Part B

NETWORK TRAFFIC

Chapter 8

Some issues of traffic in networks

The text now moves away from single lane traffic flow considerations towards road networks with ramps connecting different segments. Such a road network could be the freeway network of the German land Northrhine Westfalia, but it could also be a simple "toy" example. This chapter, the first of three, reviews other work on network traffic. The first section explains the conflict User Optimum vs. System Optimum. Section 8.2 explains traffic management systems and congestion pricing theories. In Section 8.3, it is reviewed how simulation models handle traffic streams, which is related to Section 8.4, about traffic assignment methods, and Section 8.5, about origin-destination information. Section 8.6 gives an overview of how two-dimensional cellular automata techniques have been used for traffic networks.

8.1 Nash Equilibrium vs. System Optimum

Again, as a first approximation, one would like to use the network infrastructure as efficiently as possible. Even if one discards environmental concerns, practicability, personal preferences, etc., there is still the problem that "efficient use" means something different for users than for a global observer [3, 22]:

- A *Nash Equilibrium* (NE; also called *User Optimum*) is defined as a state where no user can reduce his/here costs by unilaterally changing his/her behavior.
- A *System Optimum* (SO) is defined as a state where the sum of all individual costs is globally minimal.[1]

These system states are often in conflict, because, as is common knowledge, individual interests rarely coincide with global welfare. An example will be given in Chapter 9.

8.2 Traffic management

Different projects, as well in the U.S. as in Europe, such as PROMETHEUS, DRIVE, SCOPE, etc., attempt to make road traffic safer and more efficient (see, e.g., [54, 55, 70, 127]). I will shortly outline

[1] Whereas all SOs need to have the same sum of costs, there can be more than one SO. However, one should take this point from the pragmatic side, since in real, fluctuating road traffic, it is infeasible to find an exact optimum.

the part of these projects which are most important here. This is the traffic management part. The idea of traffic management is to guide vehicles in an efficient way through the network, i.e.

- to give them routing information if they are in an unfamiliar area,
- to warn them of incidents and to guide them around incident locations in an efficient way, and
- to redistribute traffic such that the network is used more efficiently.

The necessary data collection and data transmission are parts of the projects Advanced Transport Telematics (ATT, in Europe) or Intelligent Vehicle Highway Systems (IVHS, in the U.S.). Advanced Traffic Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS) are intended for driver information and traffic management.

Providing routing information for unfamiliar drivers does not pose any big problems. For incident management at least an improvement (if not an optimal solution) is foreseeable. For the third point, however, the benefits are unclear. It depends (i) on in how far today's commuter traffic is already operating at a Nash Equilibrium, or could benefit from better driver information, and (ii) on how much one assumes a System Optimum to be better than a Nash Equilibrium.

Mahmassani and Peeta [102] report from simulation results a 15% advantage of the SO over the NE, whereas Zallmann and König [184] found practically no difference. Hall [55] expects possible benefits of IVHS mostly in non-recurrent (i.e. caused by an incident) congestion. Emmerink et al [38] evaluate, based on simulations using behavioral rules for travelers, possible benefits of ATIS systems as a function of market penetration. Their results with respect to recurrent vs. non-recurrent congestion are similar to Hall's.

Economists do not believe that it will be possible to push people towards an SO if that means a disadvantage for the user [3]. They propose congestion pricing as a solution, and prove for a bottleneck situation that, when the social cost of using the bottleneck equals the toll, then the SO is reached, and each user pays exactly the amount of money he/she is ready to pay for the time he/she saves. Social cost is defined as the sum of all additional costs one additional user imposes on all users.

MacKie and Varian [101] propose a similar method for the Internet. The problem is very similar indeed since each longer message when submitted to the Internet is broken into pieces which are transmitted in separate packets. And each packet searches its own path through the network.

Note that all these policies propose *congestion* pricing, not road pricing. That means that it is not the use itself of the infrastructure which is priced, but the social costs one imposes to others (and himself) by over-using it.

A problem with these policies so far is that it is unclear how to implement them in a spatial, fluctuating, and unpredictable environment. MacKie and Varian propose that packets carry a certain amount of money which they use to pay congested parts of their travel. But what if, shortly before they reach their target, there is a prohibitively expensive bottleneck? Something similar is true for road traffic, where it cannot be expected that commuters want to live with a toll which is highly fluctuating from day to day and from time slice to time slice just because congestion toll is supposed to match social costs.

Thus, one has to develop this line of thinking further. Chapter 9 will present a situation where even in a highly fluctuating environment a slowly self-adjusting congestion toll leads to an acceptable result.

If people do not like pricing, other coercive measures might lead to success. In Chapter 9, one will find an example where a simple traffic light would already help considerably.

8.3 Modeling traffic streams in road networks

Since many traffic assignment methods are directly coupled to simulation models, it is necessary to review some basic features of network traffic models. Contrary to Chapter 2 with the focus on traffic flow, this section will concentrate on issues which are related to traffic stream management.

Two issues are of particular importance here: the implementation of directional information (trip plans vs. turn counts), and the handling of queues or jams. Both issues are interrelated, as will be seen.

8.3.1 Trip plans vs. turn counts

Most traffic simulation models to date, even microscopic, use turn counts as data input. This means that, for each intersection and each time slice, one either has the percentage or the number of cars relating incoming to outgoing travel, i.e. the number of left turns, right turns, cars going straight, ..., for each incoming road.The opposite, basing the simulation model on trip plans (also called paths), means that each car individually knows at each intersection where it wants to continue its trip.[2]

The decision to use turn counts in the past was due to pragmatic reasons: Turn count data is much easier available. Moreover, on a certain level of modeling, it does not make a difference. For example, a macroscopic model based on the Lighthill-Whitham equations [31, 29, 133] produces an average over possible traffic situations as output [134]. Thus, as long as one uses *average* turn counts (for example monthly averages of all weekdays) as input, then simulation and input data are compatible.

The situation becomes more problematic when unexpected incidents occur. Assume, for example, that a road is suddenly blocked, e.g. by an accident (Fig. 2.2). Then, a model based on turn-counts has no means to predict the resulting reorganization of the traffic patterns. A model based on trip-plans, however, can, based on behavioral rules or on a traffic management strategy, re-plan the affected trips.

