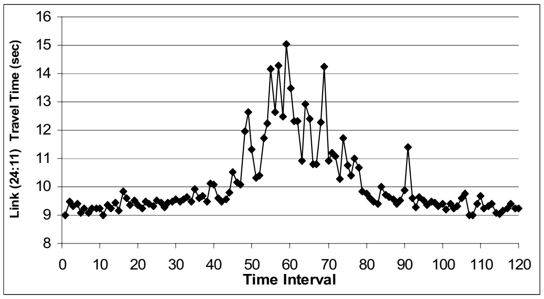

FIGURE 9 Link travel time pattern.

IV. CONCLUSIONS AND FUTURE RESEARCH

We proposed a hybrid DTA model with embedded microscopic simulation in this paper. The model is comprised by two parts: the analytical DUE model and the micro-simulation tool. The method of successive averages was used to update path flows and assure the overall convergence of the hybrid model. The Dijkstra's algorithm was adopted in this paper to perform the time-dependent shortest path search to solve the analytical DUE model. The proposed model also considered the turning penalty associated with the link cost in the shortest path search. The microscopic simulation software, Paramics V4 instead of the analytical cost-flow functions, was used to compute the link cost or travel time, which is more realistic and accurate. A real grid network at Tucson, Arizona, was coded to test the performance of the proposed model. The results showed that all the used routes for all OD pairs had nearly equal and minimum costs, implying the model achieved its equilibrium state after the iterative runs.

Comparing with the traditional DTA models, embedding the micro-simulator in DTA could 1) more accurately represent traffic dynamics such as spill-back of congestion, 2) properly model network changes such as work zone and incident, 3) have the capability to incorporate various ITS strategies, such as traffic signals, ramp metering, and variable message signs (VMS), and 4) be professionally supported since most micro-simulators are commercially available software packages.

For future research, more plausible analytical assignment approaches can be adopted for generating the time dependent path flows. Some of the candidates could be the cell transmission model (CTM) by Daganzo [14, 15] and the kinematic wave model by Jin [16]. Next, more sophisticated MSA approaches can be applied in the proposed hybrid framework to achieve better convergence [8]. The model proposed in this paper is path-based and path information needs to be explicitly stored for each OD pair at each time interval. This could bring storage and computation difficulty for large scale problems. Therefore, developing link-based hybrid models is another future research direction. Finally, more future studies need to be done to evaluate the proposed model and solution approach, in particular on applications related to advanced traffic management systems (ATMS).

REFERENCES

[1] Ortuzar, J. de D. and Willumsen, L. G. Modelling Transportation, Third Edition, John Wiley & Sons Inc., 2002.

[2] Ran, B. and Boyce, D., Modeling Dynamic Transportation Networks, Springer-Verlag, 1996.

[3] Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H.N., and Mishalani, R., 2002. Real time simulation of traffic demand-supply interactions within DynaMIT, in M. Gendreau and P. Marcotte (eds), Transportation and Network Analysis: Current Trends, Kluwver Academic Press.

[4] Mahmassani, H.S., 2001. Dynamic network traffic assignment and simulation methodology for advanced system management applications. Networks and Spatial Economics Vol. 1, 267-292.

[5] Liu, H., Ma, W. and Jayakrishnan, R. Distributed Modeling Framework for Large-scale Microscopic Traffic Simulation. Proc. of 84th Annual Meeting of the Transportation Research Board (CD-ROM), 2005.

[6] Nagel, K., Rickert, M., Simon, P. M., and Pieck, M. The Dynamics of Iterated Transportation Simulations. Available: http://maynard.tsasa.lanl.gov/PDF_Files/LAUR98-2168.pdf.

[7] Barcelo, J. and Casas, J. Heuristic Dynamic Assignment based on Microscopic Simulation. 9th Meeting of the Euro Working Group on Transportation, Bari, Italy, June 2002.

[8] Mirchandani, P., Syal, R. and Lucas, D. Traffic Assignment Using Iterated Route-based Simulation. Presented at the 82st Annual Meeting of the Transportation Research Board (CD-ROM), 2003.

[9] Wardrop, J. G. Some Theoretical Aspects of Road Traffic Research. Proceedings of the Institute of Civil Engineers, Vol. 2, pp. 325-378, 1952.

[10] Dijkstra, E., A Note on Two Problems in Connection with Graphs, Numerische Mathematik, Vol. 1, pp. 269-271, 1959.

[11] Quadstone Limited, Paramics User Guide Version 4.0, Quadstone Limited, Edinburgh, UK, 2003.

[12] Chu, L., Liu, H., Recker, W., and Zhang, H. M. Performance Evaluation of Adaptive Ramp Metering Algorithms in Paramics Simulation. Journal of Transportation Engineering, Vol. 130 (3), 330-338, 2004.

[13] Chu, L., Liu, H., and Recker, W. Capability-Enhanced PARAMICS Simulation with Developed API Library. Proceedings of the ITS World Congress, Chicago, Illinois, 2002.

[14] Daganzo, C.F. The cell-transmission model: A simple dynamic representation of highway traffic. Transportation Research 28B(4), 269-287, 1994.

[15] Daganzo, C.F. The cell transmission model, part II: network traffic. Transportation Research 29B(2), 79-93, 1995.

[16] Jin, W. Kinematic Wave Models of Network Vehicular Traffic. Ph.D dissertation, University of California, Davis, 2003.