Simple queueing model applied to the city of Portland

Patrice M. Simon, Jörg Esser, and Kai Nagel*

Los Alamos National Laboratory and Santa Fe Institute

1399 Hyde Park Rd, Santa Fe NM 87501, U.S.A.

simonp@santafe.edu, esser@santafe.edu, and kai@santafe.edu

May 9, 1999

Abstract

We use a simple traffic micro-simulation model based on queueing dynamics as introduced by Gawron [IJMPC, 9(3):393, 1998] in order to simulate traffic in Portland/Oregon. Links have a flow capacity, that is, they do not release more vehicles per second than is possible according to their capacity. This leads to queue built-up if demand exceeds capacity. Links also have a storage capacity, which means that once a link is full, vehicles that want to enter the link need to wait. This leads to queue spill-back through the network. The model is compatible with route-plan-based approaches such as TRANSIMS, where each vehicle attempts to follow its pre-computed path. Yet, both the data requirements and the computational requirements are considerably lower than for the full TRANSIMS microsimulation. Indeed, the model uses standard emme/2 network data, and runs about eight times faster than real time with more than 100 000 vehicles simultaneously in the simulation on a single Pentium-type CPU.

We derive the model's fundamental diagrams and explain it. The simulation is used to simulate traffic on the emme/2 network of the Portland (Oregon) metropolitan region (20 000 links). Demand is generated by a simplified home-to-work destination assignment which generates about half a million trips for the morning peak. Route assignment is done by iterative feedback between micro-simulation and router. An iterative solution of the route assignment for the above problem can be achieved within about half a day of computing time on a desktop workstation. We compare results with field data and with results of traditional assignment runs by the Portland Metropolitan Planning Organization.

Thus, with a model such as this one, it is possible to use a dynamic, activities-based approach to transportation simulation (such as in TRANSIMS) with affordable data and hardware. This should enable systematic research about the coupling of demand generation, route assignment, and micro-simulation output.

Keywords: large scale transportation simulations; traffic simulations; transportation planning; queueing models

*corresponding author

1 Introduction

In transportation forecasting for planning applications, the generation of the transportation demand is crucial for useful forecasts. Activity-based demand modeling [1, 2] is a promising technology here. This means that one attempts to predict people's activities (work, shop, sleep, etc.) and to obtain the transportation demand because activities at different locations need to be connected via transportation. Yet, it is clear that the *impedance* of the transportation system, i.e. the structure of the travel times and inconveniences one experiences, plays a critical role for activity patterns and activity locations. For example, high congestion may be an incentive to drop intermediate stops at home between activities, or it may be an incentive to do activities at home instead of somewhere else. Thus, a critical part of an activity-based transportation forecasting system is the representation of the transportation dynamics.

Any such approach needs to achieve "consistency" between modules. By this it is meant that the congestion assumptions on which people plan their activities need to actually be encountered when people execute their plans. Otherwise, people will adjust their plans in reaction to the traffic conditions they just found, and so the result is not stable and thus useless for forecasting.

This consistency criterion between plans-making and transportation dynamics can be formulated as a fixpoint problem [3, 4, 5]. Fixpoints can, under certain conditions, be found via relaxation. In our case this means: make all plans; execute the micro-simulation with all plans; let (some or all) people change their plans according to the simulation result; etc.; until the simulation result is consistent with people's expectations. Since for time-dependent problems with a dynamically correct representation of congestion dynamics no better approach is currently known, this is what is done by many groups [6, 4, 7, 8].

From a conceptual point of view, there is no need to use a micro-simulation for the representation of the traffic dynamics. Indeed, traditional assignment models such as emme/2 (see below) rely on link delay functions, i.e. link travel times depend directly on the demand. It is clear though that this approach becomes dynamically wrong once demand is higher than capacity, and queue spill-back spreads through the system [8, 9]. In how far this is important in practice remains an open question. But since metropolitan regions are becoming ever more congested, one should develop and test a methodology that includes a dynamically correct representation of congestion. Monte-Carlo simulations are a common approach to deal with complex systems such as a congested traffic network. Yet, since they often exhibit non-linear behavior, they are unlikely to be replaced by an analytic treatment.

A large scale approach to this problem is currently pursued by the TRANSIMS project [10]. The next TRANSIMS case study will attempt to simulate the whole city of Portland (1.5 million people) on the level of activities generation, on the level of modal choice and route planning, and on the level of the transportation and traffic dynamics. The main difference to most other projects will be that on *all* levels the approach will model individual people. The advantage of this is that the approach remains conceptually extensible since the behavioral rules of the individuals are directly accessible. The challenge with this approach is computational, since the problem is not only big (1.5 million individuals as said above, and also 200 000 links in the transportation network for a realistic representation), but the relaxation iterations means that the micro-simulation needs to be run many times (about fifty times for one relaxation).

TRANSIMS approaches this problem via a combination of fast hardware and a computationally relatively fast traffic simulation approach [11]. Nevertheless, fast hardware is not always available, such code is data intensive, and it takes time to write. Mostly for the last reason, TRANSIMS itself uses simplified micro-simulations in order to test the other modules (route planner, activities planner) and in order to test the interactions (feedback) between the modules. This paper presents such a simplified micro-simulation. Its input and output are compatible with the TRANSIMS framework. Thus, with a model such as this one, it is possible to use a dynamic, activities-based approach to transportation simulation (such as in TRANSIMS) with affordable data and hardware. If nothing else, this should at least enable systematic research about the coupling of demand generation, route assignment, and micro-simulation output. In how far the results will be quantitatively useful remains to be seen.

In this paper, we will (Sec. 2) present our simple approach to traffic dynamics in a road network, such as a metropolitan area. The dynamics of the model concentrates on the two arguably most important elements of congestion: capacity cut-offs, and queue spill-back. Capacity cut-offs are modelled by not letting more vehicles leave a link per time slice than is possible according to that link's capacity; and queue spill-back is modelled by a "storage" constraint, i.e. a link can only absorb a limited number of incoming vehicles. This is followed by a short summary of results for a Dallas scenario (Sec. 3) and by a description of simulation results for Portland (Sec. 4). The paper is concluded by a discussion and a summary.

2 Different models based on queueing theory

2.1 "Traditional" queueing models

Queueing theory in its simplest form assumes that a service is provided with a certain rate, and that requests for the service come in with a certain, usually different, rate. Clearly, when the request rate is higher than the service rate, a queue for requests will form. Although the precise dynamics are more complicated, this is also the essential mechanism of capacity cut-off of an urban street link. The service rate here is generated by a traffic light; and in most cases, the traffic that can potentially come in towards the traffic light can be higher than the traffic that can go through the traffic light.

It is maybe obvious that links composed of queues can be connected to reflect regional transportation networks. The service rate of each link/queue would then correspond to the capacity of that link. An example for such an approach is Ref. [12].

The main question about such models is if they contain the minimal description necessary to reproduce microscopic traffic characteristics. In fact, the introduction of queues with infinite storage does not allow the creation of spill backs, so typical of a crowded network. Spill backs are caused by links that become full, which happens when demand is higher than capacity, i.e. more vehicles enter the link than can leave. Full links do not accept any further vehicles, thus clogging up links which contain vehicles that want to enter the full link. In this way, a single link where demand exceeds capacity can cause congestion to spread through a network.

2.2 Gawron's model

In 1997, Gawron introduced a model similar to Simao and Powell's model in its architecture [13, 8]. The main difference from the "traditional" queueing models resides in the modelling of spill-backs. The number of vehicles leaving a link is constrained by the capacity of the link, *and* by the number of cars which can fit on the destination links. If its destination link is full, a vehicle will stay where it already is.

Each time a car enters a link, an expected travel time is calculated. An early version of this model proposed to calculate this travel time from the length and the current state of the link. A fundamental diagram proposes a desired velocity according to the current density [13]. A newer version only assumes to consider the free flow velocity to calculate the travel time [8]. We will use the same simulator in this paper and present new results on the Portland network.

2.3 Fundamental diagram

One of the most important features of the queueing model is to produce reasonable travel times in the laminar and congested regimes. A fundamental diagram can be extrapolated from the parameters of the model itself. Let us consider a queue with free flow velocity v_0 in (m/s). We call L the length of the link, C its capacity $(vehicles/second)$ and n_{lanes} the number of lanes of the link. The maximum number of vehicles that can be added to the link is $N_{sites} = L \cdot n_{lanes}/\ell_{site}$ where we set the space taken by one vehicle in a jam to the inverse of the jam density: $\ell_{site} = 1/\rho_{jam}$. For this paper, l_{site} is set to 7.5 meters. The number of sites of the link, N_{sites}, is also the maximum number of vehicles in the queue. Free flow travel time is given by the relation $T_0 = L/v_0$.

