

Traffic networks

Kai Nagel^a

^a Dept. of Computer Science, ETH Zürich
CH-8092 Zürich, Switzerland

January 14, 2002

Abstract

Transportation systems are complex dynamical systems whose dynamics unfolds on networks as the spatial substrate. Early approaches to the problem have similarities to the computation of equilibrium current flow in electrical networks, with the main difference that in traffic the particles have fixed destinations. These steady state approaches are unrealistic when describing more complex aspects of the dynamics, which is why time-dependent microscopic models are introduced. Such models resemble typical molecular dynamics simulations, except that the spatial substrate is a graph instead of flat space, and particles are “intelligent”. Both aspects are discussed in detail, the latter meaning that one has to go far beyond physics and into the area of human behavior and human learning. Another network aspect is the network of interaction between objects in the simulation, where these objects are not only travelers, but also traffic signals, traffic management centers, etc. For fast large scale simulations, one employs distributed computers, and mapping these interactions on the computational system is critical for high computing performance.

1 Introduction

Much of this book on networks is about the dynamics *of* networks. The question there is how networks form or change. Examples come from many different areas, from electrical networks to the blood system, or from the Internet to the networks of socio-economic interaction. This contribution concentrates on another aspect: on dynamics *on* networks. In the particular example of traffic, this means that there is an underlying network, the road network, and the dynamics of the system unfolds on this network. Although this is also true for other networked systems, such as for electrical networks or for biological networks (nerve system, blood transport system), the traffic dynamics on links is relatively complex and thus very interesting. The reason for this is that the particles, or agents, in the traffic simulation are “intelligent”, which means that they have strategic, long-term goals.

*nagel@inf.ethz.ch

The context for the work reviewed in this contribution is in transportation planning. This means the prediction of traffic patterns 20 or more years into the future. Let us, as an example, consider the question (relevant for Switzerland) to build a second Gotthard tunnel through the Alps. Initially, such a new tunnel would just relieve congestion (and increase safety). However, on a somewhat slower time scale of days to months, people who previously took a different route because of congestion will switch to the Gotthard route. On an again longer time scale, people will maybe make additional trips which uses this route which they may not have made before. And finally, it is possible that land use changes in reaction to such changes – in this case for example in terms of a leisure park or industry south of the Alps which needs easy access to or from the north. In other cases, people's housing decisions will depend on access to their workplace.

In consequence, there is an emerging consensus that transportation simulations for planning purposes should consist of the following modules (Fig. 1):

- **Traffic simulation** – This is where travelers move through the street network by walking, car, bus, train, etc.
- **Modal choice and route generation module** – The travelers in the traffic simulation usually know where they are headed; it is the task of this module to decide which mode they take (walk, bus, car, bicycle, ...) and which route.
- **Activity generation module** – The standard cause why travelers are headed toward a certain destination is that they want to perform a specific activity at that location, for example work, eat, shop, pick someone up, etc. The activity generation module generates synthetic daily plans for the travelers.
- **life style, housing, land use, freight, etc.** – The above list is not complete; it reflects only the most prominent modules. For example, the whole important issue of freight traffic is completely left out. Also, at the land use/housing level, there will probably be many modules specializing into different aspects.
- **Feedback** – The above modules interact, and the interaction goes in both directions: activities and routes generate congestion, yet (the expectation of) congestion influences activities and routes. This is typically solved via a relaxation method, i.e. modules are run sequentially assuming that the others remain fixed, until the results are consistent.

In addition, there need to be initialization modules, such as the **synthetic population generation module**, which takes census data and generates disaggregated populations of individual people and households. Similarly, it will probably be necessary to generate good default layouts for intersections etc. without always knowing the exact details.

In this review, we will look at this technology specifically from the view of networks. There are four aspects that we will discuss:

1. **Dynamics on networks** (Sec. 2) – As pointed out at the beginning of the introduction, the transportation system is a network of roads and other links connected at intersections, train stations, etc. These networks have interesting dynamics, both on the links and at the intersections.

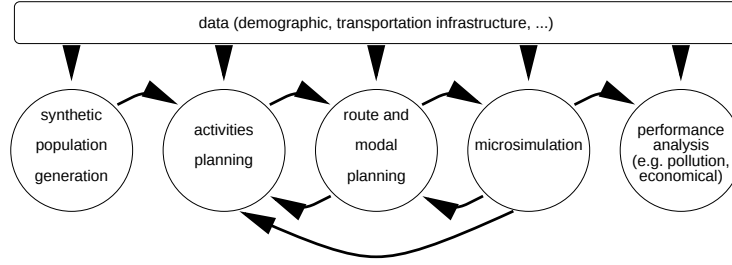


Figure 1: TRANSIMS modules

2. **Particles are intelligent** (Sec. 3) – Not strictly being a network aspect, it is however still important to note that travelers are “intelligent”, when compared to, say, molecules or blood cells. This aspect means that travelers have strategic goals, and they have internal representations of the world around them which they use to reach these goals. This means that two travelers in identical situations will in general make different decisions.
3. **The network of interactions and distributed computing** (Sec. 4) – Travelers and other objects in a transportation system interact. For example, congestion is formed by travelers being in each other’s way; ride sharing necessitates travelers to meet at a common pick-up location; adaptive traffic lights react to traffic conditions; etc. This network of interactions is generally sparse, and it is also often local although the mechanics of transmission are different from a physical force field – consider e.g. a sensor for an adaptive signal. At the opposite, non-local end, there are global radio broadcasts. There are also increasingly networked services which are both sparse and non-local – for example electronic route guidance systems where only very few travelers communicate with a center. We will review this aspect together with consequences for distributed large scale computation in Sec. 4.
4. **Dynamics of networks** (Sec. 5) – Finally, people can, via the political process, change the transportation network. Although it appears difficult if not impossible to make any reliable predictions here, it may be worthwhile to explore such models to understand the mechanisms behind it, in particular the effects of self-reinforcing decisions and positive feedback. We will look at this aspect in Sec. 5.

The paper will be concluded by a short summary.

2 Dynamics on networks

As pointed out above, traffic unfolds its dynamics on a graph, and the dynamics on the links (roads) and nodes (intersections) of this graph are complex and interesting. This section will concentrate on these dynamic aspects. The section will start by a

short review of the traditional static assignment method and then proceed to particle-based micro-simulations. As one will see, the static assignment method resembles a steady state current distribution in a fuse network while the micro-simulations resemble molecular dynamics simulation of particles flowing through a graph. Thus, transportation science follows physics on the path to more and more microscopic simulations in the attempt to go beyond steady-state phenomena.

2.1 The four step process and static assignment

The traditional method of traffic prediction for transportation planning is based on the four step process:

1. **Trip generation:** This module generates, for each traffic zone, the number of trips starting there and the number of trips ending there. This can be done for arbitrary time slices, but is often done for a typical 24-hour weekday.
2. **Trip distribution:** Trip generation results in sources and sinks, but not how they are connected. This is done in the trip distribution module. The result is an **origin-destination matrix**, which has, at row i and column j , the number of trips going from zone i to zone j .
3. **Modal choice:** In this module, the trips are split between the modes of transportation.
4. **Route assignment:** For each trip, a path is found through the network so that no other path is faster. Congestion is taken into account via the link travel time being a function of the trips using that link.

Route assignment can be formalized in the following way: Let r_{ij} be the number of trips from i to j . Routes from i to j are numbered by k ; in consequence, $r_{ij,k} \geq 0$ is the number of trips using the k -th route. Let $\delta_{ij,k,a}$ an indicator if route ij, k uses link a . The number of trips using link a then is

$$x_a = \sum_{ij} \sum_k r_{ij,k} \delta_{ij,k,a} .$$

The link travel time (link cost) is normally defined via a function $t_a(x_a)$. It makes sense to assume that this function is strictly monotonically increasing. The trip time of a route in consequence is

$$T_{ij,k} = \sum_a t_a(x_a) \delta_{ij,k,a} .$$

The problem specification now is that $r_{ij,k}$ need to be found such that

$$\sum_k r_{ij,k} = r_{ij}$$

and such that all used routes have the same travel time and no unused route has a faster (= better) travel time.