8.3.2 Handling of queues

All simulation models obey mass conservation and therefore handle queues in one way or the other: If a segment cannot deal with a certain flow demand, and if vehicles have no alternative, the vehicles have to wait somewhere. There are, however, differences in how realistic queues are handled:

- Macroscopic models [129] often handle flow-density-relationships on a link basis. If, for example, an intersection at the end of a link is blocked, this immediately increases the average density of the whole link.
- Point queue models [157] have queues before bottlenecks (e.g. intersections) which have no spatial extension. Back-spill of jams, which produces grid-lock, will not occur.
- Deterministic models (e.g. fluid-dynamical models based on the Lighthill-Whitham theory [130, 31, 29, 133], but also deterministic microscopic models [17, 116, 117] or combined approaches [154]), lead to *average* queues. Such models will give an deterministic answer to a question if a certain incident will cause, say, a network breakdown or not.
- *Stochastic* microscopic models can produce different scenarios from the same input data. For example, sometimes a queue can spill back to another intersection and cause a network break-

[2] I will, when that makes writing easier, personalize the car. DVU (driver-vehicle-unit) is used in some texts.

down, whereas in the average, the queue does not grow that long. For obtaining interpretable results, averaging over many different scenarios may be necessary.

This method is the only one available if one is interested in fluctuations, e.g. the *probability* that a certain incident can lead to a network breakdown.

Note that the average itself can be different from the averaging models, if, for example, the network breakdown has enough weight in the averaging procedure. If one wants, for example, obtain measures of efficiency for a certain traffic management strategy, then one has to take network breakdowns into account, and if they do in fact happen, than deterministic models will give a wrong answer.

The connection to the turn counts / trip plans issue is that, as soon as either the simulation or the turn counts data or both are no longer averaged (for example in real time applications, where the turn counts are not averaged), driving the model by turn counts is inconsistent as soon as fluctuations become large, e.g. by traffic jams: For example, suppose a situation where the deterministic model predicts, say, 500 vehicles at a specific intersection in a certain time slice. Meanwhile, in reality all vehicles which want to make a left turn are caught in some jam somewhere, and there are only 300 vehicles, which all go straight or to the right. When using these turn-counts percentages, the simulation will send too few cars to the left in this time slice. Research is necessary to clarify how much averaging is necessary to avoid such situations.

There will be data assimilation methods to handle this, especially in real time applications, when one does not want to give up the higher simulation speed of a macroscopic model. This is, though, not a conceptually satisfying solution. A better solution is to directly drive the model by trip plans. This does not ensure compliance with reality, but at least one can be sure that the simulation follows *one* of the possible trajectories of the system in state space.

There are a number of models in the literature which could easily support trip plans. Nonetheless, this is usually not the case. For example, both NETSIM and FREESIM (based on INTRAS) use turn counts [146].

Meanwhile, there are some notable exceptions. CONTRAM moves *packets* of vehicles with identical origin-destination pairs through the network. Emmerink and coworkers [38] use a model where 300 packets are used for each simulation run. Yet, the use of packets is only possible as long as there are enough vehicles with the same path in each space-time unit of the simulation. CONTRAM achieves this by using time-steps of 15-30 minutes and by taking links as the basic unit of space. Emmerink and coworkers use a corridor network, i.e. all vehicles start from the same node and drive to the same node. In such a case, the number of possible paths is much smaller than in a full network. Cremer and Meissner [29] consider only two routing alternatives and model them, in a fluid-dynamical model, as portions of the flow.

In terms of microscopic models, DYNASMART is a path-based microscopic simulation model used for urban traffic assignment [102]. Zallmann and König [184] use a traditional turn-count based model for the traffic assignment, but completely vehicle-based models for the microsimulations. The demo version of the TRANSIMS microsimulation [165] also uses trip plans. PARAMICS [181] includes behavioral rules and therefore trip plans for travelers.

8.3.3 Origin-destination information

Turn counts data itself does not give any information on *correlations* between turns, for example the information if a vehicle making a left turn at a certain intersection is likely to make a right turn at the next intersection.

Yet, turn counts data contains temporal correlations: Assume that one has, at a certain time and at a certain intersection A, many cars making a left turn. Assume further that one has, at the next intersection B and after the time one needs to drive from A to B, many cars making a right turn. Than it is *likely* that these cars are the same. These are the kind of correlations which are used by current methods of O-D estimation [27].

It is clear though that this method cannot reveal long distance correlations. Here, it seems more promising to estimate O-D relations from demographic data, where one has, for example, information on where people live and where they work [165, 153]. And then, evidently, one can run microsimulations based on that data and validate the simulation results with turn counts or other observational data.

As mentioned above, a major reason that one currently uses turn counts to drive the simulation models is that this data is, at least in principle, relatively easy to obtain: One just counts the cars. Origin-destination (O-D) data is much more difficult to obtain. It is though open which is the more economic method when dealing with large systems. Reliable automatic counting systems seem difficult to develop [134], and it might be better in the long run to develop a method method based on demographic data and simulation which can do with fewer sensors for validation and incident detection.

8.4 Traffic assignment methods

8.4.1 Static equilibrium methods

A commonly used method in traffic science is *static equilibrium assignment*, usually only called equilibrium assignment [164, 184, 155]. The problem is equivalent to a static multi-commodity network flow problem.

Commonly, people use a relaxation algorithm which works as follows:

(i) Assume that a network of nodes and links/arcs with link cost functions and a matrix of origin-destination-streams (O/D-matrix) are given. Choose one of the streams.

(ii) For the O-D-stream, find the cheapest path through the network.

(iii) Calculate the resulting density on each link. Calculate link costs (for example travel time on that link) from this density.

(iv) Take the next O-D-stream and go back to step (ii).

This algorithm is known in the traffic science literature as *All-or-nothing* assignment method because is assigns each traffic stream at once. It collapses, for example, when there is an O-D-demand where no path exists which carries the whole demand.

There are several variations of this general scheme in the literature, e.g.:

- In order to avoid the "all-or-nothing" problem, one can replace (*ii*) by [184, 155]

  (ii') For the O-D-stream, calculate the shortest path through the network. *Assign only* $1/k$ *of the demand to it.*

- More recent implementations often use a simulation model instead of step (iii) [164].
- The above scheme may only be taken as a first guess, which is followed by a relaxation algorithm where small parts of the streams are exchanged [184].

Since in the assignment, each stream optimizes its own cost function, the result corresponds to a Nash Equilibrium.

Note that the equilibrium assignment assumes a steady state solution, i.e. a solution where the traffic streams according to the O-D-matrix are stationary.

8.4.2 Queues

Static equilibrium assignment is obviously problematic for time-varying demands. But it completely breaks down above capacity: Queues develop at the entrances of the overloaded links, and equilibrium solutions become history dependent. In other words: Static equilibrium assignment assumes static traffic streams. Yet, under static *and* congested circumstances, queues would be infinitely long.

Arnott and coworkers [3] treat the simplest case of this problem: A single bottleneck during the rush period. It is assumed that, for a certain time, transportation demand through the bottleneck exceeds supply, such that a queue develops at the entrance of the bottleneck. After the rush hour, the queue decays due to reduced demand. One-dimensional bottleneck cases can be treated analytically.