For illustration, let us now put N_{veh} vehicles in the queue at time t and suppose that when a vehicle is allowed to leave the link, it is automatically put at the end of the same queue ("traffic in a loop with one stop light"). We therefore keep the density $\rho := N_{veh}/N_{sites}$ constant and define different regimes according to the density. Note that density here is unitless because we express lengths in terms of numbers of cells; also, time will be unitless because we express it in terms of the number of simulation time steps. There are three regimes:

- **Laminar regime.** In the laminar regime, demand is smaller than capacity. In the queueing model case, it means that nobody spends more than T_0 seconds on the link. The average velocity is simply given by v_0.

- **Capacity regime.** As soon as the build-up of the queue is longer than what the capacity of the link can dissolve in T_0 seconds, we can consider the queue to be in the congested regime. In our closed system, this simply means that the first vehicle released from the queue after the start of the simulation will have moved around the loop and be ready to leave again (according to T_0) before the vehicle in front of it has left. Since at this point $C\,T_0$ vehicles have left the queue, the critical density for which vehicles begin to wait longer than T_0 is given by

$$\rho_1 = \frac{C \cdot T_0}{N_{sites}} \ .$$

  When the density ρ is higher than ρ_1, the expression of the travel time can be given by $T = T_0 + (N_{veh} - C \cdot T_0)/C = N_{veh}/C$, which simply means that one can leave the link once all N_{veh} vehicles in front have left the link, and this takes a time of N_{veh}/C.

This leads to the expression for the velocity in the capacity regime:

$$v = \frac{L}{T} = \frac{\ell_{site}\, N_{sites}}{n_{lanes}} \cdot \frac{C}{N_{veh}} = \ell_{site} \cdot \frac{C}{n_{lanes}} \cdot \frac{1}{\rho} =: \frac{K}{\rho} \qquad (1)$$

where $K = \ell_{site}\, C/n_{lanes}$. The velocity in the capacity regime is thus a product of a model parameter (ℓ_{site}), a link parameter (C/n_{lanes}), and the inverse of the density. – Note that some traffic flow models have $v \propto \frac{1}{\rho} - \frac{1}{\rho_{jam}}$ as their functional form [14], where ρ_{jam} is the density of traffic in a jam.

- **Jammed regime.** The velocity goes towards zero when the queue is almost full and vehicles have difficulties leaving the queue because there is no space available on the destination link. For this, it is easier to imagine that the closed loop is composed of two links. First, one of the links is picked. Vehicles leave until all empty spaces in the second link are filled up, and vehicles are moved forward on the first link. Then, the same happens for the second link. Clearly, the number of vehicles n that leave the link per time step is the same as the number of empty sites, i.e. $n = n_{empty}$. n is also the same as the flow, i.e. $n = q$; remember that times are unitless since they are given in simulation time steps; for that reason, q is unitless, too. Since the density is $\rho = (N_{sites} - n_{empty})/N_{sites}$, one obtains $q = N_{sites}\,(1-\rho)$. The link is in this regime for

$$\rho \geq \rho_2 = 1 - C/N_{sites} \; .$$

A typical fundamental diagram would look like Fig. 1. Velocities in the queueing model do not go "smoothly" to zero for $\rho \to 1$, as for example a function $v \sim \frac{1}{\rho} - \frac{1}{\rho_{jam}}$ would do; instead, they have a "kink" at $\rho = 1 - C/N_{sites}$. The velocity here is $K \cdot L/(L-K)$, where K is the same as defined after Eq. 1. This means that if the link is long enough, this value is close to K, which depending on the characteristics of the link, is not necessarily close to zero.

The physical reason for this is that, in the queue model, "holes" can travel in one time step from the beginning of the link to its the end. This is opposed to real traffic, where, say, a light turns green, then the first car moves and opens up space for the second, then the second car moves and opens up space for the third, and it takes quite some time until this effect has travelled up a link.

The consequence of this behavior for traffic simulation purposes is that simulated traffic will be more "fluid" in the very congested regime than when using a model where speed goes "smoothly" to zero for $\rho \to 1$. Having somewhat fluid traffic in the very congested regime is not necessarily a disadvantage since, in a network context, current simulation models seem to grid-lock more easily than reality [15, 16].

2.4 Algorithm of the queueing model

The description in the last section should be sufficient to implement a model similar to ours. Nevertheless, in this section we present a more precise description of our algorithm.

- FOR all links DO:
  - WHILE (vehicles can still leave in this time step according to the capacity) DO

* Look at the first vehicle in the queue.
* If the free flow speed arrival time is larger than the current time, then break and go to the next link.
* Check if the destination link has space. If not, break and go to the next link.
* Calculate the expected arrival time at the end of the next link:
  Arrival time = Current time + length/freeFlowSpeed
  (length and freeFlowSpeed of the destination link)
* If passing is allowed, insert vehicle into the destination queue sorted by time
* If passing is not allowed, insert vehicle at the end of the destination queue

ENDWHILE

ENDFOR

The precise condition for "can still leave in this time step according to the capacity" is

$$N_{link} < int(C_{link}) \text{ OR } [\ N_{link} = int(C_{link}) \text{ AND } rnd < C_{link} - int(C_{link})\ ] \quad ,$$

where C_{link} the capacity of the link in vehicles/simulation-time-step, N_{link} is the number of vehicles which already left the link during the same time step, rnd is a random number between 0 and 1, and $int(x)$ is the integer part of x.

Note that the simulation runs on pre-computed route plans, as explained below. Such a simulation can become "stuck" or grid-locked, for example when a loop of full links forms, and the first car on each of these links wants to move into another of these full links [15, 16]. In order to prevent this, we remove vehicles that are first in a queue and have not moved for T_{wait} time steps of the simulation. For the simulations in this paper, we used $T_{wait} = 300$ seconds (= 300 simulation time steps). In the iterative procedure (explained later), many such vehicles were removed in the first iterations, but their number is less than 0.5% in the 40th iteration (Fig. 2 c). This models an extremely simplified version of on-line re-routing: Travellers who are stuck in congestion eventually decide to try something different. The simplest version of "something different" is to park the car, which is what we model here.

3 Previous results on a Dallas case study

We compared the queue model (QM) with two other, more realistic micro-simulations in the context of the TRANSIMS Dallas–Fort Worth case study [17]. These micro-simulations were PAMINA [18, 15] and the TRANSIMS micro-simulation [16, 19]. Comparisons of link densities and of accessibility can be found in [20]; comparisons of turn counts (also with field data) can be found in [21]. For these studies, all three simulations used the same trip table (origin–destination matrix), and they used the same router for iterations between simulation and re-routing. The major result of these comparisons was that the results of all three simulations were remarkably similar. This indicates to us that deviations from reality are at the current stage of research probably to a larger extent caused by the travel demand generation algorithms and by the routing algorithm than by the micro-simulations.

4 Simulation results on Portland

In this paper, we want to concentrate on results for Portland (Oregon), which is the study area for the next TRANSIMS case study.

4.1 Activities and iterative replanning

TRANSIMS [10], in its full design, uses data on demographics and on transportation infrastructure as input. The following steps are then performed:

- **Synthetic population disaggregation:** As the first step, TRANSIMS generates a synthetic population from the demographic data [22], that is, TRANSIMS looks at synthetic individuals and their decision-making process rather than at the behavior of aggregated quantities. The synthetic individuals possess many relevant attributes such as the number of persons living in the same household or the number of cars per family.

- **Activities generation:** Travel demand is generated via activity patterns and activity locations. This synthetic population is then combined with the land use data to produce activity assignements. This basically tells us where a person works and what her other schedules are.

- **Modal and route choice:** A modal choice and routing module generates explicit "travel plans" for each synthetic individual.

- **Travel:** The micro-simulation (such as the queue model presented in this paper, or more realistic micro-simulations) executes all travellers' plans simultaneously and thus computes the nature of the interactions between travellers, especially congestion.

It is well-known that the above steps cannot be performed uni-directionally because backward causalities exist. For example, congestion will make people change their mode of transportation and/or their routes. If switching mode or route does not help enough, they may change their activity locations and/or their activity schedule.