This is typically solved by making the $r_{ij,k}$ continuous, meaning that also trip generation, trip distribution, and modal choice can be made on real numbers. With real numbers and with the assumption that $t_a(x_a)$ is strictly monotonically increasing, it can be proven that the above problem has a unique solution in terms of the x_a . The problem can in fact be written as a minimization problem, making it amenable to the tools of nonlinear optimization. In consequence, sophisticated algorithms exist which compute numerical approximations to the unique solution [1, 2].

The above problem is very similar to a non-linear static network flow problem in physics, where the link resistance is given via a non-linear $U = R(I) I$, and where sources and sinks are given via the result of the trip generation. The only (but important) difference is that in assignment “particles know where they go”, meaning that one cannot in general exchange particles as one can, in electrical networks, do with electrons.

Static assignment has many shortcomings. For example, it does not correctly represent dynamic effects such as queue build-up, and it does not have enough microscopic information to do, for example, emission calculations. It also de-couples decisions from individual actors. For example, the only decision available for modal choice is the origin and the destination of the trips; important aspects such as income, car ownership (!), additional trips during the day, etc. are not used. Note, however, that these latter aspects could be overcome by a different software design. What cannot be overcome are the shortcomings in the representation of dynamic effects, which are treated in more detail in the next section.

2.2 Simple link dynamics and the queue model

In static assignment, one assumes a function $t_a(x_a)$ for each link. In practice, this function is parametrized by a few numbers, such as the free speed and the capacity of the link. The capacity is the maximum number of vehicles that can traverse a given location on the link per time unit; i.e., it is the maximum throughput or maximum current for that link. From a physics perspective, it is clear that such number must be an average, and that any realization of traffic can deviate from that number, especially for short times. Nevertheless, it should be clear that, if a link has a capacity of C_a , then, in the average no more than C_a vehicles can traverse the link.

Now imagine a scenario as in Fig 2, with a road 1 with a capacity of 4000 vehicles per hour connected to a road 2 with a capacity of 2000 vehicles per hour, and a demand zero for $t < 0$ and of 3000 per hour for $t \geq 0$. After one hour, 1000 cars will be queued up at the entry to the bottleneck, and the queue will grow by another 1000 cars in each hour. That is, the steady state solution of a demand in excess of capacity corresponds to an infinitely long queue *upstream* of the overloaded link. Static assignment would have to represent this via its link travel time functions $t_1(x_1)$ and $t_2(x_2)$. It would show link 2 as overloaded and congested, which is dynamically incorrect, since it is link 1 (and eventually additional upstream links) which bear the consequences, as we just saw.

It would be possible to avoid this situation in static assignment by setting $t_a(x_a) = \infty$ as soon as x_a exceeds capacity C_a , since the route assignment would then avoid to put more than C_a trips on that link. This, however, also does not correspond to reality,

since waiting queues at the entry to bottlenecks clearly exist. Alternatively, one could attempt to formulate a mathematical model which includes queues. Although this is in principle feasible, not enough mathematical facts are known about such a model to make it useful in practice (see, e.g., [3]). In addition, with every level of additional complexity the situation looks more hopeless. For example, something like individual preferences, or different link travel speeds (e.g. cars vs. trucks), or the effects of turn pockets/merge lanes, or emissions resulting from acceleration, are difficult to represent in the static assignment framework.

Putting these arguments together, it makes sense to consider microscopic approaches. In microscopic approaches, all individual objects such as vehicles or travelers will be represented individually. Here, we will start with a simple microscopic model which is called the queue model. In the queue model, links are represented by simple FIFO (first-in first-out) queues. Vehicles entering a link at time t_1 are assumed to travel along the link with free speed V_a , meaning that they cannot leave the link before time $t_2 = t_1 + L_a/V_a$, where L_a is the link length. At the end of the link there is a capacity constraint, meaning that at most C_a vehicles can leave the link per time step. Non-integer C_a are resolved by using probabilities. So far, this is indeed a standard queuing model [4]. An important addition is the introduction of a storage constraint, meaning that there is a limitation of the number of vehicles that one can put on a link. It is the link storage constraint which will eventually make the link “full” and thus cause queue build-up and congestion spill-back.

In spite of their aesthetic appeal and their computational speed, there are a number of problems with queue models. Some are relatively simple geometrical shortcomings, such as the fact that, although for city networks it makes sense to have the capacity constraint at the end, in other situations like for freeways or in Fig. 2, they are at the beginning. Others are more severe, such as the fact that intersection design has not been solved satisfactorily with this model. The difficulty stems from the fact that both the capacity and the storage constraint need to be satisfied. Especially at high congestion levels, there are typically at each intersection many vehicles from incoming links competing for the same few slots on outgoing links. In reality, this is solved via explicit prioritization, either based on traffic lights or stop/yield signs, or on explicit legal rules such as “right before left”. For transportation planning however, this information is often not available, and it is also subsumed in the capacities. Although this problem seems solvable, some more systematic work will be necessary here. Finally, it is difficult to consistently handle different vehicle speeds, vehicle types, or vehicle classes. An example for vehicle classes, which in this case differ by destination link at an intersection, is depicted in Fig. 4.

2.3 Virtual reality micro-simulations

An alternative to the queue model, avoiding the problems which come from the reduced geometrical representation, are micro-simulations which run on correct street layouts, including merging, turning, and weaving lanes, correct signal schedules etc. In terms of driving rules, such simulations consist of four major elements:

1. Car following. This describes how one vehicle follows another (or many others)

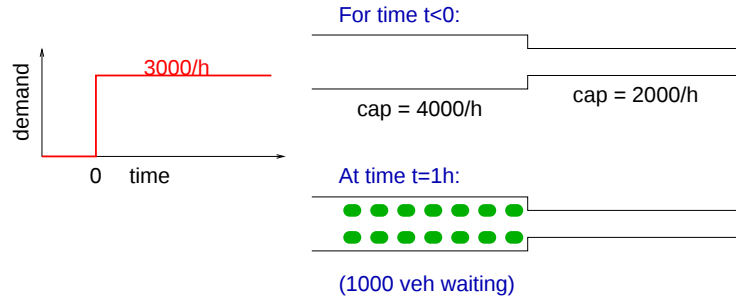


Figure 2: Example of failure of static assignment in the representation of dynamics. The scenario is a street with a capacity of 4000 vehs/hour leading into a street with a capacity of 2000 vehs/hour. The demand is zero before $t = 0$ and 3000 vehs/hour afterward. In each hour, an additional 1000 vehs will be waiting in upstream of the bottleneck.

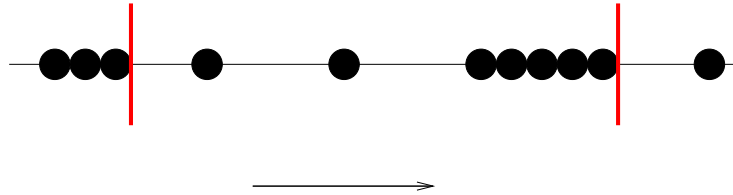


Figure 3: Queue model dynamics

on a single lane road.

2. Lane changing. This describes how vehicles change lanes on multi-lane roads. Reasons to change lane can be speed improvement, or anticipated turns in the future (a vehicle that wants to make a left turn needs to get into the left lane). Passing against oncoming traffic also belongs here.
3. Protected turns. This describes how vehicles behave at fully signalized intersections. For transportation planning purposes, it is enough to simply make vehicles stop at red and go at green.
4. Unprotected turns. Often, movements across intersections are not protected by signals, such as a left turn against oncoming traffic, or at a yield sign. Also, there may be special rules such as that the light rail always has priority.

