

Simulation and optimization of trajectories in a congested network

Gunnar Flötteröd* and Kai Nagel**

Abstract—The subject of this article is twofold. Firstly it describes how a dynamic macroscopic network loading model can be used to simulate the movement of arbitrarily complex individual particles (agents) through a network without loss of the macroscopic model’s differentiability. Based on this result the problem of minimizing a given functional of the macroscopic model’s states by adjustment of individual agents’ trajectories is considered. A solution procedure is proposed, which is based on subsequent linearizations of the overall system dynamics and time variant best path calculations.

I. INTRODUCTION

This is the second of three articles providing the theoretical framework for a novel methodology of traffic state estimation based on multi-agent simulations. The overall goal of this work is to provide an algorithm that estimates agents’ route and activity location choice from anonymous traffic measurements such as flows or densities. We expect this approach to usefully link flexible but less formalized approaches for agent-based demand generation and microsimulation [2] with mathematically well understood state estimation methodologies from control engineering [13], [16].

As a basic building block of our system, we presented in a first article an approximately differentiable first order traffic flow model, which allows for dynamic loading of traffic onto a network of arbitrary topology. Traffic flow was assumed to be anonymous insofar as route choice was represented only by exogenously provided splitting fractions at intersections [8].

In this article, we show how this model can be applied to load individual agents with arbitrarily complex behavioral algorithms onto the network and still preserve its advantageous analytical properties. Considering a general functional of the network’s states to be given, we then propose a method for iterative minimization of this functional in terms of a Nash game between all agents, which basically founds on repeated linearizations of the overall system. The individual optimization problem every single agent faces in this game is efficiently solved by a time variant best path algorithm.

The fact that the considered functional is not specified in terms of a real world application makes this work somewhat theoretical. In a third article we will use it to express the Bayesian *a posteriori* probability of an agent’s route and activity location choice given an *a priori* behavioral

assumption and additional anonymous traffic measurements [9].

Our current work is involved with estimation of agent *behavior*. Physical aspects such as traffic densities or velocities are not subject to direct estimation, but rather result from travelers’ behavioral patterns. In this regard, we hope to complement other approaches concentrating on physical properties and less on behavioral issues [1], [13], [16].

The remainder of this article is organized as follows. Section II-A provides some necessary background on the macroscopic model introduced in [8]. In section II-B, it is shown how individual particles can be moved through this model, while section II-C explains how individual particle behavior can be reproduced on average by this model. Section II-D provides the overall simulation algorithm and closes the modeling and simulation part of this article. In section III, it is shown how the proposed model can be applied for approximate solution of a control problem in terms of individual particles’ route choice. In the conclusion, an outlook on an upcoming large-scale real world application is given.

II. MODELING AND SIMULATION

A. Review of the traffic flow model

First order traffic flow models are based on a minimal set of assumptions. Beyond a continuity equation, the specification of a speed-density relationship already suffices to simulate traffic on a straight road. At intersections, the simulation problem is under-determined unless additional constraints are introduced. In our approach, which extends Daganzo’s cell-transmission model [5], [6], exogenously provided turning proportions and cell inflow priorities ensure a unique solution. The model is discretized in time and space, where spacial segments are referred to as cells. The resulting simulation scheme can be understood as an application of the Godunov method as it was noted already in [12].

The only traffic flow parameters of relevance to this article are the number of vehicles $x_i(k)$ on cell i during discrete time step k , the number $\Delta x_i^{in}(k)$ and $\Delta x_i^{out}(k)$ of vehicles entering and leaving the cell at k , the time step length T and the time variant turning proportions $\beta_{ij}(k)$ between cell i and its succeeding cells $j \in \mathcal{S}(i)$. Only for notational simplicity it is assumed that there is a constant amount of traffic moving across the network, although the method works without changes if this simplification does not hold.

Traffic states (vehicle counts) are updated in every simulation time step by

$$x_i(k+1) = x_i(k) + \Delta x_i^{in}(k) - \Delta x_i^{out}(k). \quad (1)$$

*Gunnar Flötteröd is with the Group for Transport Systems Planning and Transport Telematics, Technische Universität Berlin, D-10587 Berlin, Germany. Tel: +49-30-314-29520, fax: +49-30-314-26269, mail: fbetteroed@vsp.tu-berlin.de.