TRANSIMS (and several other projects, e.g. [6, 4, 7, 8, 5]) approach this problem via *feedback*, i.e. iterations between the modules. Initial activities and travel plans are generated, the micro-simulation runs based on these plans, some synthetic travellers change modes and/or routes, the micro-simulation is run again, etc., until some stopping criterion is fulfilled.

For the Dallas case study, the activities generation module was not yet in place. Thus, the Dallas case study used conventional trip tables as a starting point and essentially performed an assignment [23] of the trip table on the network, except that the trip tables were explicitly time-dependent, the assignment was performed on the level of individual drivers, and the dynamics of congestion and queue spill-back (and much more) were explicitely and realistically represented.

The activities used for *this* paper came from a simple home-to-work trip generation module [24]. This is not done with the intention of being as realistic as possible but with the intention of understanding the dynamics of the computational process by using a simplified partial problem. The input data here are (i) a list of all synthetic individuals in the simulation who work, and (ii) a list of all workplaces. The problem is to match workers and workplaces, thus generating home-to-work trips. This assignment is done using a distance-based preference function [24]. The version that is the basis for this paper is in fact a very simple gravity model (e.g. [25]), that is, the probability to accept a workplace at travel time τ is a function of τ. In our case, we take this function as proportional to $1/\tau$. Since we have approximately 500 000 workers, this activity set leads to approximately 500 000 trips, all between 4am and 10am. Improvements in this module are expected both because of the use of high sophisticated activity-based modules [26, 27] and because of further improvements of our simplified home-to-work method [28].

Routing is done using time-dependent fastest path. Link travel times are given in 15-minute aggregates from the last iteration of the microsimulation. For the initial route plan set, free flow speeds on the links are used. Travellers only have cars available; the integration of other modes is currently being done but not yet operational. For route re-planning, we only change routes for a fraction of the population. This fraction is approximately 5%, and selection is done with a heavy bias towards individuals who have not re-planned their route for a large number of iterations ("age-dependent re-planning", [18]). By iterating this process, we reach a relaxed state, where no more changes are observed from one iteration to the next, except for random fluctuations. For morning peak simulations, this typically takes 20 to 40 iterations, see Fig. 2.

The network that we used was the same that the Portland Metropolitan Planning Organization ("Portland Metro") uses for their emme/2 runs (see below). The important information for our model were: length of links, capacity of links, free flow speeds or speedlimits, and storage capacity for full link (computed via length and number of lanes). Except for the storage capacity, this is the same information that is used for traditional assignment. The network has 20,024 uni-directional links and 8,564 nodes.

We simulated traffic between 4am and noon; in order to simulate these 8 hours with half a million trips on the above network, we needed about 17 minutes on a 250 MHz SUN UltraSparc CPU. Computational speed on a Pentium CPU should be about the same or faster. Routers can compute more than 100 trips per second in the same situation, which means that re-routing 5% of 500 000 trips takes less than 5 minutes. This means that one iteration takes about 22 minutes, and the 40 iterations necessary for relaxation take about 15 hours, all on a single CPU.

On the same hardware, the emme/2 assignment shown in Fig. 3 would take about 2 hour of computing time [29], that is, the same study would be about 8 times faster than the queue simulation approach described in this paper. Remember, however, that we want to be able to represent time-of-day dynamics, which emme/2 does not do.

The full TRANSIMS microsimulation [10] runs about 40 times slower than the queue microsimulation [30], i.e., the same study would take 25 days. There is, however, a parallel version of the TRANSIMS microsimulation available [30],, which is not the case for the other methods.

4.2 Field data

We had traffic count data available for about 450 locations. These locations appear along a set of "cutlines" that have been identified by the Portland Metro travel forecasting staff. The cutlines are established to measure the major flows into and out of a central area, or between suburban and downtown, or suburban and outlying commercial areas. These cutlines and locations are believed to be significant in measuring traffic volumes for the region, and have been in use for several years. Since all of our other data refer to the year 1994, we will also only use field data for that year.

The data are typically collected for AM and PM peak periods, and during each peak period for a one-hour and a two-hour interval. In this paper, we will concentrate on the one-hour data, which are typically collected either between 7 and 8 am, or between 7:15 and 8:15 am. The difference between the one-hour and two-hour flow rates is not enormous and will be neglected for this paper. More details on these data will be published elsewhere [31].

4.3 emme/2 results

For comparison, we also have results from a Portland emme/2 [32] AM peak assignment available (Fig. 3). emme/2 is a computer implementation of the analytical equilibrium assignment technique (e.g. [23]).

The Mathematical Programming formulation of equilibrium assignment is as follows: Assume that the origin-destination matrix q^{rs} is given. Also given is the function which computes link travel times from link flows:

$$t_a = t_a(x_a) \ , \tag{2}$$

where a is a link index, and x_a the link flow. For each origin-destination entry q^{rs} that is larger than zero, there is at least one path that connects r and s. Paths are numbered by the index k. f^{rsk} is the flow on the k-th path connecting r and s; c^{rsk} is the cost of this path. Then we have

$$c^{rsk} = \sum_a t_a \, \delta_a^{rsk} \ ,$$

that is, the cost of path rsk is the sum of the travel times of all links that are used by that path. Also,

$$x_a = \sum_{rsk} f^{rsk} \, \delta_a^{rsk} \ ,$$

i.e. the flow on a link is the sum of all path flows that use that link. An equilibrium assignment is now obtained by solving the following optimization problem:

$$\min \, z(x) := \sum_a \int_0^{x_a} d\omega \, t_a(\omega) \tag{3}$$

subject to

$$\sum_k f^{rsk} = q^{rs} \ , \ \ f^{rsk} \geq 0 \ . \tag{4}$$

There is no direct interpretation of what this means; but one can show that the solution of this Mathematical Program is a Nash Equilibrium of the network assignment problem. Nash Equilibrium means that no path f^{rsk} can reduce its costs by unilaterally switching to a different path.

The important points here is that the whole formulation describes a *state*, with no dynamics involved. There is no dependency on the time-of-day: OD flows q^{rs} flow with a constant rate via paths rsk from their origins to their destinations; the definition of the cost of using a link, Eq. 2, only makes sense when these rates remain constant. Such formulations cannot treat congestion since in the congested regime there are two possible travel times t_a for most flow rates, and one of them, the congested one, does not generate a constant rate solution since it implies backspill of congestion. In that sense, traditional assignment is not a simulation, since it does not have a time-of-day dynamics. Note that most practitioners run separate steady-state assignments for the morning peak and the afternoon peak.

The results were again provided by the Portland MPO (Portland Metro) and were run on the same network that we were using. Yet, we are not using the same origin-destination table as the emme/2 assignments since we are generating transportation demand via activities as described above. For that reason, differences between our model results and Portland Metro's model results can be caused by all different modules involved, i.e. the travel demand generation, the routing, or the model for the traffic dynamics. A study that investigates this into more detail would certainly be useful, but it is beyond the scope of this paper.

4.4 Queue model simulation results

Fig. 4 shows results of our microsimulation-based assignment using the simplified home-to-work activities. As mentioned earlier, the simulations run on the same network as the emme/2 assignment except that some additional information is needed in order to get the storage capacity of the links. We show results of the 40th iteration of the feedback process.

The plots show average hourly speeds on all links according to the legend; dark links have low speed, usually caused by congestion. Links are underlaid in light gray, with the width corresponding to their capacity, in order to identify high-capacity links.

Clearly, even after many iterations, there remains a significant number of bottlenecks that prevents traffic from going to their destinations. This is most probably due to an over-estimation of the traffic demand in the demand generation module, which was not part of this study. As stated above, improvements in this module are expected both because of the use of high sophisticated activity-based modules [26, 27] and because of further improvements of our simplified home-to-work method [28]. However, another effect is that simulations such as ours cannot "push" demand through bottlenecks at a rate higher than capacity. So when demand at bottlenecks is higher than capacity, traffic will jam up, and the queues will spill back, delaying other parts of the network, and potentially leading to grid-lock. This is in contrast to traditional assignment, where bottlenecks show up as links with a so-called V/C-ratio of larger than one (which is physically impossible), without any direct dynamic effect on other parts of the network. Fig. 3 indeed shows several such links.