Such microscopic simulations have the advantage that, at least in principle, they can be made arbitrarily realistic by adding more and more rules. In addition, they look very convincing to a non-technical audience (Fig. 5), an important aspect since the results of transportation planning simulations are of interest to all citizens.

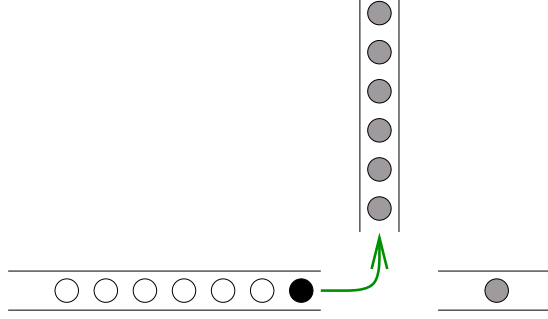


Figure 4: Limited geometric representation with queue model: The white vehicles cannot move (to the right) because the black vehicle, which is stuck, is in the way.

2.4 CA implementations of virtual reality micro-simulations

The maybe simplest approach to implement these rules are cellular automata micro-simulations. In these micro-simulations, roads are segmented into cells, each typically of the size that a single vehicle occupies in a jam (e.g. 7.5 meters). Movement is performed via jumps from one cell to another; for example, a speed of 5 cells per time-step corresponds to a jump of 5 cells. In order to translate those units into the real world, a time step of one second is customary (because of reaction time arguments), meaning that “5 cells per time-step” corresponds to

$$5 \text{ cells/time-step} \times 7.5 \text{ meters/cell} \times 1 \text{ second/time-step}$$

$= 37.5 \text{ meters/second} = 135 \text{ km/h}$. In such a cell-based system, a simple rule set to achieve the above functionality is as follows.

1. Car following.

For all cars do in parallel

- $v' = \min[v(t) + 1, \text{gap}(t), v_{max}]$, where v' is a temporary variable, $v(t)$ is the current velocity, $\text{gap}(t)$ is the number of empty cells ahead, and v_{max} is the maximum speed, for example given by the speed limit.
 gap models the effect that one has to slow down if there is a vehicle ahead; note that this simple formulation assumes infinite braking capability.
- With probability p_{noise} , $v(t + 1) = \max[v' - 1, 0]$, else $v(t + 1) = v'$.

After this is done for all vehicles, each vehicle is moved forward according to its speed.

This model [5] has been investigated in much detail in the literature; see Ref. [6] for a review. Its main feature is that at high enough densities, distinctive traffic jams form which would be interpreted as start-stop traffic by an individual driver.

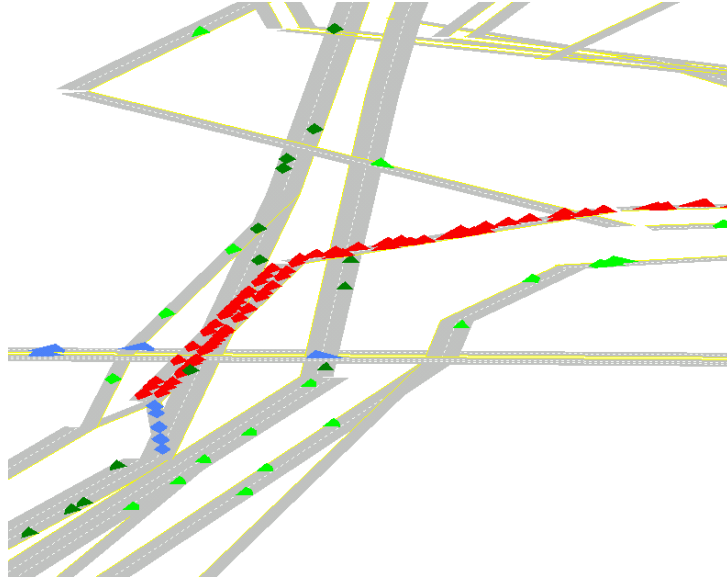


Figure 5: Virtual reality representation of simulated traffic in Portland/Oregon.

2. Lane changing. Before the speed calculation, do the following:

For all cars do in parallel

- Set “reason-to-change-lanes” to true if the vehicle eventually wants to make a corresponding turn. Also set it to true if the other lane is faster than the current lane.
- Set the safety criterion to true if there is sufficient space on the other lane.

After this is done for each vehicle, all vehicles for which both criteria are fulfilled change lanes.

There are more technical details here than with car following. For example, the above criteria need to be filled with quantitative rules, and care needs to be taken that two vehicles do not end up in the same cell during the parallel update. Also, the lane changing in anticipation of turning movements requires care, because vehicles can change lanes too early, meaning they may get stuck in a queue for a different turn, or too late, meaning they may not be able to make the intended turn. For more information see, e.g., Ref. [7, 8].

3. Protected turns. As stated above, this is relatively simple. As long as the modeled vehicles have infinite braking capability, a red light can be modeled by a virtual car of speed zero being inserted and removed at the location of the traffic signal.
4. Unprotected turns. A simple rule is (see Fig. 6):

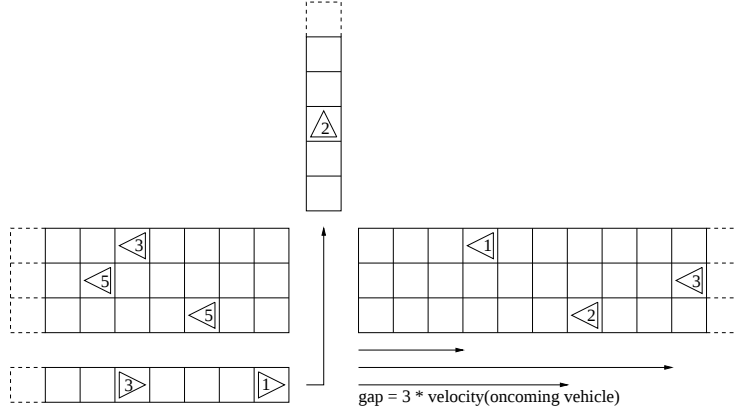


Figure 6: Illustration of gap acceptance for a left turn against oncoming traffic. From [8]

For each interfering lane (meaning a lane which has priority), check if the gap in front of the interfering vehicle is large enough. If yes, accept the turn, otherwise wait.

Both with protected and with unprotected turns, there needs to be space available on the outgoing link.

The above is simplified in many respects, in particular with respect to complicated intersection designs, which are here replaced by the assumption that a vehicle makes the complete decision at the waiting position, and once the decision for a movement has been made, it can move freely across the intersection. Another simplification concerns the use of the cellular automata (CA) technique. CA are easy to code for such applications, since most driving rules need spatially-organized access to data, for example to neighboring cells, and the CA technique provides that. It is however also possible to code vehicles as individual particles with continuous position and speed (e.g. [9, 10, 11]), similar to a molecular dynamics technique [12]. Such codes are harder to program efficiently since one needs to keep track of spatial neighbors, but when done correctly they are computationally as fast as CA codes. This is helped by the fact that for traffic, taking the limit of $\Delta t \rightarrow 0$ where Δt is the computational time step is not useful since human reaction time needs to be modeled.

The trick with using such microscopic models for transportation planning is to make them computationally fast by making them on the micro-level barely realistic enough to obtain good information on the macro-level. This is consistent with a Statistical Physics approach, where many macroscopic laws can be obtained from much simplified microscopic models. Much progress on this aspect has emerged in recent times. For example, from Monte Carlo runs it has become clear that models as simple as the ones described above are macroscopically reasonable. They generate plausible fundamental diagrams, both on closed links and at intersections (Fig. 7, also see [13]), and they can display the emergence of the infamous jam-out-of-nowhere (Fig. 8) al-

though there is discussion about its real world importance [14].

