Marginal Dynamic Network Loading for Large-scale Simulation-based Applications

Ruben Corthout*
Department of Mechanical Engineering
CIB / Traffic & Infrastructure
University of Leuven
Celestijnenlaan 300A Bus 2422
B-3001 Heverlee
T +32 16 32 9614
F +32 16 32 2986
ruben.corthout@cib.kuleuven.be

Chris M.J. Tampère
Department of Mechanical Engineering
CIB / Traffic & Infrastructure
University of Leuven
Celestijnenlaan 300A Bus 2422
B-3001 Heverlee
T +32 16 32 1673
F +32 16 32 2986
chris.tampere@cib.kuleuven.be

Rodric Frederix
Department of Mechanical Engineering
CIB / Traffic & Infrastructure
University of Leuven
Celestijnenlaan 300A Bus 2422
B-3001 Heverlee
T +32 16 32 9614
F +32 16 32 2986
rodric.frederix@cib.kuleuven.be

Lambertus H. Immers
Department of Mechanical Engineering
CIB / Traffic & Infrastructure
University of Leuven
Celestijnenlaan 300A Bus 2422
B-3001 Heverlee
T +32 16 32 1669
F +32 16 32 2986
ben.immers@cib.kuleuven.be

* corresponding author

Word count: 6228 + 5*250 = 7478
Submission date: November 8th, 2010
ABSTRACT

Currently, the scope of using macroscopic Dynamic Network Loading (DNL) models for applications such as real-time traffic management, reliability and vulnerability studies, network design and dynamic origin-destination (OD) estimation is limited by the computational overhead. The main reason is that these applications require a large number of DNL runs to be performed. Since the successive simulations often exhibit a large overlap, this problem can be overcome by introducing marginal simulation. Through marginal simulation, iterative or Monte-Carlo simulation can be performed much more efficiently by approximating each simulation as a variation to one single base simulation. Thus, repetition of countless identical calculations is avoided. The Marginal Computation (MaC) model presented here is a marginal DNL model consistent with first-order kinematic wave theory, thus realistically capturing congestion dynamics. It can model both demand and supply variations, which means it is suited for a wide range of possible applications. A case study on a medium to large-scale network (around Ghent, Belgium) is added to illustrate its performance.
1. INTRODUCTION

For practical applications such as the planning, evaluation and real-time management of large-scale congested networks, the best choice to model the traffic flow propagation is arguably a macroscopic simulation-based dynamic network loading (DNL) model consistent with first or higher order traffic flow theory. Examples are the Cell Transmission Model (CTM) (Daganzo (3)) and the Link Transmission Model (LTM) (Yperman et al. (4)), both based on kinematic wave theory (introduced by Lighthill & Whitham (1) and Richards (2)). Naturally these DNL models also exhibit shortcomings and approximations, but they model congestion formation and spillback far more realistically than analytical models and simulation-based models with link exit and performance functions, and especially than static models. Microscopic models on the other hand have the disadvantage of requiring a much higher computation time and calibration effort compared to macroscopic models.

In many of the potential applications of macroscopic, simulation-based DNL models, a large number of simulation runs need to be performed on the same network. For example, in a dynamic traffic assignment (DTA) assuming a probabilistic equilibrium, in which travellers’ choices result from stochastic traffic states, a lot of network evaluations are required to describe this stochasticity. Furthermore - and whether or not an equilibrium assumption is adopted -, variability and vulnerability studies following a Monte-Carlo approach require a very large set of perturbed samples (with varying demand and/or supply) to be simulated. Also in optimisation problems such as origin-destination (OD) estimation, (robust) network design and optimisation of traffic control measures such as dynamic pricing, ramp metering, signal planning and route guidance, iterative simulations can be applied to determine the derivative of the objective function to the input variables.

As stated earlier, macroscopic simulation-based kinematic wave DNL models are (most) suited for the above mentioned applications, especially on large-scale congested networks. However, the relatively high computation times of these models render performing a large number of simulations troublesome. Especially real-time or large-scale applications are often infeasible. For example Ukkusuri et al. (5), who present an approach for robust signal control under varying demand with an embedded CTM, state that the computational burden currently limits the scope of their research to few scenarios and small networks.