Fig. 5 shows comparisons to the emme/2 assignment for the 7-8am period. The shown volumes are hourly volumes. There are two dominant features when compared to the emme/2 assignment: (i) we have much more traffic northbound across the bridges. This comes out of assigning too many workplaces in Washington to workers living in Oregon, see [24]. Better sets of workplace assignments should correct this. (ii) We have significantly *less* flow on I-84 westbound, which is the somewhat zigzagging line extending east from downtown. Further inspection yields that this is *not* the result of not enough traffic but rather of too much traffic, which jams up when it tries to reach the bridges across the Williamette River and then spills back into the interstate. This is exactly the effect described above, that bottlenecks lead to queues, which spill back through the system. Similar effects, although to a lesser extent, can be found on some other of the freeways, especially where they merge and upstream from there.

As stated above, this effect is a consequence of a more realistic representation of one aspect of traffic dynamics, i.e. queue spillback. However, it seems that we are over-estimating the queue formation, i.e. we are predicting speeds that are too low when compared to reality. In order to obtain some preliminary insight into how close we are to reality, a comparison between simulation results and actual counts, both for 1994, is shown in Fig. 6. The emme/2 results are included for comparison. One notices that, in the average, our simulation results systematically underpredict traffic flows, wheras the emme/2 results seem less biased. Apart from that, we think that the quality of our results is encouragingly good when compared with the emme/2 results and when taken into account how simple our method is.

Tables 1 and 2 quantify this statement. The two tables compare the field data with the emme/2 results and with the queue model results, respectively. The data are aggregated into classes according to the field flow measurement. The first column shows these classes, in units of vehicles/hour. The second column shows the number of entries for each class.

The third and fourth columns show the mean absolute and relative bias in each of the classes, i.e.

$$b_{abs} = \frac{1}{N_c} \sum_i (x_i - \hat{x}_i) = \frac{1}{N_c} \left(\sum_i x_i - \sum_9 \hat{x}_i \right)$$

and

$$b_{rel} = \frac{b_{abs}}{\langle \hat{x} \rangle_c}$$

where N_c is the number of links in a class, x_i is the flow from the model for link i, $\hat{x}_i$ is the flow in the field data, and the sum goes over all links in the class. $\langle \hat{x} \rangle_c$ is the mean value for the field data for class c. Positive values for b_{abs} here mean the models are systematically over-estimating flow. – And indeed, our observation regarding systematic bias is confirmed: Our simulations systematically underestimate counts except for links with little traffic. emme/2 systematically over-estimates, but to a lesser degree.

The fifth and sixth columns show the mean deviation from the field data, i.e.

$$d_{abs} = \frac{1}{N_c} \sum_i |x_i - \hat{x}_i|$$

and

$$d_{rel} = \frac{d_{abs}}{\langle \hat{x} \rangle_c} \; .$$

Interestingly, in this value the quality of our simulation results and of the emme/2 results are comparable except for the links with the highest flows. We attribute this to our "jams on the interstates" as explained above, and we expect this to get better with somewhat more sophisticated methods for demand generation [31].

The seventh and eighth columns show the root mean square deviation from the field data, i.e.

$$var = \sqrt{\frac{1}{N_c} \sum_i (x_i - \hat{x}_i)^2}$$

and

$$\sigma = \frac{var}{\langle \hat{x} \rangle_c} \; .$$

Again, our results have a similar quality as the emme/2 results, except for the class with high flows.

5 Summary

We presented a simplified traffic micro-simulation. This micro-simulation is consistent with a microscopic approach to demand generation, that is, it operates on individual route plans. The approach is based on the simulation of queues, where vehicles can leave a queue only if capacity restrictions allow it and if there is space on the destination link. In addition, a vehicle needs to spend at least the free flow speed travel time on a link. This means that situations where demand is larger than capacity automatically lead to queue formation, and because of limited "storage" capacity on the link, the jam will eventually spill back through the network. The model uses emme/2 type network data as input, plus the number of lanes to compute the storage capacity.

We showed a first application of this model for Portland. The emme/2 network that Portland Metro uses has about 20 000 links. We generated about half a million home-to-work trips by a simple heuristic approach and ran them through this network. Feedback iterations were used to relax the routes; results of the final iteration were shown and compared to emme/2 results and to field data. We are encouraged by the observation that, in spite of the simplicity of our approach and except for high flow links, our results have about the same quality as the emme/2 results. We attribute the deviations on high flow links to an overestimation of demand for our model, and we expect this to be improved with more sophisticated demand models, one of them [28] only slightly more complicated than the one used here.

We think that the fact that we can run the complete microscopic dynamic assignment of half a million trips on a 20 000 link network in about half a day on a single workstation CPU is an astounding achievement. Clearly, technology has enabled us to make a big jump in what is feasible, and much work remains to be done to make these opportunities useful for transportation planning.

Acknowledgments

This work was performed at Los Alamos National Laboratory, which is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36 (98-3903). We greatly appreciate the help of the people at the Portland Metropolitan Planning organization, especially Keith Lawton, Bill Stein, and Dick Walker, for providing us with extensive high-quality data, out of which only a small portion was used for this paper.

References

[1] John L. Bowman. *The day activity schedule approach to travel demand analysis.* PhD thesis, Massachusetts Institute of Technology, Boston, MA, 1998.

[2] D. Ettema and H. Timmermans, editors. *Activity-based approaches to travel analysis.* Pergamon, 1997.

[3] J. Bottom, M. Ben-Akiva, M. Bierlaire, and I. Chabini. Generation of consistent anticipatory route guidance. In *Proceedings of TRISTAN III*, volume 2, San Juan, Puerto Rico, June 1998.

[4] Alfredo Garcia and Robert L. Smith. On the convergence of iterative routing-assignment procedures in dynamic traffic networks. Technical Report 95-26, University of Michigan Department of Industrial and Operations Engineering, Ann Arbor MI 48109, November 1995.

[5] C. Cantarella and E. Cascetta. Dynamic process and equilibrium in transportation network: towards a unifying theory. *Transportation Science A*, 25(4):305, 1995.

[6] DYNAMIT, see http://its.mit.edu.

[7] H.S. Mahmassani, T. Hu, and R. Jayakrishnan. Dynamic traffic assignment and simulation for advanced network informatics (DYNASMART). In N.H. Gartner and G. Improta, editors, *Urban traffic networks: Dynamic flow modeling and control*. Springer, Berlin/New York, 1995.

[8] C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(3):393, 1998.

[9] M. Florian. Keynote presentation. Presented at TRISTAN-III conference, San Juan, Puerto Rico, June 1998.

[10] TRANSIMS, TRansportation ANalysis and SIMulation System, Los Alamos National Laboratory, Los Alamos, U.S.A. See `transims.tsasa.lanl.gov`.

[11] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221, 1992.

[12] H.P. Simão and W.B. Powell. Numerical methods for simulating transient, stochastic queueing networks. *Transportation Science*, 26:296, 1992.

[13] C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. In D.E. Wolf and M. Schreckenberg, editors, *Traffic and granular flow'97*, pages 469. Springer, Heidelberg, 1998.

[14] D.L. Gerlough and M.J. Huber. *Traffic Flow Theory*. Special Report No. 165. Transportation Research Board, National Research Council, Washington, D.C., 1975.

[15] M. Rickert and K. Nagel. Experiences with a simplified microsimulation for the Dallas/Fort Worth area. *International Journal of Modern Physics C*, 8(3):483, 1997.

[16] K. Nagel and C.L. Barrett. Using microsimulation feedback for trip adaptation for realistic traffic in Dallas. *International Journal of Modern Physics C*, 8(3):505, 1997.

[17] R.J. Beckman et al. TRANSIMS–Release 1.0 – The Dallas-Fort Worth case study. Los Alamos Unclassified Report (LA-UR) 97-4502, Los Alamos National Laboratory, see `transims.tsasa.lanl.gov`, 1997.

[18] M. Rickert. *Traffic simulation on distributed memory computers*. PhD thesis, University of Cologne, Cologne, Germany, 1998.

[19] K. Nagel, P. Stretz, M. Pieck, S. Leckey, R. Donnelly, and C.L. Barrett. TRANSIMS traffic flow characteristics. Los Alamos Unclassified Report (LA-UR) 97-3530, Los Alamos National Laboratory, see www.santafe.edu/~kai/papers, 1997. Earlier version: Transportation Research Board (TRB) preprint 981332.