In terms of theory, some simple CA models can be provably related to accepted macroscopic theories of traffic: The continuous limit of the Asymmetric Stochastic Exclusion Process (ASEP) is the Burgers equation

$$\partial_t \rho + \partial_x \rho - 2\rho \partial_x \rho = D \partial_x^2 \rho ,$$

which implies a relation of $q = \rho (1 - \rho)$ between density ρ and flow q (see, e.g., [15]). This, in return, means a speed-density relation of $v = 1 - \rho$, which is known as the Greenshields relation (see [16]) in traffic flow theory. For deterministic continuous microscopic models, it has been shown, in certain cases mathematically and in others by computer simulation, that the mechanism of traffic flow breakdown (i.e. the transition from homogeneous “laminar” traffic to inhomogeneous traffic with stop-and-go waves) is the same as for Navier-Stokes models for traffic flow [17, 11]. Kinetic theory can build an, albeit still fragile, mathematical bridge from microscopic dynamics to fluid-dynamic equations [18, 19]. For an exhaustive review of models for traffic, see [6, 20]. Interestingly, the precise mechanism for traffic breakdown in stochastic models is still under discussion [21, 22]. In particular, under certain circumstances the boundary of jams is weakly fractal [23] (look at Fig. 8 for an impression) while under others it is not [10], and this is related to a discussion about a possible phase transition and its order.

A disadvantage of a “virtual reality micro-simulation” is that it needs rather a lot of input data. For example, many aspects of the street network are needed, such as merge or turn pockets, signal plans, grades, speed limits, or lane connectivities. Fig. 9 shows an example of the last: the arrows denote which incoming lanes are connected to which outgoing lanes, and the micro-simulation needs this information to induce lane changing as described above.

Since these data are usually not readily available, good “default generators” should be developed. They would for example generate plausible intersection designs and signal schedules based on other data, such as link capacities (maximum hourly flows) which are often available. Such synthetic defaults could then be used (with care) until real data became available. Also, the simulation could be used to detect obvious implausibilities, which could then be corrected on a case-by-case basis.

Since macroscopic quantities such as hourly flow are “emergent”, there is no method to systematically construct the needed microscopic from the available macroscopic data. In consequence, the only method available is to run systematic tests with many intersection layouts and to record the resulting behavior. From such simulations, lookup tables could be constructed which then generate the microscopic designs from the macroscopic data.

Finally, good care needs to be taken to clearly differentiate between synthetic and field data in the data bases. This is often not done, or it is not done automatically by the system, resulting in questionable data entries necessitating costly manual correction.

2.5 Traffic in networks

In the above, we have started from static assignment and pointed out its similarity to equilibrium flow in physical networks. In both systems, when the dynamical aspects

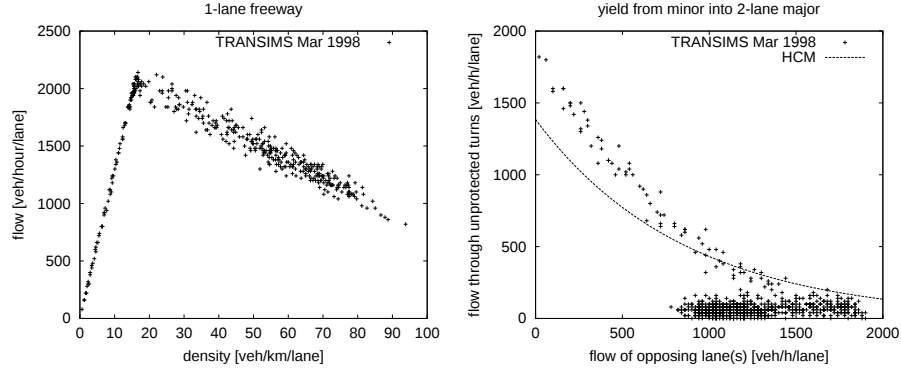


Figure 7: Fundamental diagram (flow vs. density; LEFT) and diagram for unprotected yield (unprotected flow vs. major flow; RIGHT).

become important, the steady-state equilibrium approach is no longer valid. Similar to physical networks, progress can be made by a microscopic approach, which means to represent particles (travelers) individually instead of as part of some steady state rate or steady state flow. We have then presented two possible micro-simulations, one relatively simple, based on a small but important extension of standard queuing models, and one relatively advanced, with the ability to be extended toward a virtual reality micro-simulation.

When inspecting the two presented systems, one will notice that for the first one (the queue simulation) nearly all the dynamics happens at the intersection, while for the second one there is an equal share of link and intersection dynamics. Many practitioners believe that for traffic in urban networks, the intersection dynamics is much more important than the link dynamics. This motivates a much simplifying approach to traffic in networks, which is to model traffic on a two-dimensional grid [24, 25, 26, 27]. Such simulations show interesting phenomena, such as phase transitions to grid-lock. It is an open question in how far these observations can be translated to more realistic traffic networks, with their more complicated (and more “forgiving”) intersection dynamics.

3 Particles are intelligent

We have argued in the introduction that transportation planning tools need to include effects ranging from traffic flow via human decision-making up to land-use planning. We have then presented the static assignment approach to transportation planning, where a restricted representation of the traffic dynamics made it impossible or useless to make the other modules more realistic. Following that, we have described two alternatives for the traffic dynamics which “repair” these problems. Both traffic simulations that we describe are dynamic (i.e. time-dependent) and microscopic (i.e. each traveler is individually represented), which are the minimal pre-requisites for the following ar-

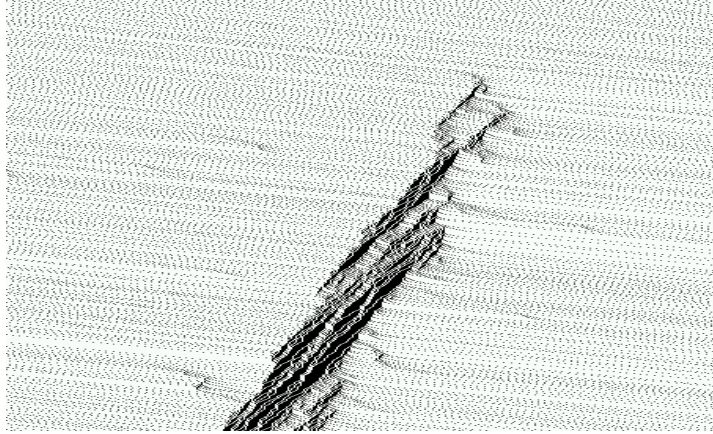


Figure 8: Traffic jam out of nowhere. The plot shows a space-time plot, space being horizontal, and time increasing from top to bottom. In consequence, the lines of the plot show consecutive time steps. The jam emerges spontaneously, and it shows a fragmented, weakly fractal structure. Traffic moves from left to right; the jam moves against the traffic direction.

guments. We have presented the two approaches to make clear that there is a wide range of models which fulfill these criteria, ranging from relatively simple (such as the queue simulation) to extremely realistic; the only restriction is that computation needs to be fast enough. Clearly, there are other models which fulfill these specifications [28, 29, 30, 31, 32, 33, 34].

Given that, it is now possible to improve the other aspects of the four step process. As in the traffic flow simulation, instead of gradual improvement we will focus on a bottom-up approach from first (or microscopic) principles. Again, the microscopic actors of the system are the travelers. As pointed out, the main difference to a simulation of, say, electrons in an electric network is the internal intelligence and adaptability of travelers. In contrast to water molecules, two travelers in exactly the same external situation can make different decisions.