Kuwahara and Newell [89] extend the idea to a single core city: It is assumed that workers want to reach a point-like central business district (CBD) during the morning rush hour, and that queues develop on all arterials leading to the CBD.

One option to deal with the queuing problem would be to make the assignment procedure time-dependent, and then to use a simulation model to calculate the cost functions. However, as long as the simulation model is driven by turn counts, the non-queueing planning process and the simulation are incompatible. It is again the same problem as above, that, for example, all people who are supposed to make a left turn at a certain intersection according to the assignment can be caught in a queue. Meanwhile, the turn-counts keep on sending the predetermined percentage of all arriving people to the left, which then is wrong. And the error is likely to amplify during later time-steps. Indeed, differences between the simulation-based assignment and reality are reported [164].

One way out is to use a simulation model driven by individual trip plans in order to calculate the cost function. The main reason why this is not done is that conventional microscopic models, when used with trip plans, are computationally prohibitive, especially in an iterative assignment loop.

On the algorithmical side, it is only recently that congestion/queueing has been taken into account. See the collection of articles in [126]. However, each of these papers has at least one of the following restrictions: (i) No algorithmical solution is proposed. (ii) The algorithm is only valid if one has a single destination. (iii) The algorithm does not take into account that the traffic pattern will vary during the driving time; in other words: A path is assigned to a driver when she enters the network, and this path is optimized *based on a snapshot of the current traffic situation.* Each of these restrictions prevents the practical application for the dynamic congested traffic problem. See [129] for a a framework for a dynamic treatment of multidestination traffic networks.

8.5 Cellular automata models for traffic in road networks

8.5.1 Two-dimensional cellular automata models on a grid

Two-dimensional cellular automata models [30, 112, 119, 166] are useful to understand the *grid-lock transition* (also called jamming transition by several authors) occurring on overcrowded road networks: A blocked intersection leads to a queue, which blocks an intersection somewhere else, etc., until a large portion of the network is in grid-lock.

The CA models represent traffic on a two-dimensional grid where vehicles can move to adjacent grid-points in each iteration. In all the models so far, the vehicle dynamics follows the deterministic asymmetric exclusion process, i.e. $v_{max} = 1$ and $p = 0$ in the language of Chapter 3. Meanwhile, they employ more sophisticated rules for making turns at intersections.

The models show that a grid-lock transition occurs at a certain vehicle density ρ_c; above that density, most or all vehicles are caught in a few very large grid-lock clusters; below that density, no stationary grid-lock clusters exists. Yet, in these models the maximum flow is given entirely by the capacities of the intersections, which is not always realistic, e.g. for arterials. Nagatani [118] investigates a case where a portion of the intersections is two-level.

8.5.2 Road networks composed of cellular automata segments

A more realistic, but more laborious, approach is to connect one-dimensional CA traffic flow segments to road networks. This is similar to conventional microscopic traffic flow simulation [124], except that the CA model will run much faster (see Chapter 11).

As mentioned in Chapter 2, Schütt [151] describes a tool-box of urban and rural roads and freeways plus the corresponding intersections in order to construct road networks.

Rickert [141] reports an implementation, based on this work here, with two lanes in each direction for arbitrary freeway networks. He describes results of extensive simulations with a network implementation of Northrhine-Westfalia.

Both models do not, however, implement trip plans, the missing of which have been identified as one of the major shortcomings in current methodologies.

I will, in the next two chapters, describe results obtained with a single-lane network implementation which includes trip plans.

Chapter 9

Predictability vs. network performance

This chapter is the first of two dealing with traffic in road networks. It considers a simple road network where, using a basic model of human decision behavior, issues such as User Equilibrium and Congestion Pricing are investigated in a simulation. An important results will be that traffic management makes traffic indeed more efficient, but also more fluctuating in certain quantities and therefore less predictable.

The chapter starts with the introduction of a new quantity, the vehicle-to-vehicle fluctuations for trip times. Then it introduces the specific network example which will be used, discusses traffic management aspects and the resulting set-up for the simulations. Section 9.3 presents the simulations results, followed by a summary.

9.1 Variability and predictability of travel times

Measuring the life-time distribution of traffic jams as in Chapters 5 and 6 is convenient for a theoretical understanding, but it is not very useful for everyday traffic. The probably most important reason for this is that life-times of jam-clusters are practically not amenable to measurements in reality.

A quantity which is much easier to measure and which is extremely relevant in the context of transportation management is the individual travel-time and its variation from vehicle to vehicle using the same route. For the following simulations, we still use a closed loop of size L. We define a subsegment of length $l < L$ and measure, for each car, the time t_l between entry and exit of this subsegment.

The relative variation of travel-times is defined as

$$\sigma(t_l) := \frac{\sqrt{\langle (t_l - \langle t_l \rangle)^2 \rangle}}{t_l} \; .$$

$\langle \dots \rangle$ denotes the average over all cars during the simulation; $\langle t_l \rangle$ therefore is the average travel-time for all cars during the simulation.

This can be seen as a measure of the predictability of the trip time. When a vehicle enters the measurement section, one would like to tell the driver how much time he will need. One would tell him $\langle t_l \rangle$ as expected time; and for the expected relative error of this prediction one would use σ.

Results of these measurements as a function of density are shown in Fig. 9.1. A system of length $L = 10^3$ is used where trips along a designated subsegment of $l = 100$ are measured. The simulation runs, after 10^4 iterations to let the transients die out, for 10^5 time steps, and every time a car finishes