[20] K. Nagel, M. Rickert, and P.M. Simon. The dynamics of iterated transportation simulations. Los Alamos Unclassified Report (LA-UR) 98-2168, Los Alamos National Laboratory, see www.santafe.edu/~kai/papers, 1998. Presented at the 3rd TRIannual Symposium on Transportation ANalysis (TRISTAN-III) in San Juan, Puerto Rico.

[21] K. Nagel, M. Pieck, P.M. Simon, and M. Rickert. Comparison between three different micro-simulations and reality in Dallas. Los Alamos Unclassified Report (LA-UR) 98-2944, Los Alamos National Laboratory, see www.santafe.edu/~kai/papers, 1998.

[22] R.J. Beckman, K.A. Baggerly, and M.D. McKay. Creating synthetic base-line populations. *Transportion Research Part A – Policy and Practice*, 30(6):415, 1996.

[23] Y. Sheffi. *Urban transportation networks: Equilibrium analysis with mathematical programming methods*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1985.

[24] P. Wagner and K. Nagel. Microscopic modeling of travel demand: Approaching the home-to-work problem. Annual Meeting Paper 99 09 19, Transportation Research Board, Washington, D.C., 1998.

[25] D. Lohse. *Verkehrsplanung*, volume 2 of *Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung*. Verlag für Bauwesen, Berlin, 1997.

[26] M. Bradley. Personal communication.

[27] P. Speckman. Personal communication.

[28] J. Esser and K. Nagel. Census-based travel demand generation for transportation simulations. In M. Schreckenberg al, editor, *Proceedings of the workshop "Traffic and Mobility*, Aachen, Germany, Sep/Oct 1998. Also Los Alamos Unclassified Report LA-UR 99-10, see www.santafe.edu/~kai/papers.

[29] Bill Stein from Portland METRO. personal communication.

[30] M. Rickert and K. Nagel. Dynamic traffic assignment on parallel computers. to be submitted to: Proceedings of a Traffic Workshop at HPCN'99 in Amsterdam.

[31] J. Esser and K. Nagel. In preparation.

[32] A. Babin, M. Florian, L. James-Lefebvre, and H. Spiess. Emme/2: Interactive graphic method for road and transit planning. *Transportation Research Record*, (866):1, 1982.

class	n	mean bias	mean err	RMS err
<250	99	93 (64%)	155 (106%)	222 (152%)
250-500	121	-1 (-0%)	180 (49%)	240 (65%)
500-750	83	-87 (-14%)	227 (37%)	291 (47%)
750-1000	42	-165 (-19%)	272 (31%)	342 (39%)
1000-1500	57	-196 (-17%)	327 (27%)	462 (39%)
>1500	25	-615 (-26%)	776 (32%)	1038 (43%)

Table 1: Quantitative comparison between our model results and counts from reality. The results are separated into classes, shown in column 1, according to the field data value. The unit is vehicles/hour. The second column shows the number of entries for each class. The third and fourth columns show the mean absolute and relative bias in each of the classes. Negative numbers mean that the model systematically under-estimates flows in this class. The fifth and sixth columns show the average deviation from the field data, whereas the seventh and eight columns show the root mean square deviation from the field data.

class	n	mean bias	mean err	RMS err
<250	99	85 (58%)	148 (101%)	263 (180%)
250-500	121	61 (17%)	194 (53%)	258 (70%)
500-750	83	57 (9%)	213 (34%)	301 (49%)
750-1000	42	104 (12%)	317 (37%)	379 (44%)
1000-1500	57	141 (12%)	366 (31%)	477 (40%)
>1500	26	-61 (-3%)	504 (22%)	647 (28%)

Table 2: Comparison between field data and emme/2 results obtained from the Portland Metropolitan Planning Organization from their "standard am-peak 1994" run. The columns are the same as in Table 1. These results have fewer bias than our simulation runs (column three), and they are better for the highest flow counts. Apart from this, the quality of the quantities in the table looks similar.

Figure 1: Fundamental diagrams flow vs. density and velocity vs. density for the Q-model when run in a closed loop. Capacities C here are given unitless in "number of vehicles per simulation time step".

Figure 2: Different indicators of routing relaxation. (a) Sum of all travel times vs. iteration number. (b) Number of vehicles in the simulation as function of the time-of-day. Different curves for different iteration numbers. (c) Number of removed cars. As explained in the text, cars that are first in the queue but do not move for T_{wait} time steps because their destination link is full are removed from the simulation. Note that the state in iteration 40 is in between iteration 20 and iteration 30, indicating that we are in the fluctuating steady-state regime.

Figure 3: Result of the emme/2 assignment. The width of the light gray denotes capacity. "emme/2 volume"-to-capacity ratios 0.5–1, 1–1.2, and > 1.2 are marked as indicated in the legend.

Figure 4: Result of our own route assignment using simplified home-to-work trips, and feedback iteration between a fastest path re-planner and the queue model (QM) micro-simulation. The width of the light gray denotes again capacity. Average hourly speeds are marked as indicated in the legend. Averaged from 7 to 8 am.

Figure 5: Using the same simulation results as in Fig. 4 and comparing them to the emme/2 assignment. The width of the light gray denotes again capacity. Differences between our simulation results and the emme/2 assignment flow results are marked as shown in the legend; positive values mean that we have more flow.

Figure 6: Comparison of: (i) field data from Portland (black line); (ii) our simulation results (vertical spikes); (iii) results from the Portland emme/2 run (crosses). The x-axis is an arbitrary numbering of the measurement sites; y-axis shows hourly flows. The sites are ordered according to their field data hourly flow values. Our simulation results are in general too low when compared to the counts, wheras the emme/2 results seem to be more correct in the average. However, the quality of the solutions seems comparable. This statement is quantified in Tables 2 and 1.

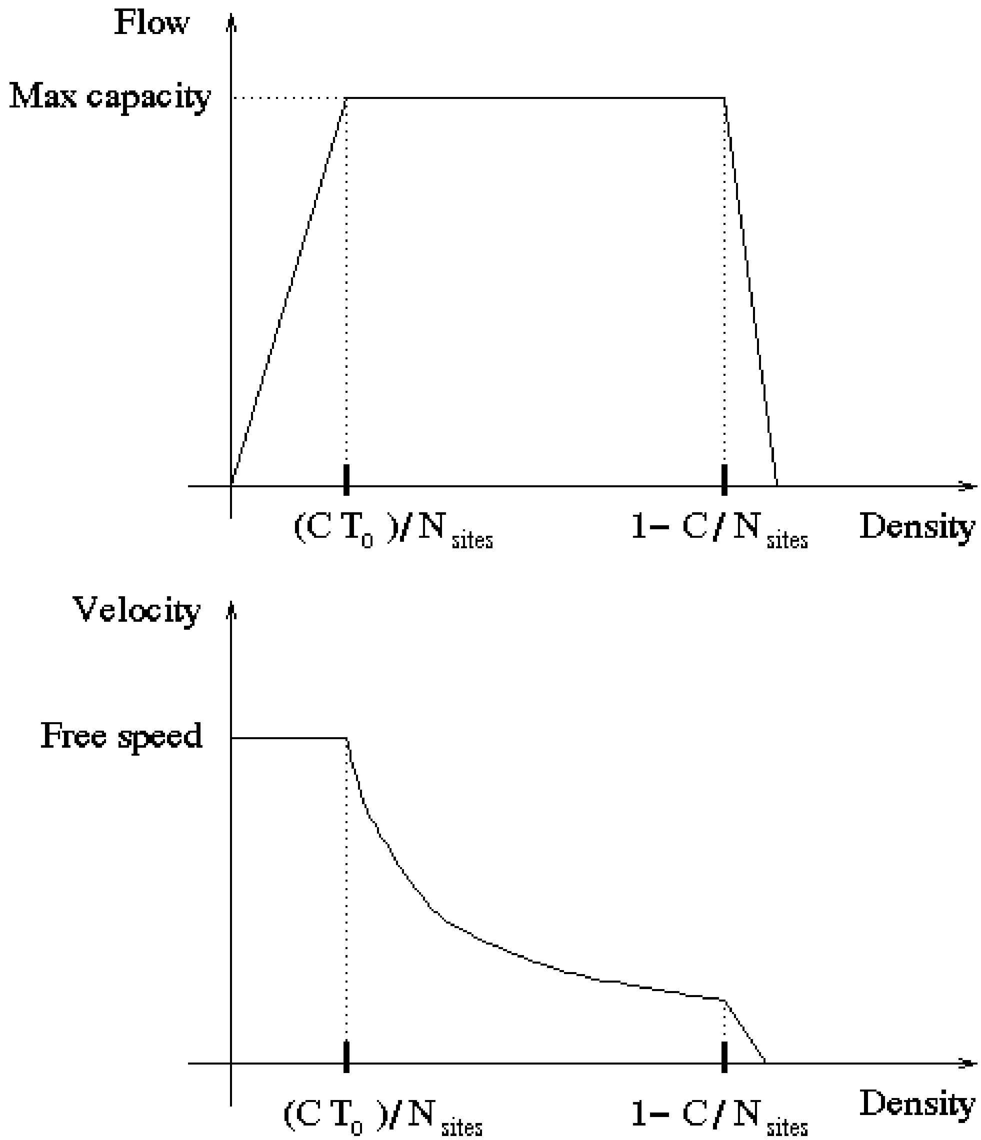

Flow
Max capacity
(C T0)/ N sites
1− C / N sites
Density
Velocity
Free speed
(C T0)/ N sites
1− C / N sites
Density