Several ways to reduce the computation time are reported in the literature. A first way is to limit the number of simulation runs. For example in optimization problems, imposing a maximum number of runs is often unavoidable. Obviously, this is undesirable, since reaching an optimum is not guaranteed. Also in travel time variability studies, a small set of samples does not allow to correctly construct probability distributions of travel times. Preferably, the number of simulations should be reduced by ‘smart sampling’, i.e. focussing on those samples for which a large impact on the (network) performance or the objective function is expected (see Tampère et al. (6)). For instance vulnerability analysis are sometimes defined as two player games, in which one player – the ‘evil entity’ – aims to maximize the damage by striking only the most vulnerable links (e.g. Murray-Tuite & Mahmassani (7)). Although smart sampling is advisable when tackling large-scale problems, additional computation time savings are necessary.
A second option is to revert back to the simpler, faster tools like (static) models with link exit or performance functions. Examples of studies applying such models are plentiful in the literature. For instance Noland et al. (8) and Lo & Tung (9) apply a model based on link performance functions to obtain the probabilistic distribution of travel times under stochastic link capacities. Also Chen et al. (10) use a static assignment in their Monte-Carlo framework to assess reliability under correlated variations of link capacities. Snelder et al. (11) do not consider congestion effects for their optimal redesign of the Dutch road network, and neither does Jenelius (12) when assessing network vulnerability by imposing link closures. Although the application of simple but fast models can be justified on networks with relatively low traffic loads, proper modeling of congestion spillback is vital to obtain credible results on congested networks (Knoop et al. (13)).

Another strategy that is often applied to lower the computational burden is to develop other (approximate) methods so that repeated simulations are avoided. For example in real-time applications, feedback approaches are often used instead of potentially more effective, but troublesome iterative procedures (e.g. Pavlis & Papageorgiou (14)). Others use analytical approximations, e.g. Du & Nicholson (15) and Bell et al. (16), who study network reliability through analytical sensitivity analysis. Also Clark & Watling (17) avoids Monte-Carlo sampling by analytically estimating the network travel time distribution under stochastic demand. Furthermore, Ukkusuri & Waller (18) propose several approximation schemes to determine the solution to the probabilistic equilibrium problem through evaluation of one single-point demand pattern. Following the same philosophy, Ng & Waller (19) transform a range of stochastic link capacities to one deterministic value. By doing so, they account for travel time variability due to supply variations in their dynamic route choice model. Some other work concerning vulnerability analysis avoids cumbersome simulations by developing simple quick scan methodologies (Scott et al. (20), Tampère et al. (21)). Furthermore, in dynamic gradient-based OD estimation (e.g. Cascetta & Postorino (22), Bierlaire & Crittin (23)), the gradient of the objective function (usually expressing the deviation between simulated link flows and detector counts) to the demand is typically not determined through repeated simulations (finite differences) but approximated by a so called assignment matrix to speed up the optimization procedure. Frederix et al. (24) discuss the importance of using the correct gradient in dynamic OD estimation on congested networks. Finally, an alternative to repeated simulation is presented recently by Sumalee et al. (25) in the form of a stochastic CTM, modeling stochastic network flows under uncertain demand and supply. The mean and standard deviation of densities are calculated and propagated through the network, as are the probabilities of various traffic states. The Markovian property is adopted, i.e. these probabilities are uncorrelated between adjacent cells. This is probably a too harsh assumption; see analogously the work of Nelson & Raney (26) concerning kinetic traffic flow models. However, without this assumption of uncorrelated traffic states, stochastic DNL models seem computationally infeasible. In conclusion, approximate methodologies as described above – like the simple, fast models - typically lack a sufficiently realistic consideration of congestion dynamics and the link interdependencies. Therefore,
while some can definitely be valuable in cases with no or mild congestion, they are expected to fail under more extreme conditions.