**Kai Nagel is head of the Group for Transport Systems Planning and Transport Telematics, mail: nagel@vsp.tu-berlin.de.

Cell i 's outflow $\Delta x_i^{out}(k)$ is a function of (a) the states of all downstream cells of i , (b) all upstream cells' states of these downstream cells, and (c) all parameters influencing flow transmission between these cells. In the context of this article, only turning proportion parameters $\beta_{ij}(k)$ are considered. A sufficiently precise formalization of this is given by

$$\Delta x_i^{out}(k) = \Delta x_i^{out}[\mathbf{x}(k), \boldsymbol{\beta}(k), k] \quad (2)$$

where $\mathbf{x}(k) = (x_i(k))$ and $\boldsymbol{\beta}(k) = (\beta_{ij}(k))$ comprise all states and turning proportions. Cell i 's inflow then follows from its predecessor $\mathcal{P}(i)$'s outflows according to

$$\Delta x_i^{in}(k) = \sum_{l \in \mathcal{P}(i)} \beta_{li}(k) \Delta x_l^{out}(k). \quad (3)$$

As the linearization of such a model was first demonstrated in [8] and continuously extended since then, the entire model as given by (1), (2), and (3) can assumed to be approximately differentiable with respect to both, states and turning proportions.

B. Particle movement

The macroscopic flow model is based on a speed-density relationship. For every cell, this can be transformed into a relationship between speed and vehicle count given by

$$v_i(k) = v_i[x_i(k), k]. \quad (4)$$

Velocity $v_i(k)$ prevails on cell i during time step k 's entire duration T .

We now consider a set \mathcal{M} of particles (a "population" of travelers, agents, or vehicles) floating through the system. Since we ignore traffic entering or leaving the system, \mathcal{M} is of constant size. Particles have no "mass" insofar as they do not contribute to the macroscopic occupancy on a cell. It is assumed that at the time of a particle's entrance into the network, an appropriate amount of macroscopic flow has also been dismissed into the system, resulting in a mass balance between particles and total macroscopic occupancy.

During every time step, each particle μ moves according to the velocity on its current cell as given by (4). Here, movement is regarded as continuous in time: When μ crosses an intersection during a single move of duration T , it can freely choose its next cell and continue with the velocity encountered there until its time step ends. An example of this procedure is given in Figure 1.

C. Particle route choice

Having stated the influence of macroscopic dynamics onto individual particles, we now consider the opposite problem of synchronizing the macroscopic traffic flow with the particles' behavior. For this purpose, we introduce additional states $x_{ij}(k)$ representing the accumulated number of vehicles having moved from cell i to cell $j \in \mathcal{S}(i)$ until the beginning of time step k . In order to specify these states' dynamics, we also define a control vector $\mathbf{u}(k) = (u_{ij}(k))$ with component $u_{ij}(k)$ being equal to the number of particles having moved

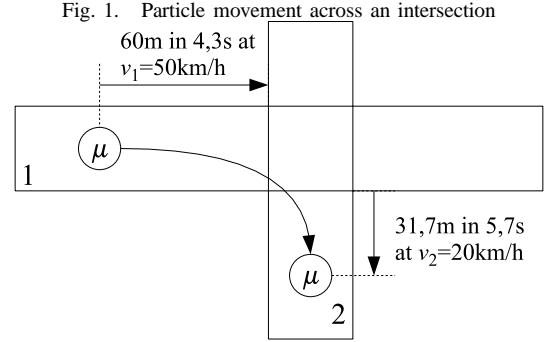


Fig. 1. Particle movement across an intersection

In this example, particle μ comes from the left and makes a right-turn into the downwards road segment. The time step duration is $T = 10\text{s}$. The particle needs 4,3s to reach the end of link 1 at $v_1 = 50\text{km/h}$. During the remaining 5,7s, it advances another 31,7m on link 2 at $v_2 = 20\text{km/h}$.