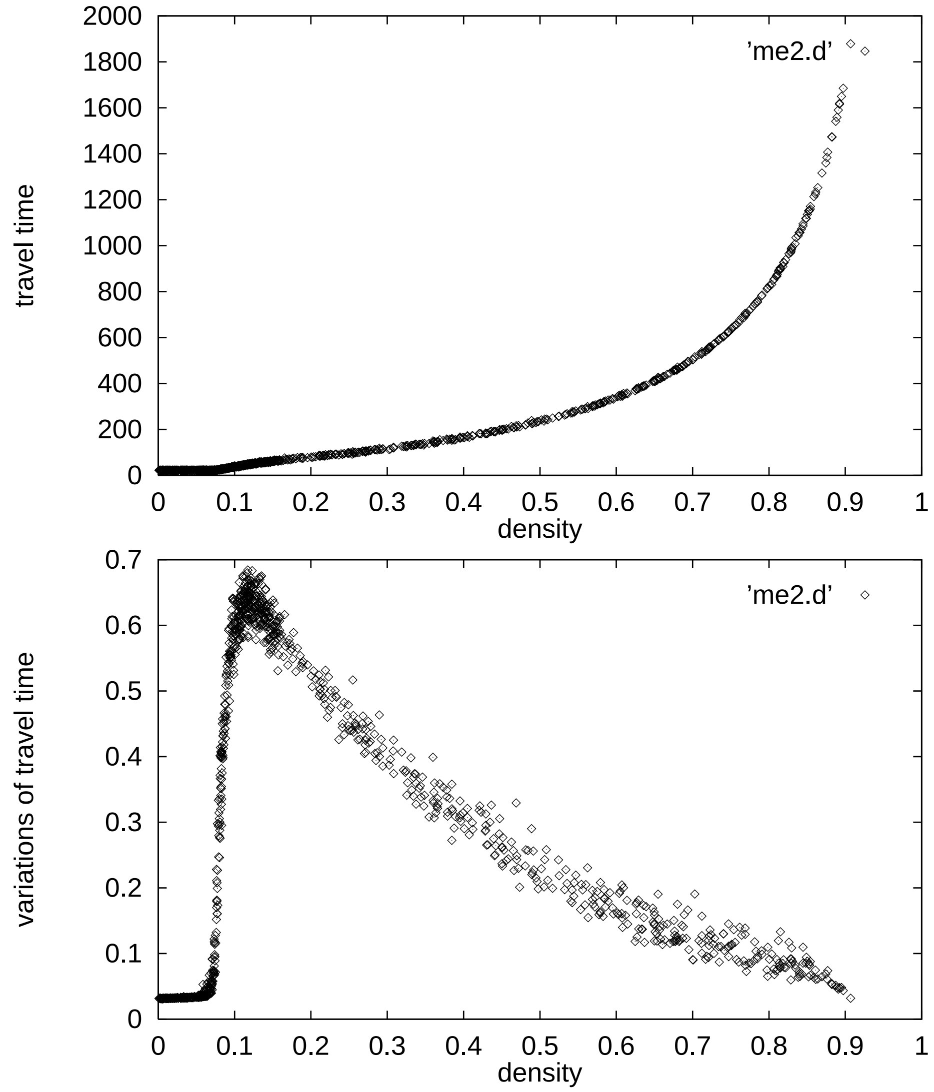

Figure 9.1: Travel time and variations of travel time as a function of density. System size $L = 10^3$, length of traveled subsection $l = 10^2$, measured time $T = 10^5$ time-steps.

a complete travel along the measurement subsegment, its travel time is taken into account for the average.

One clearly sees that both the travel time and the vehicle-to-vehicle fluctuations are approximately constant up to a density around 0.09. There, the travel time starts to rise as a function of density. Meanwhile, the fluctuations go up very steeply and reach a maximum near $\rho = 0.11$. In other words, one can not only show that the region of maximum throughput shows near-critical behavior in a theoretical sense, but also that this behavior has practical consequences: It implies that, passing from slightly below to slightly above capacity, one comes from a regime where the travel time is predictable with an accuracy of approx. $\pm 3\%$ to a regime where the error climbs up to more than $\pm 65\%$.

9.2 A simple transport network

Let us now move away from the single lane closed loop system to a single lane highway network with ramps connecting the different segments. The travelers on this network have route plans so that they know which ramps they need to exit to reach their individual destinations. We assume that each traveler always has the same origin/destination pair. Each traveler remembers the last travel time for each alternative route between his or her origin and destination. The network may have traffic density sensors at specified locations which can be used to identify congested areas and perhaps introduce toll for the use of such links. The travelers are able to re-plan depending on their aggregated transportation costs which is their remembered travel time plus eventual toll. Such a sensor setup is an example of a (Advanced) Transportation Informational System (ATIS), and the introduction of toll for the use of highly congested links is a simple example of an (Advanced) Transportation *Management* System (ATMS) [55, 70]. The rationale behind such a toll policy is to make the highway traffic more efficient by pushing a larger part of the system towards the density corresponding to maximum flow. Interestingly, this implies that more traffic intentionally will be moved into the critical regime as defined above which in turn will increase the fluctuations of the travel times as well as the non-predictability of transportation system dynamics. This effect is the topic of this section.

9.2.1 Ramps

In order to simulate a network, one first needs a reasonable algorithms for transferring vehicles from one road to another at junctions. This involves two parts: Including the vehicle into the traffic stream on the target road; and then deleting it from the source road.

Unfortunately, introducing an additional car into a given traffic stream can cause problems. Just adding the inflow to the traffic on the main road easily leads to disturbances which (i) block the traffic on the main road, and (ii) lead to an outflow, downstream from the ramp, which is *below* capacity. For this reason, we chose an algorithm where access to the main road is only possibly when there is sufficient space between vehicles. We believe that this is realistic enough to represent metered ramps (i.e. ramps with regulated access), and since we are often concerned with the analysis of future traffic systems, it seems, in a first step, appropriate to model a technically advanced traffic control system here.

The algorithm works as follows. Imagine a ramp, as in common experience, as two parallel stretches of road; these parallel stretches have a length of 5 sites in the model. The target stretch is part of a longer road and therefore is connected at both ends, whereas the source stretch is only backwards connected. If there is a vehicle (velocity v) on the source stretch, then

- it looks, on the target stretch, for the next car ahead (which may be its neighbor; $\rightsquigarrow gap_{forward}$); and
- it looks for the closest car behind on the target stretch ($\rightsquigarrow gap_{backwards}$).

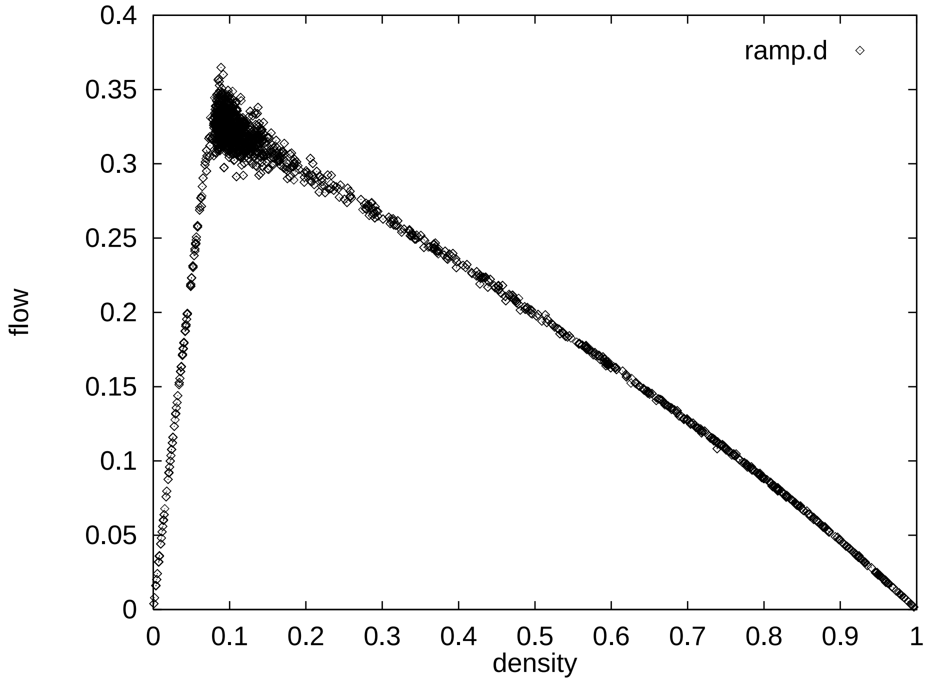

Figure 9.2: Fundamental diagram for ramp. A circular segment of length $L = 10^3$ may be partially bypassed by a second segment. 50% of the traffic uses the bypass; at the end of the bypass, it again merges with the main stream of the traffic. The measurements were taken at the part of the main segment where no bypass exists.