The approach proposed in this paper is also an approximation aiming at computational gain, but not by avoiding repeated (Monte-Carlo) simulations. Rather, the simulation runs itself are approximated. In that sense, philosophically similar approaches can be found in Chiu et al. (27) and AbdelFatah & Mahmassani (28). The former propose a method to transform the network and OD-matrix to allow more efficient simulation of mass evacuation. In the latter, an approach is described to optimize signal green times through repeated simulation of only a part of the network, namely the manually defined local area around the considered signal control. However, they do not use this approach in their experiments.

We present the Marginal Computation (MaC) model, which approximates explicit DNL runs in the form of partial, marginal simulations. The idea behind marginal simulation is to only calculate the changes in traffic and congestion propagation that arise from a local variation (in demand or supply) to a so called base simulation. Thus, the computational efficiency is substantially increased, enabling applications that are infeasible with an explicit simulation approach. Meanwhile, MaC stays conceptually and algorithmically very close to the explicit DNL model it is derived from – in this case the LTM (Yperman et al. (4)). This means that the drawback of fast, simple models and the approximate methodologies in the literature is avoided: MaC exhibits a realistic representation of congestion formation and spillback just like the LTM. Moreover, while the approximate methodologies in the literature typically exploit some of the specific features of the problem at hand, rendering them not generally applicable, MaC can be used for a wide range of applications such as reliability studies, vulnerability analysis and optimization procedures like dynamic OD estimation.

The remainder of the paper is structured as follows. Section 2 elaborates further on the philosophy, concepts and benefits of marginal simulation. In Section 3 the working of MaC is explained, as well as the approximations compared to explicit simulation. Section 4 presents a case study on the network around Ghent. Therein, a sensitivity analysis of the link flows to the demand is performed with the aim of comparing MaC to explicit simulation in terms of outcome and computation time. In Section 5, concluding remarks and future research directions are formulated.

2. MARGINAL SIMULATION

As stated in the introduction, explicitly performing a large number of simulations with DNL models is computationally troublesome. Solutions for this problem have been looked for in the literature mainly in the form of using faster but less realistic models or approximate methodologies.

Looking at the core of the computational inefficiency, the main reason is that a lot of identical calculations are executed throughout the repeated explicit simulations on the same network. Indeed, a common feature in studies like reliability studies, robust network design and so on, is that the successive simulations typically exhibit quite some overlap. Both in terms of supply (network topology, link capacities and other characteristics) and demand (route choice, OD matrix), the input is to a large
extent identical (and therefore also the output). Marginal (or partial) simulation exploits this fact by not performing each sample as a full, explicit DNL, but only aiming at calculating the changes to the outcome of a full base simulation arising from variations to the base input. By doing so, only the part of the input that differs from the base situation invokes new calculations and thus no or few identical calculations are performed. By focusing only on the (most) relevant part of the network and time period, the simulation effort is limited in space and time. Especially for large-scale problems, marginal simulation provides significant computational advantages.

Naturally, some approximation errors cannot be avoided. To some extent, marginal simulation could be regarded as the simulation-counterpart of an analytical Taylor approximation. Locally a good approximation is obtained; the further away from the base situation the larger the error. How this error evolves with the distance to the base situation is probably impossible to describe exactly, but in Section 3 we elaborate further on the approximation errors in MaC and try to create some notion on when these errors arise and grow. Since marginal simulation is an approximation designed for fast iterative or Monte-Carlo simulation, its application is limited to problems where the outcome of one particular simulation is not directly of interest. Rather the aim should be to quantify variability (e.g. in reliability and vulnerability studies) or to approximate the derivative of an objective function to the input variables (e.g. in OD estimation). Especially in the first iterations - when the distance to the optimum is still large – fast marginal simulation probably suffices to determine the search direction. In the last iterations, one could divert to explicit simulation if a higher accuracy is required to reach the optimum.