from cell i to cell $j \in \mathcal{S}(i)$ in time step k . This allows to give an additional state equation

$$x_{ij}(k+1) = x_{ij}(k) + u_{ij}(k) \quad (5)$$

for every turning relation ij . The control vectors' components $u_{ij}(k)$ are additively comprised of all particles' turning behavior, which is stated by

$$u_{ij}(k) = \sum_{\mu \in \mathcal{M}} u_{ij}^{\mu}(k), \quad (6)$$

where $u_{ij}^{\mu}(k) \in \{0, 1\}$ is 1 if particle μ proceeds from cell i to cell j during k and 0 otherwise.

In order to connect these quantities to the traffic flow model, we assume that the turning decisions of particles leaving any cell i follow a multinomial distribution with unknown choice probabilities. A raw maximum likelihood estimator of these probabilities is then used to calculate the model's turning proportions via

$$\beta_{ij}(k) = \frac{x_{ij}(k)}{\sum_l x_{il}(k)}. \quad (7)$$

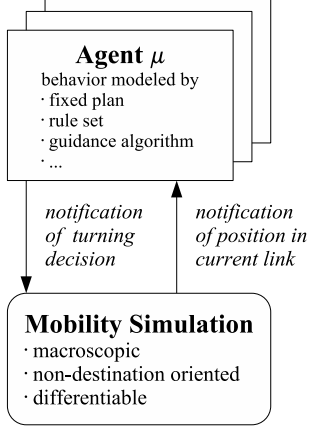
We are aware of more sophisticated calculation schemes [11], but even this straightforward formula makes the point perfectly clear. By substitution of these turning proportions into (2) and (3), we can split the macroscopic traffic flow according to the individual particles' behavior.

Recapitulating, we express the macroscopic system dynamics by a general state equation

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k], \quad (8)$$

where $\mathbf{x}(k)$ comprises both, vehicle and turning counter states, $\mathbf{u}(k)$'s influence on the system is given by (5) and (7) and function \mathbf{f} subsumes the previously given state update equations (1) and (5). Note that \mathbf{f} stays differentiable with respect to both, \mathbf{x} and \mathbf{u} .

Fig. 2. Interaction between agents and mobility simulation



Interaction between (microscopic) agents and macroscopic mobility simulation. Note that both components can be independently chosen from broad classes of models, which is possible because of the very narrow interface in between.

D. Practical simulation of the overall model

The calculation scheme given so far assumes time-invariant turning probabilities. A straightforward approach to introduce time variance is to define an additional forgetting parameter $w \in [0, 1]$ in a modified turning counter update equation

$$x_{ij}(k+1) = wx_{ij}(k) + u_{ij}(k). \quad (9)$$

In the absence of newly observed particle movements, this causes an exponential forgetting of previously learned counts. It will have to be verified experimentally if other update schemes perform better. One possible problem with (9) is the danger of deadlock: If a jam on *one* of an intersection's outgoing cells causes *all* ingoing cells' velocities to drop, it might take a long time until new particles reach this intersection and provide fresh turning counts reflecting drivers' avoidance of the unavailable outgoing cell.

The overall simulation of a single time step can now be conducted in three stages:

- 1) Calculation of turning proportions from turning counts according to (7);
- 2) Calculation of traffic flows and update of cell occupancies as specified by (1), (2), and (3);
- 3) Movement of particles according to cell velocities as given by (4) and synchronous update of turning counters according to (6) and (5) or (9).

This approach allows us to move arbitrarily complex agents through an analytically tractable traffic flow model, as it is depicted in Figure 2.

We expect this simulation scheme to perform well even in larger scenarios for two reasons:

(1) The model does not require a realistic number of particles. If, for example, only a 10 percent sample of the complete population is loaded onto the network, the macroscopic equivalent of 10 vehicles is inserted into the system together with every particle. The chosen number of

particles must be large enough to properly represent the actual population's properties in terms of sufficiently low variance of the resulting macroscopic turning parameters, but otherwise can be minimized for fast computational performance.