3.1 Route generation

Having travelers move around randomly is not enough. For example, if a car approaches an intersection, the driver needs to decide the turning direction. A traditional method is to use turn counts, meaning that there is empirical data with the information about what fraction of the traffic goes into which direction. For any kind of transportation planning question, this is not enough information. The most drastic example is the addition of a new road: There would be no information available of how the traffic at the connecting intersection redistributes when the new road becomes connected. One would also assume that turn counts at other intersections change, since some of the

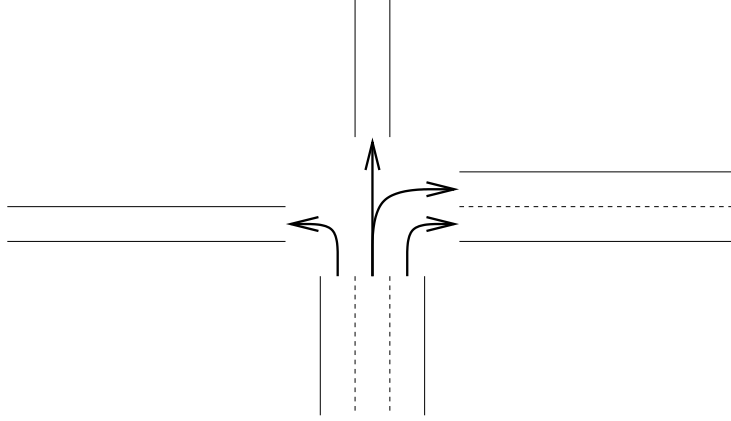


Figure 9: Lane connectivities

traffic would adapt to use the new road.

This means that for transportation planning simulations it is indispensable to know the destinations, and to have routes for each vehicle. In this way, when a new road or a railway connection is added, every traveler can consider to adapt their routing in order to use this new connection. The route generation module of the transportation planning simulation should be multi-modal (i.e. include other modes besides cars), although some of the mode decision is better done in the demand generation module (see next).

A typical method for route generation is a time-dependent fastest path algorithm. Given a starting time t_0 , an origin i and a destination j , and, for each link, information how long it will take to traverse the link when entering at a specific time, this algorithm will compute the fastest path from i to j when starting at time t_0 . The time-dependent Dijkstra algorithm which solves this problem is, with a heap implementation, of complexity $M \log N$, where M is the number of links and N is the number of nodes (intersections). This is in fact a very low complexity, and it is difficult to construct a heuristic which is significantly faster [35].

3.2 Activity generation

For many questions, having the routes adaptive while the activities remain fixed is not enough. For example, making travel faster usually results in people making more trips. This is called **induced traffic**. Conversely, increasing congestion levels will eventually suppress trips which would otherwise be made, although it is not always clear which trips are suppressed and what congestion level is necessary to have that effect.

In order to deal with these and other effects, one has to make demand generation adaptive to congestion. A recent method for this is activity generation, meaning that, for each individual in the simulation, one generates a list of activities (such as sleeping, eating, working, shopping) plus locations and times (Fig. 10). Since in this method each traveler is treated individually, it is possible to use arbitrary decision rules, which

HUSBAND'S ACTIVITIES

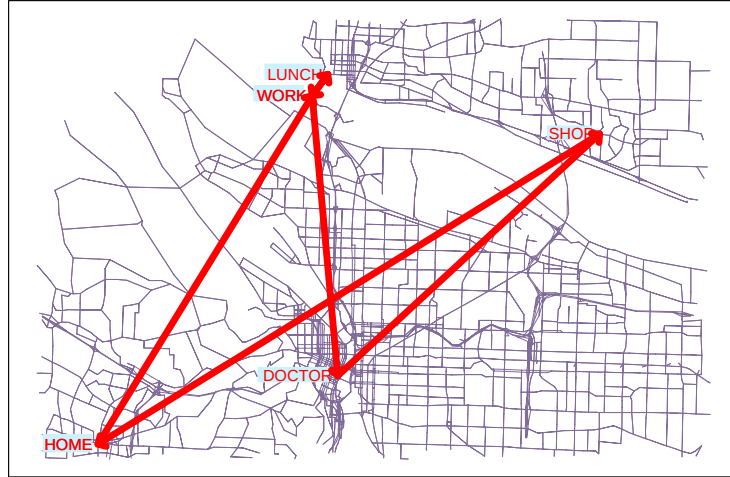


Figure 10: Example of a sequence of activities for a person in Portland/Oregon. From R.J. Beckman.

means that arbitrary methods can be investigated. The currently best-accepted methods are based on random utility theory and are called discrete choice models [36].

As stated above, activity generation needs to be done in conjunction with mode decisions. For example, having a car clearly changes the list of preferable destinations for a given activity, or may even make other activities more desirable.

3.3 Housing, land use, freight, life style, et al

Transportation planning does not stop at activities. For example, making commuting roads faster by increasing capacity usually results in more people moving to the suburbs. That is, housing decisions are closely related to transportation system performance. Similarly, questions of land use (e.g. residential vs. commercial vs. industrial areas) clearly influence and interact with transportation. Freight traffic needs to be considered. Life style choices (e.g. urban life style, often without car ownership, vs. rural life style, usually with car ownership) need to be considered; as already alluded to above, such long-term commitments have strong influence on activity selection and modal/route choice.

3.4 Day-to-day learning, feedback, and relaxation

There is strong interaction between the above modules. For example, plans depend on congestion, but congestion depends on plans. A widely accepted method to resolve this is systematic relaxation (e.g. [37]) – that is, make preliminary plans, run the traffic micro-simulation, adjust the plans, run the traffic micro-simulation again, etc., until

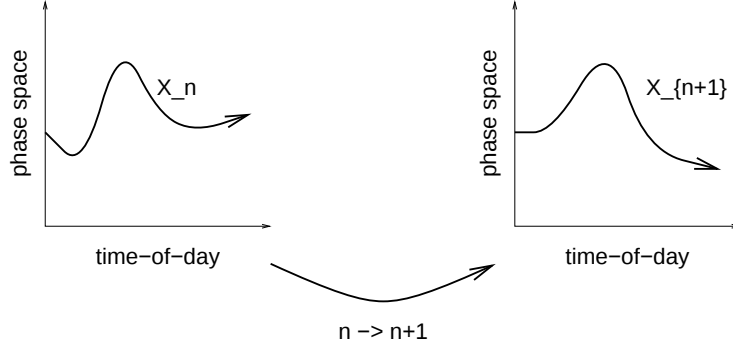


Figure 11: Schematic representation of the mapping generated by the feedback iterations. Traffic evolution as a function of time-of-day can be represented as a trajectory in a high dimensional phase space. Iterations can be seen as mappings of this trajectory into a new one.

consistency between modules is reached. The method is somewhat similar to a standard relaxation technique in numerical analysis.

Such iterated simulations can be treated as discrete dynamical systems (Sec. 5.3). A state is the trajectory of the simulation through one day; an iteration is the update from one day (period) to the next (Fig. 11). As such, one can search for things like fix points, steady state densities, multiple basins of attraction, strange attractors, etc. Typically, one would first analyze the steady state behavior, and then the transients. Under certain conditions the existence of a unique steady state can be proven [38], although for the computationally feasible number of iterations the possible occurrence of “broken ergodicity” [39] needs to be taken into account. Broken ergodicity is the property of a system to be mathematically ergodic but to remain in parts of the phase space for long periods of time.

3.5 Within-day re-planning

All the above lines of thought still assume, in some sense, “dumb” particles. Particles follow routes, but the routes are pre-computed, and once the simulation is started, they cannot be changed, for example being adapted to unexpected congestion and/or a traffic accident. In other words, the intelligence of the agents is external to the micro-simulation. In that sense, such micro-simulations can still be seen as, albeit much more sophisticated, version of the link cost function $c_a(x_a)$ from static assignment, now extended by influences from other links and made dynamic throughout time. And indeed, many dynamic traffic assignment (DTA) systems work exactly in that way (e.g. [37]), in spite of several problems in particular with quick congestion build-up [40].

Another way to look at this is to say that one assumes that the emergent properties of the interaction have a “slowly varying dynamics”, meaning that one can, for example, consider congestion as relatively fixed from one day to the next. This is maybe realistic under some conditions, such as commuter traffic, but clearly not for many

other conditions, such as accidents, adaptive traffic management, impulsive behavior, stochastic dynamics in general, etc. It is therefore necessary that agents are adaptive (intelligent) also on short time scales not only with respect to lane changing, but also with respect to routes and activities. It is clear that this can be done in principle, and the importance of it for fast relaxation [41, 42] and for the realistic modeling of certain aspects of human behavior [43, 44] has been pointed out. Nevertheless, we are not aware of operational implementations of this aspect.