It should be noted that marginal simulation could be beneficial not only in the field of transportation. It might provide a solution in any research domain or application in which simulations with large overlap are performed and in which computation time is an important constraint. What the marginal approximation of the explicit model should look like of course depends on the process that is being simulated. Preferably, the marginal model should be closely related to the ‘maternal’ explicit simulation model, in so far as the computational constraints allow this. As such, marginal simulation does not require the development of new theories to describe the process. Section 3 describes how MaC approximates the LTM; marginal models could be derived from other DNL models analogously.

3. **THE MARGINAL COMPUTATION MODEL**

First, we briefly summarize previous related work on which MaC is based conceptually and algorithmically. In the second paragraph, the working and required input of MaC is explained. Finally, the approximation errors that can arise compared to explicit simulation are discussed.

**Earlier Work**

The philosophical predecessor of our current developments is the Marginal Incident Computation (MIC) model (Corthout et al. (29, 30)). This model is conceptually highly similar to MaC, since it approximates incident congestion effects by
superimposing them onto a base simulation. Although computationally highly efficient, the MIC procedure proved not to be extendable to demand variations. The new MaC model has the advantage of combining supply and demand variations, which significantly broadens the scope of possible applications and future developments (such as including en-route rerouting in incident cases). Furthermore, the approximation errors are reduced because MaC is algorithmically closer to the maternal explicit model.

The DNL model from which MaC originates is the Link Transmission Model (LTM) by Yperman et al. (4). The LTM is a macroscopic DNL model, consistent with simplified first order kinematic wave theory after Newell (31). It thus assumes traffic states and shock waves to be in accordance with a triangular fundamental relation between density and flow and exhibits a realistic representation of congestion formation and spillback. In the LTM, the simulation is performed by updating the nodes with a node-dependent fixed time step. In a node update, the flows crossing the node are calculated and stored – in the form of cumulative vehicle numbers or curves – at both the downstream end of the incoming links and the upstream end of the outgoing links. Thereby, a multi-commodity framework is adopted, where each commodity corresponds to a specific pre-defined route.

Most of the functions of the LTM can be deployed in MaC. Naturally, some additions have to be made to detect and track the variation to the base simulation in each marginal sample and to improve the computational efficiency. So, MaC models the propagation of flows and congestion on links just like the LTM does (see (4)), except for the multi-commodity representation which is approximated by a more efficient single-commodity approach (see the next paragraph). The node model applied in MaC – and currently also embedded in the LTM - to determine the flows transferred over intersections is the one presented more recently in Tampère et al. (32).

The LTM also performs the base simulation which is input to MaC. In principle, a base simulation run from any existing DNL model could be fed into MaC. However, if the underlying theory of the DNL model deviates from the first-order kinematic wave theory adopted in MaC this could hinder the proper functioning of MaC.

**The working of MaC**

MaC performs marginal DNL runs as variations to one single base simulation. For each variation, calculations are only performed for the active part of the network and simulation period, i.e. that part where the traffic flows (are expected to) differ from the base flows obtained on beforehand. The links on which different flows are detected are the affected links, constituting the affected area. Differences compared to the base results can arise from both supply and demand variations to the base input.

First, the MaC procedure is presented in a stepwise scheme. Afterwards, each of these steps is explained in detail.

0. Read input; initialize $i=1$
1. Impose change according to variation $i$ to base input
2. Activate part of network and time period for simulation
3. Simulate traffic propagation in active part of network
   Check if affected area grows and if so, activate additional part of network
4. Postprocessing:
   Write output
   Restore base input
   Set $i = i + 1$ and return to step 1

0. Read input
MaC requires the following input:

- Base input

  By the base input, we mean the traditional input any DNL model requires (network characteristics, dynamic OD matrix and route choice). This is stored in MaC as the base input variables, which can be changed to simulate the variations.

- Characteristics of variations

  For each variation (demand or supply), it needs to be known which of the base input variables change and to what extent. In case of a supply variation (typically an incident on a link), the incident location, starting and ending time and the fraction of the base capacity that remains are specified. For demand variations, the OD-pair(s) or route(s) on which the demand increases or decreases are needed, as well as starting and ending time of the change and the fraction by which the base demand changes.