- Then the following rules are applied:

---

if $(gap_{forward} > v$ & $gap_{backwards} > v_{max})$
  change_lane
  $v = \max(v_{max}, gap_{forward})$ on new lane
else
  $v = 0$
  $(*)$
endif

One may imagine that this is emulating a ramp metering system, where a technical device upstream of the ramp determines where to fit in a car. The car then gets a green light and arrives at maximum speed, in between two other cars on the target road.

In line $(*)$ one has to take some precautions that the vehicle really does not move, which depends on the algorithmical structure.

The details of this algorithm will probably not matter for our results, as long as it allows maximum flow downstream from the ramp. That this indeed is the case is shown in Fig. 9.2, which may be compared to Fig. 3.6. It gives the fundamental diagram for a system with two road segments where one is a closed loop and the other one provides an alternative route for a certain length, connected to the main road by one exit and one entry ramp. Half of the vehicles use this alternate route. Density and throughput are measured on the undivided part.

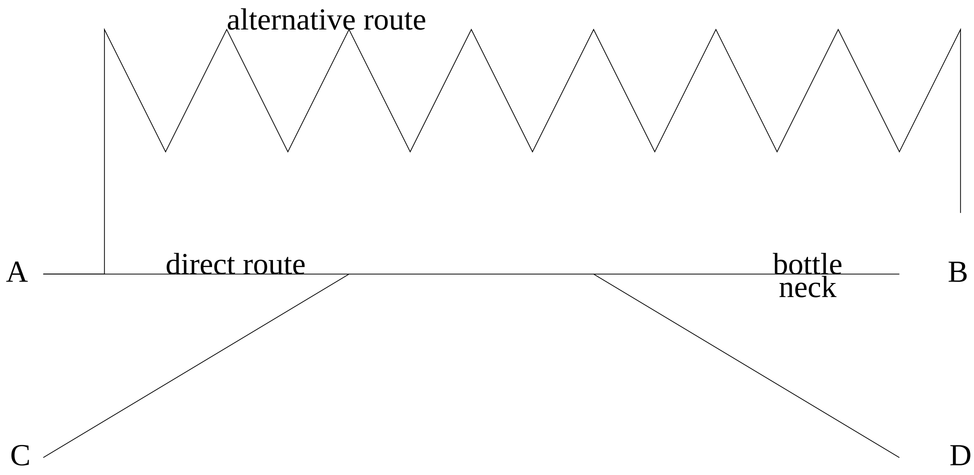

Figure 9.3: Schematic sketch of the network used for the simulations. Vehicles drive from A to B and can choose between the direct route and the much longer alternative route. On the direct route, the encounter a bottleneck. Other vehicles drive from C to D.

9.2.2 Nash Equilibrium versus System Optimum: An example

The conflict between Nash Equilibrium (NE) and System Optimum (SO) can perhaps best be illustrated in terms of a simple transport network example (a variation of [22]), especially as we will use the same example for simulation experiments later on.

Imagine a road from A to B with capacity q_{max}, with a bottleneck with capacity $q_{bn} < q_{max}$ before B (Fig. 9.3). Further imagine that there exists an alternative, but longer route between A and B. On the direct route from A to B additional travelers from C have to go to destination D. First assume that there are no travelers with origin in C.

If many drivers are heading from A to B, they will, without knowing anything about the overall traffic situation, all enter the direct road. In consequence, a queue builds up from the bottleneck.

A Nash-Equilibrium is defined as a situation where no agent (= driver) can lower his or her cost (= decrease travel time) by unilaterally changing behavior. Assuming that the drivers have complete information, this implies that, in the NE, their waiting time in the queue exactly compensates for the additional driving time on the alternative route.

Now assume that there are additional travel demands from C to D (see Fig. 9.4), with the exit to D lying shortly before the bottleneck. Obviously, this traffic is suffering from the bottleneck queue upstream (= left) of the bottleneck, and from these travelers' point of view it would be much better if the queue were located to the left of the ramp that the travelers from C use to enter the link. Note that moving the queue further upstream does not make any difference for the NE of the drivers originating in A.

This could, for example, be achieved by a traffic light near the entrance ramp from C. This light would restrict traffic flow from A to the bottleneck capacity. As a result, traffic between the traffic light and the bottleneck would flow freely, and the traffic from C to D would be much less disturbed.

Note, however, that in more complicated situations such actions are no longer possible. In our case, already some travel demand from A to D would make this traffic light a doubtful solution. And note, in addition, that the *steady state* optimum solution for our system would be to send all A-B travel via