3.6 Individualization of knowledge

Another aspect of intelligent agents is that their “knowledge” should be private, i.e. each agent should have a different set of knowledge items. For example, people typically only know a relatively small subset of the street network, and they have different knowledge and perception of congestion. This is called “mental maps”; some experimental implementations are Refs. [45, 46, 47, 48]. We will come back to computational aspects in Sec. 4.2.

3.7 State of the art

No simulation package currently integrates all the aspects that are discussed. TRANSIMS [49] comes from the transportation planning side and is maybe the most advanced in terms of using advanced computing methods for large scale scenarios. The TRANSIMS research program is reaching completion in 2002, with a full-scale simulation of a scenario in Portland/Oregon, with a network of 200 000 links and several million travelers. We ourselves are in the process of using TRANSIMS for a full-scale simulation of all of Switzerland [50], see Fig. 12. DYNAMIT [29] and DYNASMART [30], originally started as transportation simulation tools for the evaluation of ITS (Intelligent Transportation System) Technology, also advance into the area of transportation planning by the addition of the demand generation modules. Some comparison between field data and simulation results, obtained from a queue model micro-simulation and much simplified demand generation, can be found in [51]. METROPOLIS [28] is a package designed to replace static assignment by a simulation-based but very simple dynamic approach. It allows the user to specify arbitrary link-cost functions but in its current version still does not allow the queue build-up as discussed in Sec. 2.2. Its strength lies in the self-consistent computation of departure time choice. A more complete overview of regional transportation simulation models can be found in [52].

4 Distributed computing and the network of interactions

So far, this text has assumed that agents do their “strategic planning” independently of each other, and interactions occur in the traffic micro-simulation alone. These interactions are entirely local, since cars/drivers react to the situation around them, including other cars, signals, speed limits, etc. In consequence, the network of interactions is

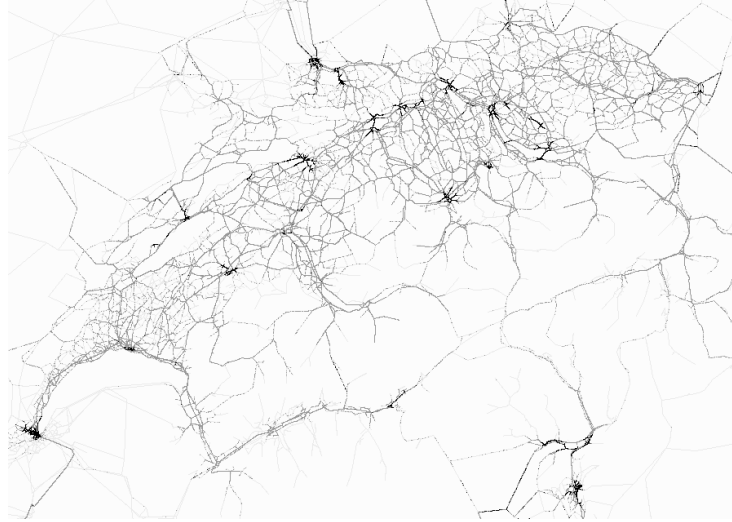


Figure 12: Microscopic simulation of all of Switzerland. Preliminary result

similar to, say, a molecular dynamics simulation with short-range interactions, where the network of interactions reflects the spatial dimensionality of the scenario.

We have also seen that the results of interaction, such as congestion, for many reasons cannot assumed to be fixed from one run to the next, and that an on-line (or within-day) replanning capability should be included. From here, it is easy to see that long-range interactions clearly play a role, for example the telephone call to another household member to pick up the child from the kindergarten since one is running late. Other examples for long-range interaction are:

- Interaction of agents with the transportation infrastructure, for example at adaptive traffic lights
- Reaction to radio broadcasts from a traffic management center
- Reaction to reports from friends
- Car sharing
- Coordination (ride sharing, household activities)

The graphs of these interactions can be seen as (meta-)networks. These networks are *sparse* and *dynamic*, meaning that in spite of the long-rangedness only a small number of particles interacts (in contrast to long-range forces in some molecular dynamics simulations), and that the network links re-organize over time.

The network aspects of interaction become particularly clear when one looks at computational aspects. As long as one runs everything on a single CPU, it is in principle possible to write one monolithic software package. In such a software, an agent

who wanted to change plans would call a subroutine to compute a new plan, and during this time the computation of the traffic dynamics would be suspended. However, single CPUs are too slow for large scale simulations, and so one uses parallel computers, typically large clusters of connected PCs. In such a cluster, the regional area of the simulation is distributed across the PCs so that each PC only deals with a small part (domain decomposition). Other PCs deal with, say, the computation of routes or of activities. This approach means that interactions between simulation objects often result in interactions between PCs, which need to be explicitly coded, usually via message passing.

4.1 Distributed computing of the traffic micro-simulation

The most compute-intensive part of current implementations is usually the traffic micro-simulation. A simple calculation gives an approximate number: Assume a 24-hour simulation ($\sim 10^5$ sec) and a one second time step, 10^7 travelers, and a 1 GHz CPU (10^9 CPU-cycles per sec). Further assume that the computation of one time step for each traveler needs 100 CPU-cycles – remember the driving rules (car following, lane changing, protected turns, unprotected turns) and include overhead for route following etc. The result is that such a simulation takes about

$$\frac{10^5 \times 10^7 \times 10^2}{10^9} = 10^5$$

seconds or approximately 1 day on a single CPU. This is indeed approximately correct for a TRANSIMS simulation of a corresponding Switzerland scenario (5 mio travelers; network with 28 622 links); the queue simulation is 10–100 times faster [53].

The simulations can be accelerated by using parallel computers. This becomes indispensable for large applications when including feedback learning as discussed in Sec. 3.4 since this multiplies the computing times by a factor of 50, resulting in 50 days of computing time for the above scenario when using the TRANSIMS micro-simulation. We focus on so-called Beowulf architectures, since they are the most probable ones to be available to prospective users (metropolitan planning organizations; traffic engineering consulting companies; academics). Beowulf clusters consist of regular workstations (such as Pentium PCs running Linux) coupled by regular local area network (such as 100-Mbit Ethernet).

The idea is to divide the simulation area into many pieces, each of which is given to a different CPU. The CPUs communicate e.g. via message passing. In principle, using, say, 100 CPUs should result in a speed-up of 100. In practice, there are many limiting factors coming from the hardware and from the operating system. For traffic micro-simulations, the most important limiting factor is the latency of the Ethernet, which (in an off-the-shelf system without tuning) is of the order of 1 msec [54]. Since each CPU in the average needs to communicate with six other CPUs, this means that each time step needs approx. 6 msec for communication. This limits the speed-up to $1 \text{ sec} / 6 \text{ msec} \approx 167$, independent of the number of CPUs that one uses. In practice, “100 times faster than real time” is a good rule of thumb [53, 55]. This domain decomposition approach is similar to a parallel computing approach to “standard” particle dynamics, for example in molecular dynamics [12], with the maybe only distinction

that molecular dynamics simulations rarely use a graph instead of regular Cartesian space as spatial substrate.

Unfortunately, in contrast to many other computing aspects, latency does not seem to improve in commodity hardware: it has been virtually unchanged from 10 Mbit Ethernet to 100 Mbit Ethernet to Gbit Ethernet; FDDI is even slower. This has some interesting consequences:

- The above result refers to the speed-up with given system size when using more and more CPUs. Alternatively, one can run larger and larger systems when using more and more CPUs. As is well known, scale-up is much less problematic on parallel computers than speed-up. In consequence, it is possible to run scenarios of virtually arbitrary size 100 times faster than real time.
- Alternatively, one can make the micro-simulations more realistic while still being able to compute 100 times faster than real time.
- It should be noted that parallel supercomputers do not have the same limitation since they employ special purpose hardware for the communication between CPUs. This results in an improvement by a factor of 100 for latency, meaning that for practical scenarios other factors play a more important role.