  In the remainder of the paper, we focus on demand variations since these are more intricate to model and because in future work, the modeling of supply variations will be further improved by adding some of the functionalities of demand variations to simulate downstream propagating effects.

- Base results

  From the base simulation run, the following information is derived as input to MaC:

  - Total cumulative vehicle numbers at upstream and downstream link ends

    These represent the total traffic flows (single-commodity, not disaggregated by route) and states that occur in the base simulation.

  - Turning fractions at downstream link ends
The dynamic profiles of the turning fractions $f_a(t)$ at each link $a$ (i.e. the fraction of the total link flow on $a$ that is headed for each downstream direction) are needed for the single-commodity approach adopted in MaC (see step 3). Since in the LTM the cumulative curves are stored per route, discretized turning fractions can easily be derived.

- **Dependency of turning fractions on demand**

To simulate demand variations, it needs to be known how the turning fractions change with a change in demand. The dynamic turning fractions $f_a(t)$ of link $a$ are a function of the demand profile ($q_{r,a}(t)$) for each route $r$ that passes through $a$ and the travel time $t_{r,a}(t)$ vehicles of route $r$ have experienced when arriving at the downstream end of link $a$ since departing from the origin ($f_a(t) = g(q_{r,a}(t), t_{r,a}(t))$). The time dimension ($t$) is omitted from now on for notational convenience. This function is likely to be non-linear and is not described analytically. Rather, we derive this dependency numerically for the discretized turning fractions from the base multi-commodity cumulative curves. This information allows the propagation of a demand variation through the network by adapting the turning fractions along the way.

1. **Impose change**

According to the characteristics of the variation, the proper change(s) are made to the base input variables. For a supply variation, this implies for example to decrease a link’s capacity for the duration of an incident. A demand variation is imposed by changing the demand at the origin of a certain route or OD pair. Furthermore, the turning fractions at the downstream link ends along the route must be updated to guide the demand variation through the network to its destination. For example, consider the network depicted in FIGURE 1, and a demand variation in the form of an increased route flow $q_r$ from $t=t_0$ on, with route $r$ traversing links 1 to 6. The turning fractions $f_a$ ($a=1..5$) are updated for all $f_a(t>t_a=t_0+t_{r,a}^0$ on, with $t_{r,a}^0$ being the base travel time from the origin to the downstream end of $a$ for a vehicle departing on route $r$ at $t_0$. For example, the increased route flow causes $f_3$ to be altered so that more traffic turns left towards link 4.
2. **Activate part of network**
The marginal simulation of the current variation is initialized by activating that part of the network where changes to the input variables (e.g. a decreased link capacity or altered turning fractions) are imposed in step 1. Since simulations in MaC (as in the LTM) are governed by node updates, it is the nodes that need to be activated. In the example in FIGURE 1, nodes 1 through 5 are activated from $t_a$ on, $t_a$ again being the arrival time at node $a$ of the first vehicle of the increased route flow $q_r$ (as in step 1). Also, the status of the links of route $r$ is set to ‘affected’.

Note that nodes, once activated, are not deactivated in MaC before the simulation of the current variation is finished. This would necessitate introducing additional checks to detect the end of the change, which would be prone to errors.

3. **Simulate**
In step 2, some of the nodes in the network have been activated. For this active part, node updates are performed which calculate the flows crossing the node from each incoming link to each outgoing link. In each node update, MaC - like the LTM - derives the local demand and supply at the node level from the cumulative curves of the links. For affected links, from the new curves inherent to this variation; for unaffected links this information is derived from the base curves. The main difference with the LTM is that the multi-commodity approach is approximated by a single-commodity representation. This implies that traffic is simulated as one homogenous flow, regardless of the followed route. Thus, only one, total cumulative curve is stored at each link end, instead of one for each route as in the LTM. The turning fractions divide the total flow at each downstream link end towards the various directions. More details on this single-commodity approach can be found in Bellei et al. (33) and Taale (34), who apply this approach iteratively in a DTA.