(2) The macroscopic mobility simulation only moves non-destination oriented flow. No care has to be taken e.g. of partial densities, as it would be the case if route and destination choice were represented macroscopically as well.¹

III. OPTIMIZATION

If the model presented so far was only used for straightforward network loading, its usefulness might be put into question, since microscopic traffic simulation is possible with far simpler models [15]. The special contribution of this model is its ability to both move individual particles through the network and to allow for analytical analysis. In this section, we will show how to exploit this property for optimization of particle trajectories.

A. General problem statement

Assume a functional J for evaluation of the network's states during time steps $1 \dots K$ to be given by

$$J = \sum_{k=1}^K \phi[\mathbf{x}(k), k], \quad (10)$$

where ϕ maps the network state vector onto a finite, real-valued number. Further assume a linear functional

$$J^\mu = \sum_{k=0}^{K-1} \sum_{ij} c_{ij}^\mu(k) u_{ij}^\mu(k) \quad (11)$$

to be given for every particle μ , where

$$c_{ij}^\mu(k) > 0 \quad (12)$$

represents μ 's positive and finite cost of moving from i to j in time step k . In the following, we will develop an algorithm for approximate minimization of the combined functional

$$J + \sum_{\mu \in \mathcal{M}} J^\mu = \min! \quad (13)$$

by appropriate choice of all agents' individual control variables $\mathbf{u}^\mu(k) = (u_{ij}^\mu(k))$, $k = 0 \dots K - 1$. These turning decisions clearly are constrained, since any particle's route depends on its initial location, on the network structure, and on the time variant velocities on the network links. Still, we state this restriction only verbally, since compliance with it will be enforced by the chosen solution algorithm anyways.

¹An additional speedup is achieved by variation of cell sizes. By choosing larger cells for longer roads, we do not only reduce the total number of cells: Since larger cells are also updated at a lower frequency, agents being on such cells are accordingly moved less often. Still, a strict inspection of this calculation scheme in terms of first order traffic flow theory has not yet been undertaken.

B. A single trajectory

In this section, we consider a linearized version of the full problem, where only one agent μ 's trajectory is subject to optimization. Since J^μ is already linear, the task remains to linearize J : Calculation of J 's reduced gradient with respect to all $\mathbf{u}^\mu(k)$, $k = 0 \dots K - 1$ is possible via a two-pass calculation [14].² Firstly, costates $\lambda(k)$ are calculated by solving the following difference equation backwards through time:³

$$\lambda(k) = \begin{cases} \frac{\partial \phi[k]}{\partial \mathbf{x}(k)} + \left(\frac{\partial \mathbf{f}[k]}{\partial \mathbf{x}(k)} \right)^T \lambda(k+1) & k < K \\ \frac{\partial \phi[K]}{\partial \mathbf{x}(K)} & k = K. \end{cases} \quad (14)$$

In a second step, sensitivities with respect to control variables can be obtained via

$$\frac{\partial J}{\partial \mathbf{u}^\mu(k)} = \frac{\partial J}{\partial \mathbf{u}(k)} = \left(\frac{\partial \mathbf{f}[k]}{\partial \mathbf{u}(k)} \right)^T \lambda(k+1) \quad (15)$$

for $0 \leq k < K$. Note that because of (6) this result is identical for all agents. Dropping constant terms, a linearization of J can be given by means of (15):

$$\bar{J} = \sum_{k=0}^{K-1} \sum_{ij} \frac{\partial J}{\partial u_{ij}(k)} u_{ij}^\mu(k). \quad (16)$$

From this, we obtain a linearized functional

$$\bar{J}^\mu = \sum_{k=0}^{K-1} \sum_{ij} \left(\frac{\partial J}{\partial u_{ij}(k)} + c_{ij}^\mu(k) \right) u_{ij}^\mu(k) \quad (17)$$

for every agent μ .