While a parallel Beowulf costs of the order of 2000-3000 U.S.-\$ per node, a parallel supercomputer is about 20 times more expensive. Since this makes supercomputers irrelevant for the expected users, even when considering the use of a supercomputing center, we have done little research in this direction.

It is however possible to use more advanced communication hardware for Beowulf clusters, for example Myrinet (www.myri.com). This should improve latency and thus maximum speed-ups by a factor of 10-50.

- Finally, it should be mentioned that, while for 10 Mbit Ethernet the main limiting factor was the hardware, for Gbit Ethernet this is no longer true: Special purpose implementations [56] bring Gbit Ethernet in the range of Myrinet. It is unclear if these improvements will make it into the mainstream.

Alternatively, one can consider other means of speeding up the computation. A possibility is to replace day-to-day replanning by within-day replanning, as discussed in Sec. 4. Experiments have shown that this reduces the number of necessary iterations considerably [42]. Possible distributed implementations of this are discussed in Sec. 4.2.

4.2 Distributed computing of plans generation

Additional complications come in with within-day replanning (Sec. 3.5) and with non-local interaction. Two examples:

- Re-planning. On a single CPU, a traveler calling the re-planning subroutine will just suspend the traffic simulation. On a parallel computer, if one traveler on one CPU does this, *all* CPUs have to suspend the traffic simulation since it is not

possible (or very difficult) to have simulated time continue asynchronously. A more plausible approach is to have the re-planning module on a different CPU. The traveler then sends out the re-planning request to that CPU, and the traffic simulation keeps going. Eventually, the re-planning will be finished, and it will be sent to the traveler, who picks it up and starts acting on it. An experimental implementation of this using UDP (User Datagram Protocol) for communication shows that it is possible to transmit up to 100 000 requests per second per CPU [47], which is far above any number that is relevant for practical applications. This demonstrates that such a design is feasible and efficient.

An additional advantage of such a design is that the implementation of a separate “mental map” (Sec. 3.6) for each individual traveler does not run into memory or CPU-time problems. Specific route guidance services can be simulated in a similar way.

- Non-local interaction between travelers. So far, everything assumes that travelers make autonomous decisions, and they interact in the micro-simulation only. This is however not always the case, for example for ride sharing, or when family members re-organize the kindergarten pick-up when plans have changed during the day. This will necessitate complicated negotiations between agents, and neither the models nor the computational methods for this are developed.

Some readers may have noticed that, in particular in the first example, success of the re-planning operation is not guaranteed. For example, the new plan may say to make a turn at a specific intersection, and by the time the new plan reaches the traveler, she/he may have driven past that point. Such situations are however not unusual in real life – how often does one recognize that a different decision some time ago would have been beneficial. Thus, in our view the key to success for large scale applications it to not fight asynchronous effects but to use them to advantage. For example, once it is accepted that such messages can arrive late, it is also not a problem to not have them arrive at all, which greatly simplifies message passing.

This design is similar to many robot designs, where the robots are autonomous on short time scales (tactical level) while they are connected via wireless communication to a more powerful computer for more difficult and more long-term time scales (strategic level); see, e.g., Ref. [57] for robot soccer. Also, it seems that the human body is organized along these lines – for example, in ball catching, it seems that the brain does an approximate pre-“computation” of the movements of the hands, while the hands themselves (and autonomously) perform the fine-tuning of the movements as soon as the ball touches them and haptic information is available [58]. This approach is necessitated by the relatively slow message passing time between brain and hands, which is of the order of 1/10 sec, which is much too slow to directly react to haptic information [59].

5 Outlook: Dynamics of networks

A further step for transportation simulations could be to make infrastructure changes (such as the addition of roads or train connections) endogeneous to the simulation pack-

age. This would mean that one would enable the simulation system to autonomously find out where changes to the transportation systems should be done, and include them into the system. On the simpler level of trail selection, this has been done by the method of active walkers [60]; Yamins et al investigate methods to grow urban roads [61]. For urban planning, one would have to make assumptions about political power distributions and related policies, but based on those it should be possible to run such a model. If it would yield anything useful will remain an open question for the foreseeable future.

6 Conclusion

For many systems, the dynamics does not unfold on “flat space”, but on a graph or network. Although many concepts of dynamical systems still apply, they need to be adapted for dynamics on a graph. As a non-mathematical example, look at the visualization of dynamics on a graph, which is rather different from visualization of dynamics in two- or three-dimensional space.

Transportation simulation is a prime example of a real-world dynamical system on a graph. It is particularly interesting, since the one-dimensional dynamics on the links interacts with the networks aspect. For example, kinematic waves, as described by the Burgers equation or by an Asymmetric Stochastic Exclusion Process, can travel through an intersection, causing complicated dynamics there [62, 63]. In fact, very little seems to be known of these link-network-interactions, especially for large systems with many links (roads) and vertices (intersections).

In addition, the particles/agents in traffic systems are “intelligent”. This means that they have strategic, long-term goals, with the consequence that no two particles are interchangeable, and that different particles, when confronted with the same situation, can make different decisions. In practical terms, for transportation simulations this “intelligence” involves aspects like route choice, mode choice, or activity generation. Moreover, agents adapt or learn, which means that they should be able to remember past behavior and past performance, to construct new plans, and to try them out.

For large scale scenarios, distributed computing is a necessity. The typical starting point is domain decomposition of the traffic micro-simulation, which means that each CPU runs the micro-simulation on a piece of the network. For efficiency reasons, this implies that the “intelligence” modules need to be separate from the traffic simulation itself. Mapping the resulting system well on parallel computer architectures seems to be a necessity for efficient large scale transportation simulations.

Finally, one can look at the re-organization of the transportation network as a consequence of a political process. This aspect is touched only very briefly.

In summary, transportation simulations combine elements from many areas, ranging from dynamical systems via networks and graph theory to socio-economic human behavior. The current technology is advanced enough to start helping with policy decisions, yet many aspects remain unsolved and offer challenging problems for years to come.

Acknowledgments

Los Alamos National Laboratory makes the TRANSIMS software available to academic institutions for a small charge.

The Swiss Federal Administration provides the input data for the Switzerland studies.

References

- [1] Y. Sheffi. *Urban transportation networks: Equilibrium analysis with mathematical programming methods*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1985.
- [2] Michael Patriksson. *The Traffic Assignment Problem: Models and Methods*. Topics in Transportation. VSP, Zeist, The Netherlands, 1994.
- [3] Stochastic and dynamic models in transportation, special issue, 1993.
- [4] H.P. Simão and W.B. Powell. Numerical methods for simulating transient, stochastic queueing networks. *Transportation Science*, 26:296, 1992.
- [5] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221, 1992.
- [6] D. Chowdhury, L. Santen, and A. Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329(4–6):199–329, May 2000.
- [7] K. Nagel, D.E. Wolf, P. Wagner, and P. M. Simon. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, 58(2):1425–1437, 1998.
- [8] 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, 1997.
- [9] R. Wiedemann. Simulation des Straßenverkehrsflusses. Schriftenreihe Heft 8, Institute for Transportation Science, University of Karlsruhe, Germany, 1994.
- [10] S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, University of Cologne, Germany, 1997. See www.zpr.uni-koeln.de.
- [11] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51(2):1035, 1995.
- [12] D.M. Beazley, P.S. Lomdahl, N. Gronbech-Jensen, R. Giles, and P. Tamayo. Parallel algorithms for short-range molecular dynamics. In D. Stauffer, editor, *Annual reviews of computational physics III*, pages 119–176. World Scientific, 1995.
- [13] N. Wu and W. Brilon. Personal communication.