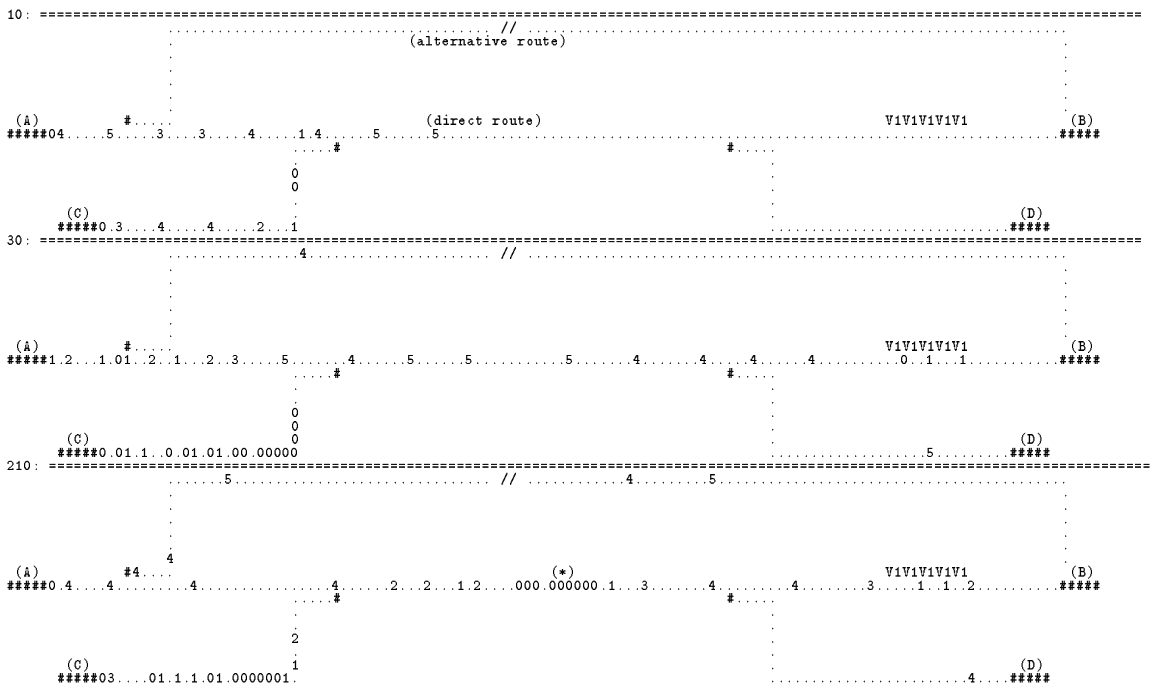

Figure 9.4: Schematic network representation with traffic, showing time-steps 10, 30, and 210 of a particular simulation. Traffic entering at (A) is bound for (B) and may use the "direct route" or the "alternative route". Traffic entering at (C) is bound for (D). The bottleneck is denoted by `V1V1V1V1V1` (maximum speed 1). One observes that the traffic coming from (C) has difficulties entering into the main stream; and—in time-step 210—a disturbance denoted by `(*)` has traveled backwards from the bottleneck.

the long route: In this way, throughput is maximized. However, for rush hour traffic, such a solution is no longer optimal.

A rather general way to push a traffic system from an NE towards an SO is to keep the density on each road at or below $\rho(q_{max})$, the density of maximum throughput. Then there would not exist queues anywhere in the system, since queues are regions of high density, This would ensure that additional traffic which is not using the bottleneck could proceed undisturbed. Note that this could for instance imply (in the limit of a perfect implementation) that drivers have to wait to enter the road network until sufficient capacity is available for them.

9.2.3 Egoistic drivers: Travel plans and individual decision logic

In our simple network, there are only two different types of travelers: Travelers from A to B, and travelers from C to D. Travelers from A to B can choose between the direct and the longer, alternate route. In order to make decisions, each AB-driver remembers his or her last travel-time on each of the two routes.

A traveler calculates expected costs [3] according to

$$cost_{direct} = toll + \alpha \cdot t_{direct}$$

and

$$cost_{alt} = \alpha \cdot t_{alt}$$

where $cost_{direct}$ and $cost_{alt}$ are the expected costs for the two route choices, *toll* is the toll for the current day (see below), t_{direct} and t_{alt} are the remembered travel times for each route, and α is a conversion factor which reflects trade-off between time and money. α could be different for each driver, but is uniformly equal to one in this work. α reflects "standard values of time", VOT, which can be looked up for traffic systems [6].

Then, each driver chooses the cheaper route, except that there is a 5% probability of error (which gives each driver a chance from time to time to update her information about the other possibility).

As long as the traffic dynamics is deterministic and completely uniform, this scheme leads to a Nash equilibrium [3]. However, in our case of stochastic traffic dynamics, this is no longer true: There is no easy way to define complete rationality because the interactions become too complicated. It is therefore better to speak of 'bounded rationality', which means that individual decisions are plausible instead of mathematically optimal. By dealing with stochastic traffic dynamics, the notions of economic equilibrium theory have to be used with care.

9.2.4 Space-time dynamics

Before we discuss how to determine the toll, we shortly turn to a space-time plot of the direct route from A to B (Fig. 9.5). As before, vehicle movement is to the right and time is downward. The figure contains the first 300 time-steps, and then time-steps 2000 to 2950.

The major dimensions of the system are:

- direct route from A to B: 1021 sites (full size of plot)
- cars leave for the alternative route at (a)
- cars coming from C enter at (b) and leave again at (c)
- the bottleneck ($v_{max} = 1$) is at (d).

20% of the A-B vehicles are preselected to leave at the junction for the longer route, as can be seen in the picture by a change of the gray shading at (a). The entry-point of the C-D (b) vehicles is marked by the permanent existence of a disturbance, which is very often connected to other disturbances which travel "backwards" through the system.

The point of exit for the C-D (c) vehicles is covered by dense traffic most of the time, but it may be seen near the top right of the figure as a change in gray shading and as a sudden stop of some trajectories.

The bottleneck (d) is visible at the very right edge of the figure, where the trajectories of the vehicles are diagonally pointing downwards to the right.

The striking feature of this picture is the graphic illustration of the highly dynamic and nonlinear structure of traffic patterns. Vehicles do *not* wait orderly in front of the bottleneck, but instead self-organize into backwards moving jam waves. If one of these waves reaches back into an area with higher density (in our case upstream, i.e. to the left, of (c)), then the survival probability of this jam wave suddenly becomes much larger, and it may move deeply back into the system. A single snap-shot of such a traffic situation (see time-step 210 in Fig. 9.4) cannot uncover the origin of such a wave (cf. the arguments of Chapters 5 and 7). The implications to traffic measurement and modeling are important.