Since \bar{J}^μ is a sum of time variant costs

$$d_{ij}^\mu(k) = \frac{\partial J}{\partial u_{ij}(k)} + c_{ij}^\mu(k) \quad (18)$$

attached to the chosen turning moves ij , the application of a time variant best path algorithm on a modified network suggests itself as a solution procedure to this problem, where the original network's links comprise the new nodes and every possible turning movement in the original network is represented by a new link ij with time variant cost given by $d_{ij}^\mu(k)$. Unfortunately, it cannot be guaranteed that all $d_{ij}^\mu(k)$ are nonnegative, which can cause loops of negative cost to occur in the modified network, rendering the application of standard best path algorithms impossible. While this problem certainly is an interesting topic of mathematical research, we confine ourselves to assuming that $c_{ij}(k)$ is sufficiently large, so that the use of

$$\tilde{d}_{ij}^\mu(k) = \max\{0, d_{ij}^\mu(k)\} \quad (19)$$

²The reduced gradient regards for the dynamic constraints given by state equation (8). See [10] for another traffic related application of this method.

³Costates can be interpreted as sensitivities of J with respect to system states: Denote $J(k, K) = \sum_{c=k}^K \phi[c]$. Since the system is causal, $\frac{\partial J}{\partial x_i(k)} = \frac{\partial J(k, K)}{\partial x_i(k)} = \frac{\partial \phi[k]}{\partial x_i(k)} + \frac{\partial J(k+1, K)}{\partial x_i(k)}$ results. The dependency between different time steps is fully given by state equation (8), so we can use the chain rule: $\frac{\partial J(k+1, K)}{\partial x_i(k)} = \sum_j \frac{\partial J(k+1, K)}{\partial x_j(k+1)} \frac{\partial f_j[k]}{\partial x_i(k)}$. Substitution of $\lambda_i(k) = \frac{\partial J(k, K)}{\partial x_i(k)}$ then yields (14).

instead of $d_{ij}^\mu(k)$ provides an acceptable approximation of the exact problem.⁴

A dynamic best path search based on Dijkstra's well-known algorithm [7] efficiently provides an optimal solution to this problem in terms of a coherent path that obeys all constraints imposed by the congested network as stated above.

From a single particle's point of view the traffic situation is linear in good approximation, since control variables $u_{ij}^\mu \in \{0, 1\}$ are small compared to actual turning counts in a congested network; but even in this case we only reach an approximate minimum of J^μ 's linearization (17). The nonlinear problem is discussed in the next section.

C. Many trajectories

We now consider the problem of minimizing (13) by synchronous modifications of *many* agents' trajectories. Clearly, the increased number of degrees of freedom has the potential for a better overall solution, still this setup results in certain problems also encountered in dynamic route guidance: If many drivers are independently of each other informed of a low travel time route, they might all switch towards this route, causing a jam and very high travel times [3]. Similarly, the individual linearization (17) of the overall functional does not allow for a coordination of different particles' route optimizations.

Our proposed algorithm resembles the fixed point solution approaches to self consistent route guidance in the sense that it iteratively updates only a subset of all particle trajectories. One iteration of the algorithm is given below:

- 1) Load all particles onto the network;
- 2) evaluate target functional J and stop if desired;
- 3) differentiate target functional via (14) and (15);
- 4) choose a subset $\mathcal{M}' \subset \mathcal{M}$;
- 5) calculate a new trajectory \mathbf{u}^μ for every $\mu \in \mathcal{M}'$ that approximately minimizes \bar{J}^μ by dynamic best path algorithm;
- 6) continue with 1.

This algorithm becomes identical to a popular traffic assignment heuristic that solves the equilibrium problem in terms of a fixed point iteration [4], if $J = 0$ and J^μ represents μ 's perceived travel cost.⁵ Since traffic assignment based on this method has become common practice, we expect the method to also work well for our purposes.

D. Behavioral modeling vs. optimization

One might suspect a contradiction in our methodology: Firstly, we stated that agents' route choice is the result of an arbitrary behavioral model. Now, we calculate agents' routes by explicit optimization, invalidating any behavioral aspect. These two apparently different premises can be consolidated,

⁴The application described in [9] suggests that this approximation is reasonable.

⁵In the context of a subsequent article, such a network loading algorithm naturally results as the degenerated case of a state estimation problem if the number of observations approaches zero [9].

if the individual functionals J^μ properly reflect the relevant aspects of agent behavior.