Fig. 1

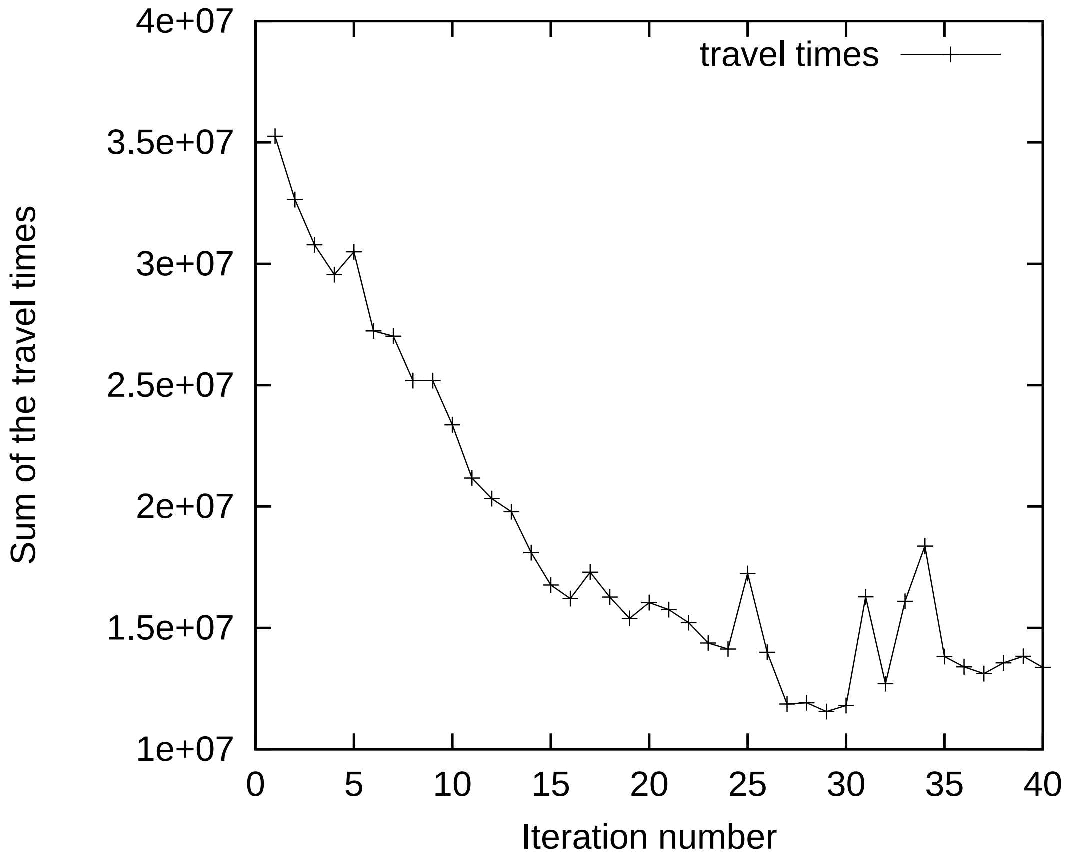
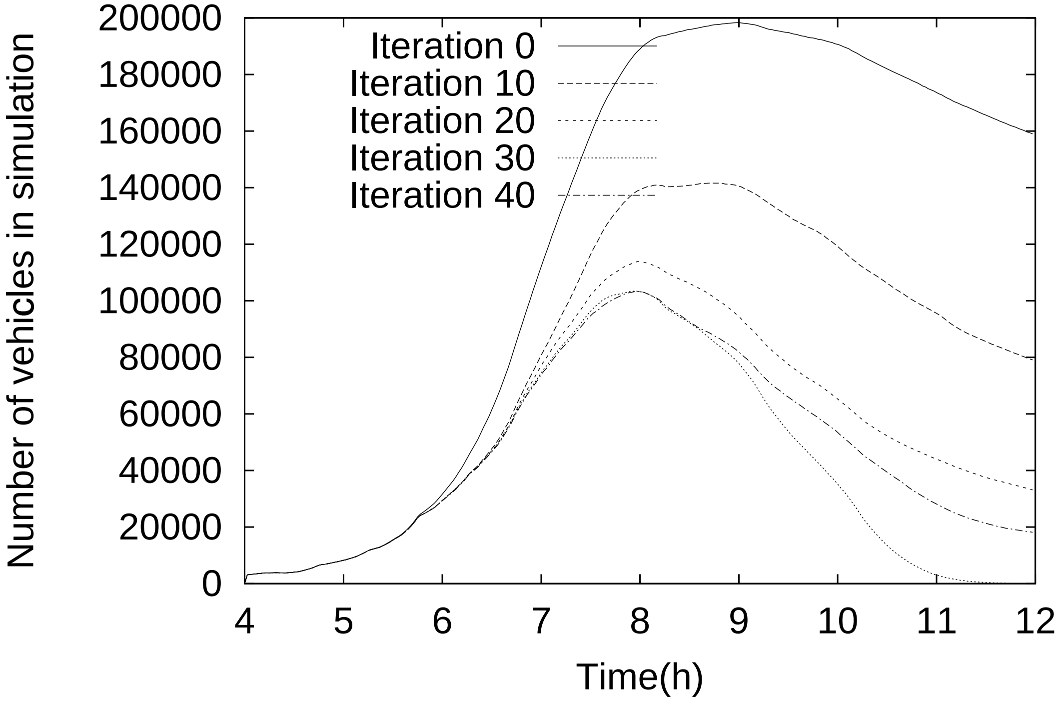

travel times
Sum of the travel times
Iteration number
4e+07
3.5e+07
3e+07
2.5e+07
2e+07
1.5e+07
1e+07
0
5
10
15
20
25
30
35
40

Fig. 2 a



Number of vehicles in simulation
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
4
5
6
7
8
9
10
11
12
Time(h)
Iteration 0
Iteration 10
Iteration 20
Iteration 30
Iteration 40