- [14] C.F. Daganzo. Requiem for second-order fluid approximations of traffic flow. *Transportation Research B*, 29B(4):277, 1995.
- [15] H. Spohn. *Large scale dynamics of interacting particles*. Springer, Berlin, 1991.
- [16] D. L. Gerlough and M. J. Huber. *Traffic Flow Theory*. Special Report No. 165. Transportation Research Board, National Research Council, Washington, D.C., 1975.
- [17] B. S. Kerner and P. Konhäuser. Structure and parameters of clusters in traffic flow. *Physical Review E*, 50(1):54, 1994.
- [18] D. Helbing. Gas-kinetic derivation of Navier-Stokes-like traffic equations. *Physical Review E*, 53(3):253–282, 1996.
- [19] P. Nelson. Synchronized traffic flow from a modified lighthill-whitman model. *Physical Review E*, 62(2):R6052–R6055, 2000.
- [20] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics*, Oct/Nov 2001. Also www.arXiv.org, cond-mat/0012229.
- [21] D. Chowdhury et al. Comment on: “Critical behavior of a traffic flow model”. *Physical Review E*, 61(3):3270–3271, 2000.
- [22] K. Nagel, Chr. Kayatz, and P. Wagner. Breakdown and recovery in traffic flow models. In Y. Sugiyama et al, editor, *Traffic and granular flow '01*. Springer, Heidelberg, in press.
- [23] K. Nagel and M. Paczuski. Emergent traffic jams. *Physical Review E*, 51:2909, 1995.
- [24] O. Biham, A. Middleton, and D. Levine. Self-organization and a dynamical transition in traffic-flow models. *Physical Review A*, 46:R6124, 1992.
- [25] F.C. Martinez, J.A. Cuesta, J.M. Molera, and R. Brito. Random versus deterministic 2-dimensional traffic flow models. *Phys. Rev. E*, 51(2), 1995.
- [26] T. Nagatani. Jamming transition in a two-dimensional traffic flow model. *Physical Review E*, 59(5):4857–4864, 1999.
- [27] J. Freund and T. Pöschel. A statistical approach to vehicular traffic. *Physica A*, 219(1–2), 1995.
- [28] A. de Palma and F. Marchal. Real case applications of the fully dynamic METROPOLIS tool-box: an advocacy for large-scale mesoscopic transportation systems. *Networks and Spatial Economics*, forthcoming.
- [29] DYNAMIT, 1999. Massachusetts Institute of Technology, Cambridge, Massachusetts. See its.mit.edu.

- [30] 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.
- [31] T. Schwerdtfeger. *Makroskopisches Simulationsmodell für Schnellstraßennetze mit Berücksichtigung von Einzelfahrzeugen (DYNEMO)*. PhD thesis, University of Karlsruhe, Germany, 1987.
- [32] A. Dupuis and B. Chopard. Cellular automata simulations of traffic: a model for the city of Geneva. *Networks and Spatial Economics*, forthcoming.
- [33] G. D. B. Cameron and C. I. D. Duncan. PARAMICS — Parallel microscopic simulation of road traffic. *J. Supercomputing*, 10(1):25, 1996.
- [34] H. A. Rakha and M. W. Van Aerde. Comparison of simulation modules of TRANSYT and INTEGRATION models. *Transportation Research Record*, 1566:1–7, 1996.
- [35] R. R. Jacob, M. V. Marathe, and K. Nagel. A computational study of routing algorithms for realistic transportation networks. *ACM Journal of Experimental Algorithms*, 4(1999es, Article No. 6), 1999.
- [36] M. Ben-Akiva and S. R. Lerman. *Discrete choice analysis*. The MIT Press, Cambridge, MA, 1985.
- [37] J. A. Bottom. *Consistent anticipatory route guidance*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.
- [38] E. Cascetta and C. Cantarella. A day-to-day and within day dynamic stochastic assignment model. *Transportation Research A*, 25A(5):277–291, 1991.
- [39] R. Palmer. Broken ergodicity. In D. L. Stein, editor, *Lectures in the Sciences of Complexity*, volume I of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 275–300. Addison-Wesley, 1989.
- [40] Issues of feedback routing in iterated transportation simulations. in preparation.
- [41] J. Esser. *Simulation von Stadtverkehr auf der Basis zellularer Automaten*. PhD thesis, University of Duisburg, Germany, 1998.
- [42] M. Rickert. *Traffic simulation on distributed memory computers*. PhD thesis, University of Cologne, Germany, 1998. See www.zpr.uni-koeln.de/~mr/dissertation.
- [43] K.W. Axhausen. A simultaneous simulation of activity chains. In P.M. Jones, editor, *New Approaches in Dynamic and Activity-based Approaches to Travel Analysis*, pages 206–225. Avebury, Aldershot, 1990.
- [44] S. T. Doherty and K. W. Axhausen. The developement of a unified modelling framework for the household activity-travel scheduling process. In *Verkehr und Mobilität*, number 66 in Stadt Region Land. Institut für Stadtbauwesen, Technical University, Aachen, Germany, 1998.

- [45] H. Unger. An approach using neural networks for the control of the behaviour of autonomous individuals. In A. Tentner, editor, *High Performance Computing 1998*, pages 98–103. The Society for Computer Simulation International, 1998.
- [46] H. Unger. *Modellierung des Verhaltens autonomer Verkehrsteilnehmer in einer variablen staedtischen Umgebung*. PhD thesis, Universitaet Rostock, in preparation.
- [47] Chr. Gloor. Modelling of autonomous agents in a realistic road network (in german). Diplomarbeit, Swiss Federal Institute of Technology ETH, Zürich, Switzerland, 2001.
- [48] S. Weinmann. *Simulation of spatial learning mechanisms*. PhD thesis, Swiss Federal Institute of Technology ETH, Zürich, Switzerland, in preparation.
- [49] TRANSIMS, TRansportation ANalysis and SIMulation System, since 1992. See transims.tsasa.lanl.gov.
- [50] A. Voellmy, M. Vrtic, B. Raney, K. Axhausen, and K. Nagel. Status of a transims implementation for switzerland, forthcoming.
- [51] J. Esser and K. Nagel. Iterative demand generation for transportation simulations. In D. Hensher and J. King, editors, *The Leading Edge of Travel Behavior Research*, pages 659–681. Pergamon, 2001.
- [52] K. Nagel and P. Wagner (editors). Special issue on regional transportation simulations. *Networks and Spatial Economics*, forthcoming.
- [53] N. Cetin. personal communication.
- [54] K. Nagel and M. Rickert. Parallel implementation of the TRANSIMS micro-simulation. *Parallel Computing*, 27(12):1611–1639, 2001.
- [55] P. Gonnet. A thread-based distributed traffic micro-simulation. Term project, Swiss Federal Institute of Technology ETH, Zürich, Switzerland, 2001.
- [56] <http://pdswww.rwcp.or.jp/>, since 1993.
- [57] J.H. (editor) Kim. Special issue about the first micro-robot world cup soccer tournament, MIROSOT. *Robotics and Autonomous Systems*, 21(2):137–205, 1997.
- [58] D. Sternad. personal communication.
- [59] J.D. Rothwell. *Control of Human Voluntary Movement*. Chapman and Hall, 1994.
- [60] D. Helbing, F. Schweitzer, J. Keltsch, and P. Molnar. Active walker model for the formation of human and animal trail systems. *Physical Review E*, 56(3):2527–2539, 1997.
- [61] D. Yamins, S. Rasmussen, and D. Fogel. Growing urban roads. *Networks and Spatial Economics*, forthcoming.

- [62] H.Y. Lee, H.-W. Lee, and D. Kim. Origin of synchronized traffic flow on highways and its dynamic phase transitions. *Physical Review Letters*, 81(5):1130–1133, August 1998.
- [63] D. Helbing and M. Treiber. Gas-kinetic-based traffic model explaining observed hysteretic phase transition. *Physical Review Letters*, 81(14):3042–3045, 1998.