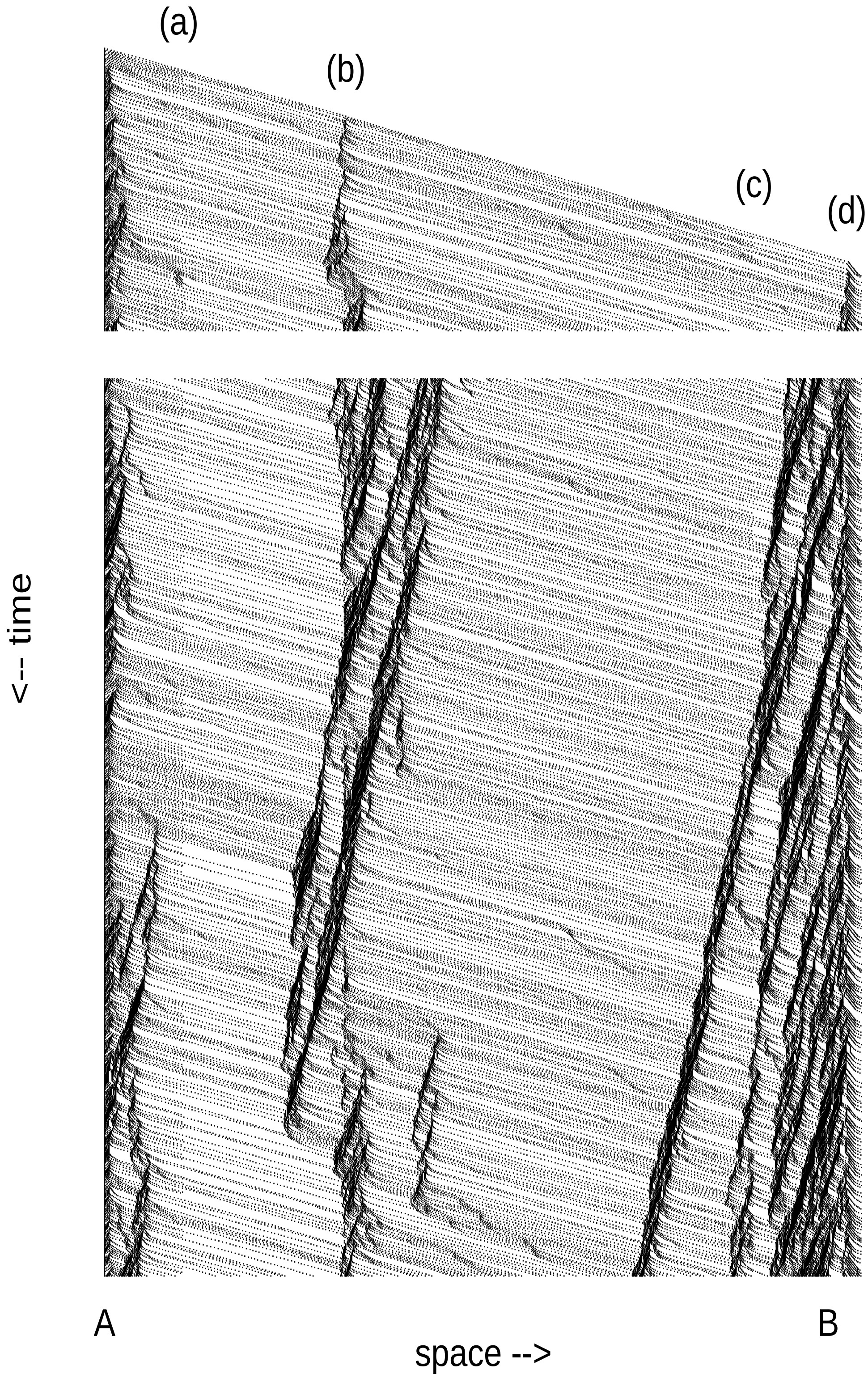

Figure 9.5: Space-time plot of the main segment (A-B) of the network. The cars are injected at the left. At (a), a change in gray indicates the junction where vehicles to the alternative route leave. (b) marks the position of the on-ramp for travel from C. At (c), another change in gray indicates the off-ramp to D. Very close to the right (d) is the bottleneck, together with jams emerging from this region and traveling backwards into the system.

Something similar is true for the region where the C-D traffic stream enters the main road. It is not a process where both traffic streams line up to wait until they can jointly proceed. Instead, it is often possible that the additional traffic enters into the main stream without causing a major disturbance right away. But the locally enhanced density is unstable and leads sooner or later to the initiation of a disturbance, which then travels backwards to the junction (and often beyond).

Both effects cannot be described by deterministic models, neither by microscopic [116, 117, 172] nor by fluid-dynamical [31, 133, 80] ones.

9.2.5 Congestion detection and congestion pricing

We now come back to a further determination of the tolling scheme. For simplicity, assume that the toll is based on some traffic observation of the last period (day). Let us further assume that each driver only drives this trip once in each period (day). Note that this is over-simplistic, and further investigations are needed to make it work for, e.g., workdays versus weekend.

Algorithmically, we proceed as follows: (i) The traffic microsimulation is executed for one period. Each driver updates her travel time information just after arrival at the destination. (ii) After all cars have reached their destinations, the toll is adapted according to the average value provided by the sensor. (iii) Each A-B driver makes her route choice. (iv) The next microsimulation period starts. — This results in a day-to-day evolution of the decision pattern [104]. The procedure is actually very similar to standard multi-period game-theory [5], except that we obtain the pay-off from the microsimulation and not from a predefined matrix.

A critical question remains: Where should one place the traffic sensor(s) for the determination of the toll? Placing it *inside* the bottleneck is not very useful, because traffic there is always at or below the "efficient" density (i.e. at or below the density corresponding to maximum throughput).

Intuitively, it would make sense to measure the length of the queue before of the bottleneck. However, as we showed in the last section, the dynamics of the traffic does not lead to the built-up of a regular queue but to a system of back-traveling jams instead, which makes this approach infeasible.

Since there is no obvious solution, it makes sense to start with something simple and local. It was therefore decided to use the temporal and spatial average density on the segment upstream of the bottleneck, i.e. between the exit to D and the bottleneck. A local as opposed to a global strategy is for example discussed by Faieta and Huberman in the context of traffic light control [40].

Then, the next question is, which should be the target density for the control algorithm? When one is measuring traffic upstream of a bottleneck, then stationary traffic can never reach maximum throughput: Either traffic operates at densities corresponding to flow rates lower than the bottleneck capacity, or dense traffic builds up. Traffic can only "use" the part of the fundamental diagram which is below the capacity of the bottleneck; in consequence, densities are either far below or far above the ones corresponding to maximum throughput [56, 132, 69, 172].

However, having some knowledge about the bottleneck is not really helpful: In a more complicated traffic network, it may be the case that further downstream of one bottleneck there is another one, which has even lower capacity. Or the bottleneck may be the on-ramp to a crowded major road: Here the performance of the bottleneck depends on the time-dependent and fluctuating load on the main road.

We therefore again follow a simplistic and completely local approach, which will nevertheless prove to be quite effective. Assume that the toll is operated by a local "toll agent", who does not have any global knowledge. However, she knows the fundamental diagram (flow as function of density) of *her* sensor area. If she wants to keep the traffic at maximum flow, she has to keep the density in the correct range, i.e. near $\rho = 0.08$. We implement this by the following rules:

---

if ($\rho < 0.06$) **then**
  $toll = toll - delta$
else if ($\rho > 0.10$) **then**
  $toll = toll + delta$
endif ,

---

where *delta* is an external parameter.

According to our arguments above, is not obvious that this approach will produce meaningful decision behavior: The toll agent tries to keep the traffic at a density regime which is dynamically impossible because of the bottleneck downstream. It is not clear, a-priori, what effect this will have, and it was one of our main points of interest how this control mechanism would work.