Fig. 2 b

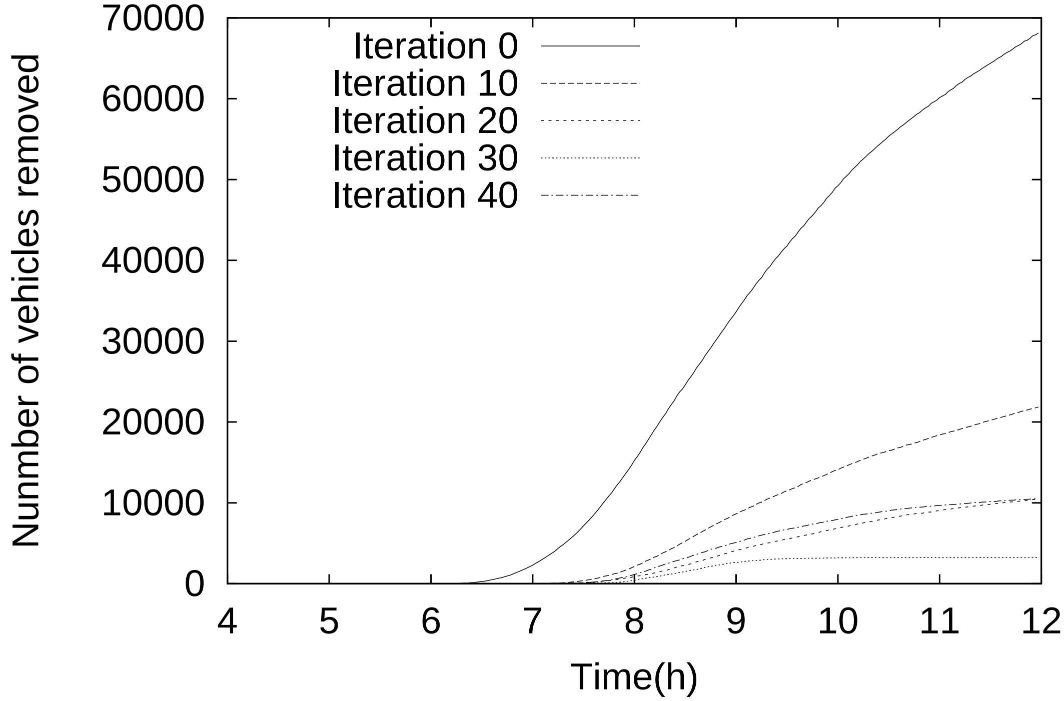

70000
60000
50000
40000
30000
20000
10000
0
Nunmber of vehicles removed
Iteration 0
Iteration 10
Iteration 20
Iteration 30
Iteration 40
4
5
6
7
8
9
10
11
12
Time(h)

Fig. 2 c

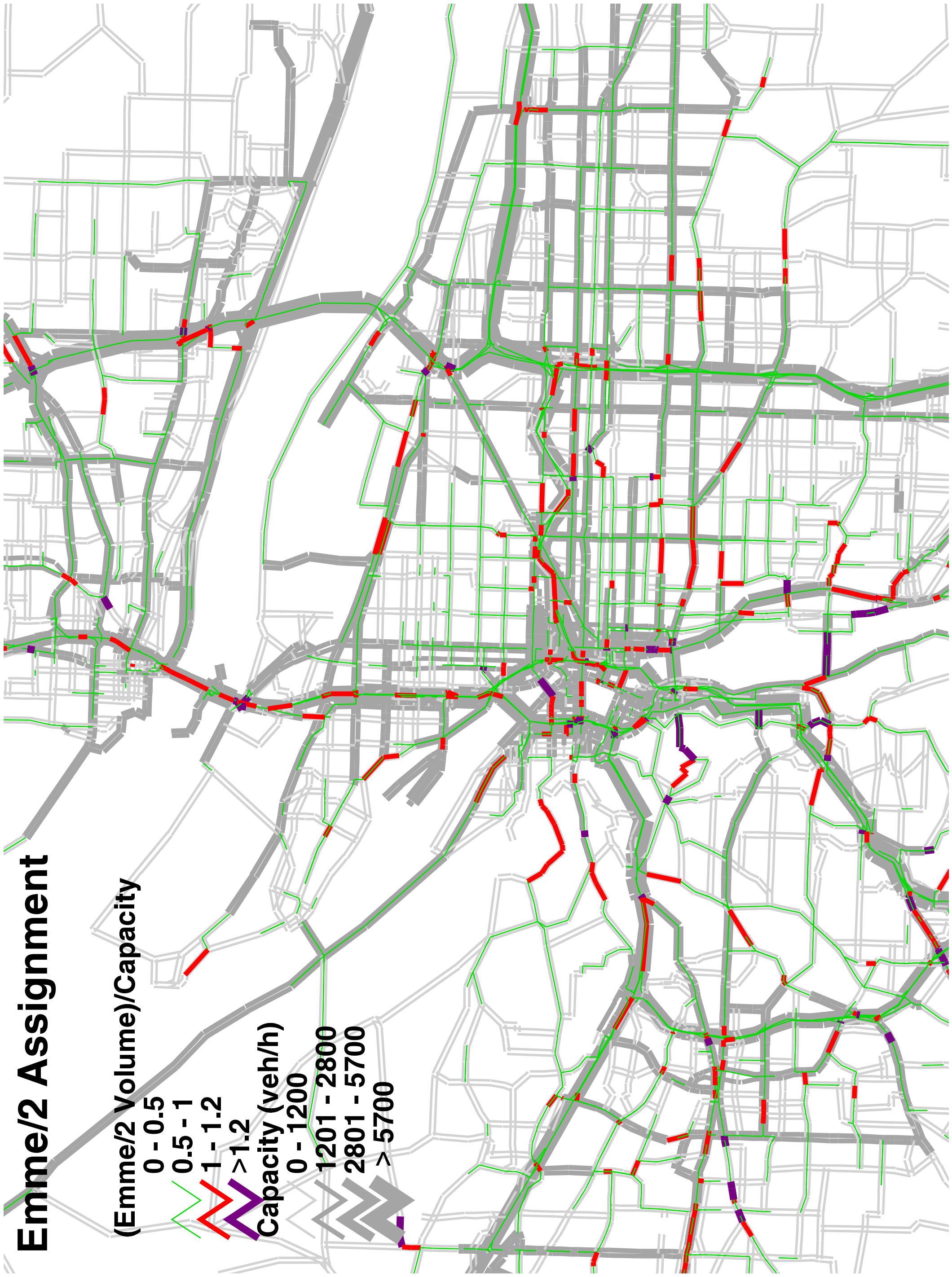

Emme/2 Assignment
(Emme/2 Volume)/Capacity
0 - 0.5
0.5 - 1
1 - 1.2
>1.2
Capacity (veh/h)
0 - 1200
1201 - 2800
2801 - 5700
> 5700