This is the subject of a subsequent article [9], providing a behavioral model that is flexible but still tractable in this optimization context. There, we also formulate a complete state estimation problem in terms of the optimization problem discussed here.

IV. SUMMARY AND OUTLOOK

This article builds upon the availability of a dynamic, macroscopic traffic flow model that can at least approximately be differentiated. It demonstrates how such a model can be used to simulate the movement of complex individual particles through a network without loss of the macroscopic model's differentiability.

Based on this result the problem of minimizing a given functional of the macroscopic model's states by adjustment of individual agents' trajectories is considered. A solution procedure is proposed, which is based on subsequent linearizations of the overall system dynamics and makes efficient use of a well known best path algorithm.

Since we are interested in state estimation, the question remains of how to choose an appropriate target functional in the context of such an application. This problem will be discussed in another article, which provides details on the combined modeling of agents' route and activity location choice as well as the formulation of a complete Bayesian estimator.

The overall system will be tested with real world data collected Berlin during the upcoming soccer world championship, which will take place in June 2006. Since this opportunity occurred at short notice, it disarranged our original plan of first testing the system on smaller, synthetic problems before considering real world scenarios, as we envisaged when writing an earlier publication.

REFERENCES

- [1] K. Ashok, Estimation and Prediction of Time-Dependent Origin-Destination Flows, *PhD Thesis*, Massachusetts Institute of Technology, 1996.
- [2] M. Balmer, K. Axhausen, K. Nagel, A Demand Generation Framework for Large Scale Micro Simulations, *TRB 2006 (85. Annual Meeting)*, Washington DC, USA.
- [3] J. Bottom, Consistent Anticipatory Route Guidance, *PhD Thesis*, Massachusetts Institute of Technology, 2000.
- [4] E. Cascetta, *Transportation systems engineering: theory and methods*, Kluwer Academic Publishers, Dordrecht, 2001.
- [5] C.F. Daganzo, The cell-transmission model: A simple dynamic representation of highway traffic, *Trans. Res.*, vol. 28B(4), 1994, pp 269-287.
- [6] C.F. Daganzo, The cell-transmission model, part II: Network traffic, *Trans. Res.*, vol. 29B(2), 1995, pp 79-93.
- [7] E. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik*, 1959, pp 269-271.
- [8] G. Flötteröd, K. Nagel, Some practical extensions to the Cell Transmission Model, *ITSC 2005 8th International IEEE Conference on Intelligent Transportation Systems*, Wien, Austria.
- [9] G. Flötteröd, K. Nagel, Bayesian modeling and estimation of combined route and activity location choice, *submitted to ITSC 2006 9th International IEEE Conference on Intelligent Transportation Systems*, Toronto, Canada.
- [10] A. Kotsialos, M. Papageorgiou, M. Mangeas, H. Haj-Salem, Coordinated and integrated control of motorway networks via nonlinear optimal control, *Transportation Research Part C 10*, 2002, pp 65-84.
- [11] M.C. Jones, S.K. Vines, Choosing the smoothing parameter for unordered multinomial data, *Sociedad de Estadística e Investigación Operativa Test*, vol. 7, no. 2, 1998, pp 411-424.
- [12] J. P. Lebacque, The Godunov scheme and what it means for first order traffic flow models, in *13th International Symposium on Transportation and Traffic Theory*, Lyon, France, 1996, pp 647-677.
- [13] S. Lorkowski, P. Wagner, Parameter calibration of traffic models in microscopic online simulations, *TRB 2005 (84. Annual Meeting)*, Washington DC, USA.
- [14] M. Papageorgiou, *Optimierung: Statische, dynamische und stochastische Verfahren für die Anwendung*, Oldenbourg Verlag, 1996.
- [15] P. M. Simon, K. Nagel, Simple queueing model applied to the city of Portland, *International Journal of Modern Physics C*, vol. 10, no. 5, 1999, pp 941-960.
- [16] Y. Wang, M. Papageorgiou, Real-time freeway traffic state estimation based on extended Kalman filter: a general approach, *Trans. Res.*, vol. 39B, 2005, pp 141-167.