At each node update, the calculated flows of affected links are stored in the new cumulative curves. For unaffected incoming links, the calculated flows are
compared to the base flows of the same time step. Hereby an accuracy level $\epsilon$ is used in the check $(\frac{|q_{\text{new}} - q_{\text{base}}|}{q_{\text{base}}} > \epsilon)$ so that very small changes are not needlessly tracked. If a difference is detected on a link (i.e. congestion forms or the throughput in existing congestion in- or decreases), it becomes affected. The upstream node of this link $a$ is then activated (if currently inactive) at $t_c + \frac{L_a}{w_a}$ to propagate the change upstream.

With $t_c$ the current time, $L_a$ the length of $a$ and $w_a$ the maximum negative shockwave speed from the triangular fundamental diagram, this is the earliest time the change could reach the upstream node. Changes in flow are not propagated downstream (see next paragraph).

To illustrate how the affected area and the active part of the network can expand, regard again the example network (FIGURE 2). Suppose a bottleneck has activated on link 5 due to the increased route flow $q_r$. While link 7 remains unaffected, congestion spills back to link 8, affecting it and activating node 8 at $t_s = t_c + \frac{L_8}{w_8}$. The downstream effects (in this case a decreased flow) in non-affected directions are disregarded (indicated by the blue arrow).

In conclusion, it is clear from FIGURE 2 that the computational effort is limited in space and time. Outside the affected area and time period, the flows and states are assumed identical to the base situation.

4. Postprocessing
The main output of MaC consists of the new cumulative vehicle numbers of affected links. From this, other output such as total time spent and link and route travel times can be derived. Afterwards, all variables (demand, supply, cumulative curves and...
turning fractions) are reset to the base values and the next variation is simulated (starting at step 1).

**Approximation errors**
This paragraph elaborates on the possible causes of approximation errors in MaC compared to explicit simulation.

- **Discretization of turning fractions**
  
The turning fractions are discretized for input in MaC with a fixed time interval, which may cause small deviations from explicitly calculated flows. Since this time interval can be chosen reasonably small (in the order of a few minutes), large errors are unlikely to arise from this approximation. Naturally, the error increases with the time interval and with the volatility of the turning fractions.

- **Dependency of turning fractions on travel times**
  
  As explained earlier, the dependency of the turning fractions of each link $a$ on the base demand and base travel times are input to MaC. To impose a demand variation, this relationship is used to update the turning fractions. However, the demand variation may cause the travel times to change and thus alter this relationship. This can cause a time discrepancy in the turning fractions in MaC compared to explicit simulation. This error increases as the marginal travel times deviate more from the base travel times, and with the volatility of the turning fractions. In case of unacceptable errors, MaC could be run in an iterative loop to update the relation between turning fractions and demand according to the new travel times. Testing indicates that the error decreases rapidly, so that only two or three iterations would be needed. Of course, the computation time would increase accordingly.

- **Neglecting downstream effects**
  
  In demand variations, the primary downstream change on the route itself is accounted for, but (in the current version of MaC), links can only become affected during simulation from the downstream end. Thus, only upstream propagating changes are tracked and secondary downstream effects neglected. Usually this is acceptable, since only a shift in the cumulative vehicle numbers is expected on downstream links, not an additional delay. However, possibly a relieving or overloading of downstream bottlenecks is missed. This error increases with the load in the base simulation and the magnitude of the variations. In future research, we intend to improve MaC to account for the most important downstream effects.
4. CASE STUDY ON THE NETWORK AROUND GHENT

In this section, a case study on the network around Ghent (Belgium) is presented to compare the performance of MaC with that of explicit simulation. The explicit simulations, as well as the base simulation are performed with the LTM.

A four hour simulation period is considered on the Ghent network, which consists of 326 nodes, 992 links and 1916 routes connecting 32 origins and destinations. FIGURE 3 depicts the network and the traffic states in the base situation at the peak moment. The bar width represents the link density and the color the speed (with red representing congestion and green free flow). A heavily loaded case is chosen, with a lot of base congestion. Thus, MaC is tested in quite extreme conditions, invoking approximation errors to obtain an upper bound of the error to be expected in a real-life case study.