Fig. 3

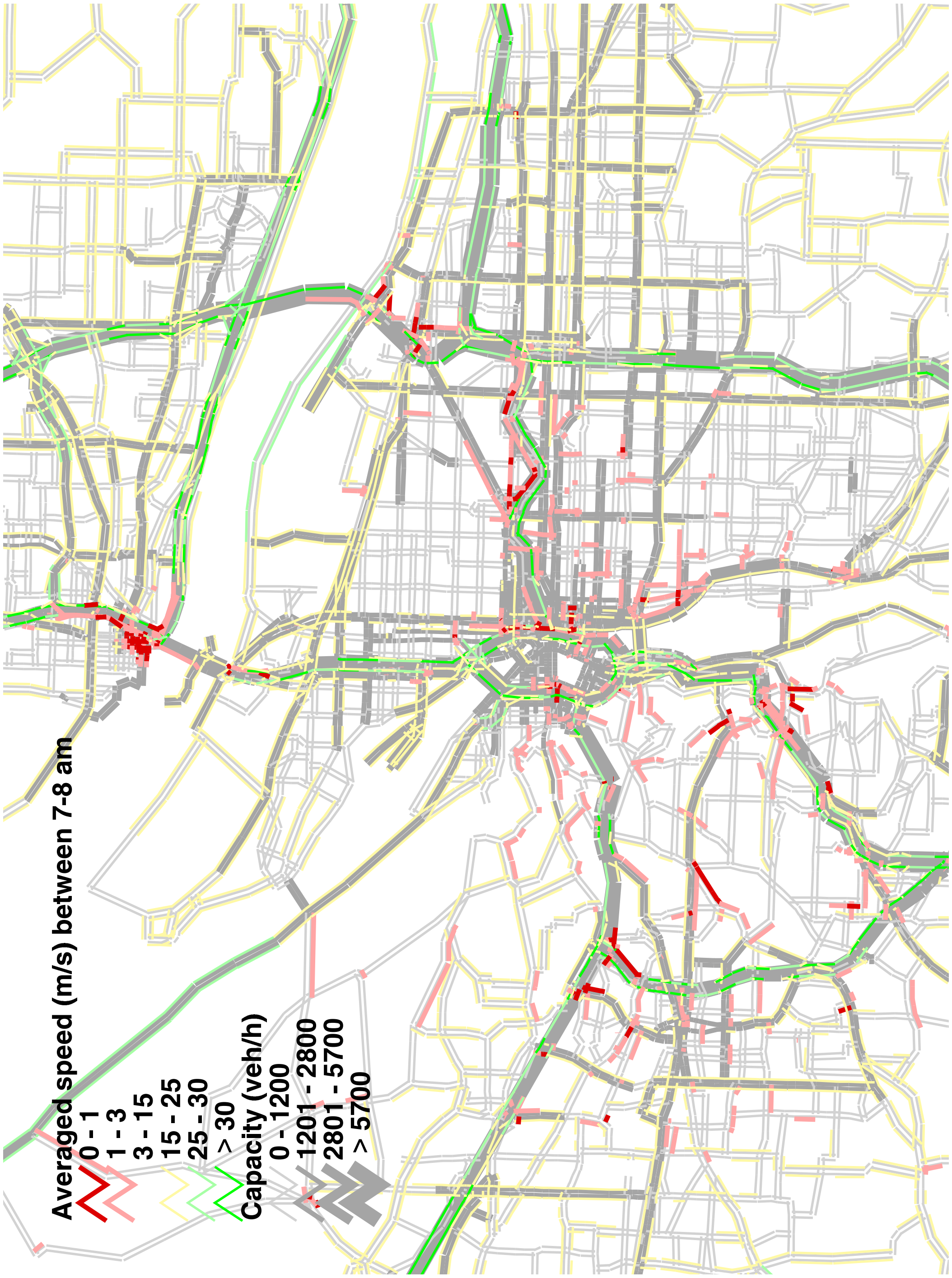
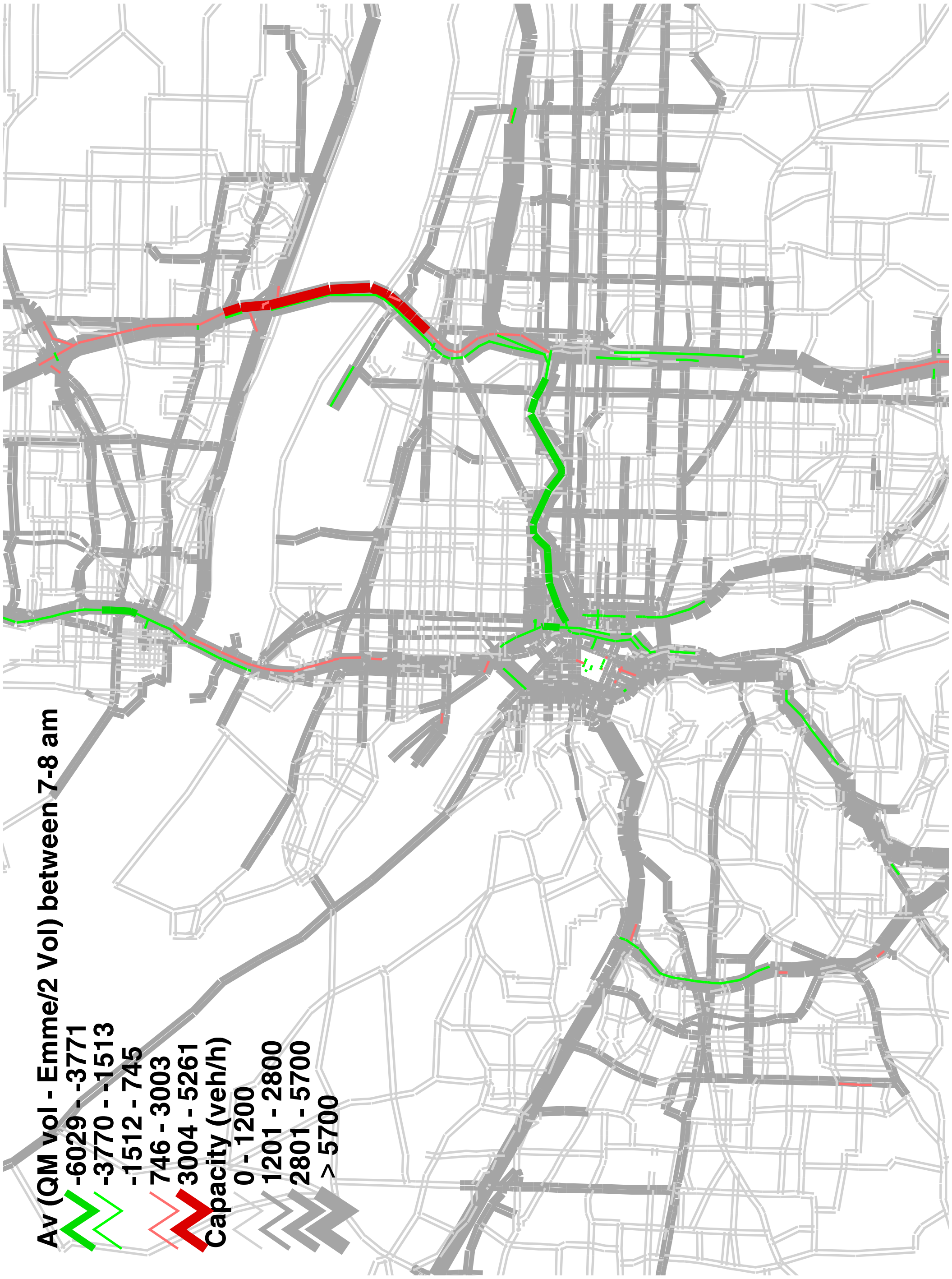

Averaged speed (m/s) between 7-8 am
0 - 1
1 - 3
3 - 15
15 - 25
25 - 30
> 30
Capacity (veh/h)
0 - 1200
1201 - 2800
2801 - 5700
> 5700

Fig. 4



Av (QM Vol - Emme/2 Vol) between 7-8 am
-6029 - -3771
-3770 - -1513
-1512 - 745
746 - 3003
3004 - 5261
Capacity (veh/h)
0 - 1200
1201 - 2800
2801 - 5700
> 5700

Fig. 5

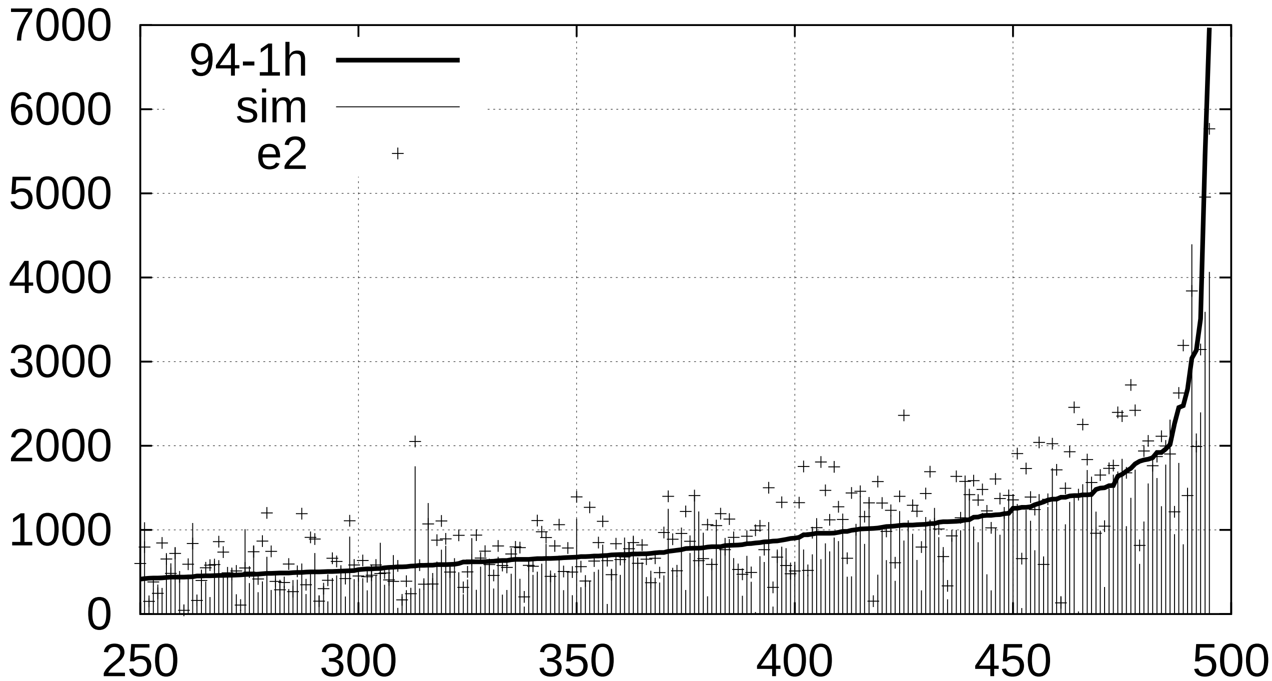

7000
6000
5000
4000
3000
2000
1000
0
250
300
350
400
450
500
94-1h
sim
e2

Fig. 6