![FIGURE 3 Base situation on the Ghent network](image)

On this network, a set of demand variations is simulated with MaC on the one hand and explicit simulation on the other. The set consists of one demand increase on each route with 50 veh/h (1916 variations), during one of the four hours of the simulation period (randomly chosen). In Frederix et al. (35), MaC is analogously applied to determine the gradient of the objective function in a dynamic OD estimation (through forward finite differences).

A time interval of 2.4 min is used for the discretized turning fractions; the accuracy level $\epsilon$ is set to 0.5%. The outcome of MaC for each variation is compared to that of explicit simulation by means of the link flows (per 15 min). For link flows unaffected in MaC, the base flow is used as the MaC result.
For all 15 minute link flows, 1916 values are obtained (one for each variation). Of these, MaC reports 0.9% as affected (i.e. changed compared to the base simulation). Only 4.9% have changed in the explicit simulations but not in MaC. FIGURE 4 shows the cumulative density function (CDF) of the difference between the explicit variation flow and the corresponding base flow; for the 4.9% ‘wrongfully unaffected’ flows in red and for the 0.9% affected flows in blue. FIGURE 4 thus depict the explicit results, which MaC tries to approximate as closely as possible.

![FIGURE 4 CDF difference in link flows explicit-base](image)

Naturally, since the flows in the red curve are unaffected in MaC, the error is equal to the explicit difference in FIGURE 4. For only 2.9% of these (0.1% of the total number of flows) the absolute value of the error is \( >5 \text{ veh/h} \). This indicates that the errors made in the unaffected area are acceptable.

FIGURE 5(a) shows that for 92.3% of the affected flows, MaC predicts a flow within a range of 2 veh/h from the explicit result. In FIGURE 5(b), the relative error of the largest deviations (explicit flow differs \( >5 \text{ veh/h} \) from base flow) is plotted. For 88.1% of these flows, the error in MaC is smaller than 10%.
Comparing computation time, the explicit simulations take almost 25 hours, whereas MaC needs 2 hours and 19 minutes. However, MaC requires only 2.5% of the number of node updates that are performed explicitly. The reduction factor of the affected area compared to the network size is thus significantly larger than the computational gain. This indicates that considerable inefficiencies exist in the MaC code. Carrying out a thorough code optimization should thus significantly increase the computational gain.

5. CONCLUDING REMARKS AND FUTURE RESEARCH

This paper discusses the concept and benefits of marginal dynamic simulation of traffic flows in large-scale networks. By performing marginal simulations as demand or supply variations to a full base simulation, calculations are limited to a small part of the network and time period. By doing so, the computational restrictions of repeated macroscopic DNL can be relieved to a large extent. This opens the door to applications that are currently infeasible (except on a very small scale) such as reliability and vulnerability studies accounting for the stochasticity of demand and supply, optimization problems (e.g. dynamic OD estimation and (robust) network design) and real-time traffic management (ramp metering, variable speed limits, route guidance and so on). Moreover, the concept of marginal simulation should be extendable to other fields of study where computationally demanding simulations limit the research scope (e.g. optimal factory design and supply chain management).

In a case study on a medium to large-scale network, the MaC model presented here yields a computational gain of a factor 11 compared to explicit simulation with the LTM. Although this gain is considerable already, the fact that the marginal simulation covers on average only 2.5% of the whole network and time period indicates a higher potential. Therefore, a thorough code optimization constitutes an important future task to further improve the computational gain.
Furthermore, the case study shows that MaC approximates the explicit results reasonably well. Moreover, in future research we intend to further improve MaC by accounting for some of the secondary downstream effects, for instance in case of en-route rerouting around incidents. Therefore, additions have to be made to the algorithm to check whether or not it is necessary to propagate a change (depending on the magnitude of the change and the presence of downstream bottlenecks). This way, MaC will be further refined for practical applications.

References


(32) Tampère, C.M.J., R. Corthout, D. Cattrysse & L.H. Immers. A generic class of first order node models for dynamic macroscopic simulation of traffic flows. Accepted for publication in *Transportation Research Part B*, in press.