9.3 Simulation results: How to play traffic games

9.3.1 Technical set-up

In the following, we describe one particular simulation run in more detail. We used a network of overall size 1962 sites, composed of the following parts:

- direct route from A to B: 241 sites (smaller than for Fig. 9.5 to reduce computational demand)
- alternate route from A to B: 1570 sites (*much* longer than direct)
- connection from C to main route: 103 sites
- connection from main route to D: 48 sites
- length of the section shared by A-B and C-D-travelers: 101 sites
- thus, overall length from C to D: 252 sites
- length of bottleneck (with maximum velocity reduced to one): 10 sites
- position of the bottleneck: starts 20 sites before reaching B

The density ρ_{toll} for the update of the toll is measured between the junction where the vehicles heading for D leave the main route, and the start of the bottleneck.

We have $N_{AB} = 16000$ vehicles which want to travel from A to B, and the same amount $N_{CD} = 16000$ which want to travel from C to D. At each "day" of the simulation, they are lined up outside the simulated system in the same sequence; and they enter the system at their respective entry points as soon as the simulated traffic allows it (cf. Fig. 9.4).

When the vehicles enter the system, they already have decided on their travel plans, so they just execute these plans. The simulation runs until all vehicles have reached their respective destinations. Then the toll is updated and drivers decide on their route for the next day, as described above.

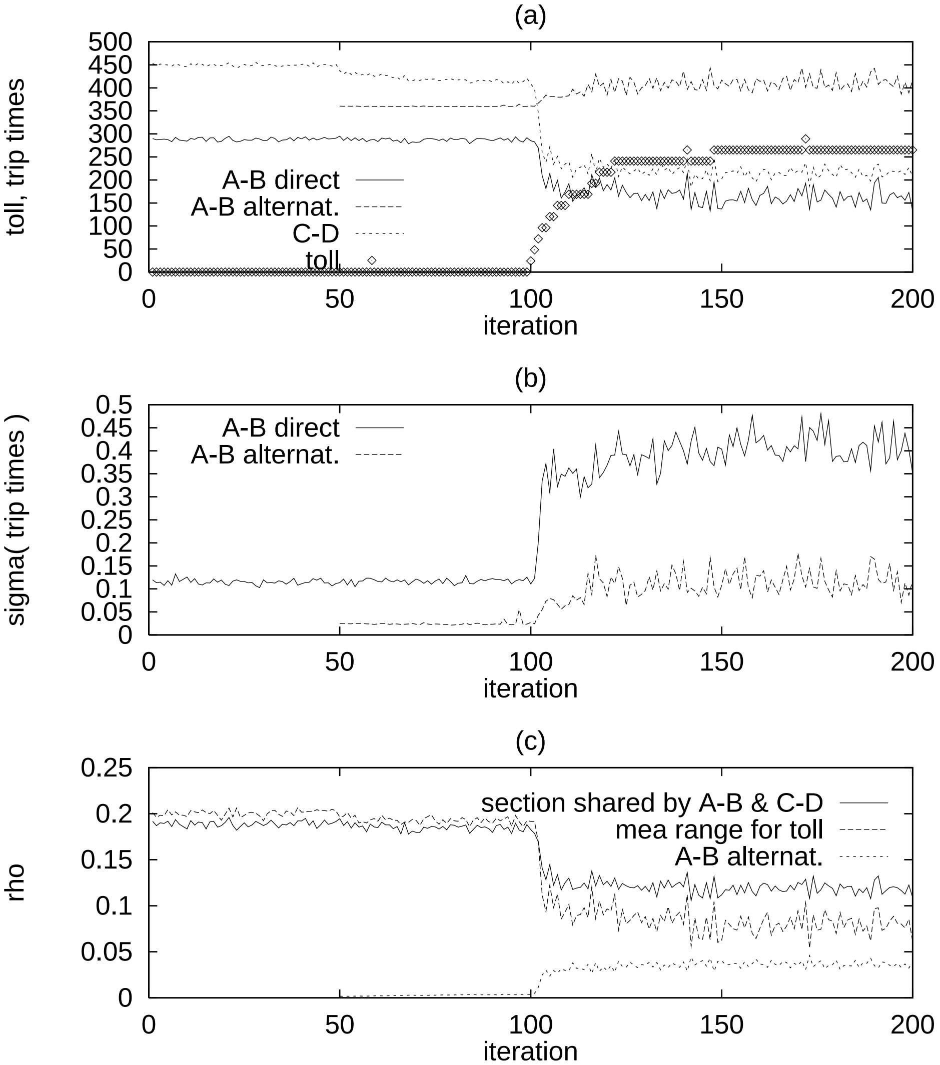

Figure 9.6: Simulation output for 200 iteration of the simple corridor network model. Time-steps 1-50: No adaption; 51-100: drivers can choose alternative route; 101-200: drivers can choose alternative route, and the toll adapts in order to keep the density at the specified level. *Top:* Average trip times for the direct and for the alternative route from A to B as well as for the route from C to D, and toll for the direct route from A to B. *Middle:* Vehicle-to-vehicle fluctuations of trip time for the direct and for the alternative route from A to B. *Bottom:* Densities on the segment shared by A-B-direct travelers and C-D-travelers, on the segment shortly before the bottleneck used for determination of the toll, and on the alternative route from A to B.

9.3.2 A simulation of 200 periods (days)

We describe 200 days of a simulation where the toll was kept at zero during the first 100 days, and in addition all A-B-travelers were forced to use the direct route during the first 50 days.

Fig. 9.6(a) shows results for the trip times and the adaptive toll, Fig. 9.6(b) the vehicle-to-vehicle variations of the trip time (as defined earlier), and Fig. 9.6(c) the day-averaged density, on selected road sections. These road sections are: (i) the section where the density for the toll adaption is measured, (ii) the section of the main road between the on-ramp from C and the off-ramp from D, and (iii) the alternative route.

Even when allowed (i.e. after day 50), not many of the A-B drivers use the new option of the alternative route. This is to be expected, since it is more than six times longer than the direct one. In consequence, travel times and fluctuations do not change much.

After day 100, the adaptive tolling starts and fairly quickly reaches a stationary value around 260 (Fig. 9.6(a)). As the "toll" line in Fig. 9.6(c) indicates, this keeps indeed ρ_{toll} near the specified range between $\rho = 0.06$ and 0.10. In addition, the density on the main segment (used by both A-B and C-D travelers) drops to around 0.11, above, but close, to the density of maximum throughput.

Travel times for C-D and for A-B-direct travelers decrease (Fig. 9.6(a)); and the toll just offsets the time gain for use of the direct route: $time_{direct} + \alpha \cdot toll \approx time_{alternat.}$; remember that $\alpha = 1$.

Vehicle-to-vehicle fluctuations (Fig. 9.6(b)) for the use of the alternative road increase from ca. 2% to around 12%, and for the use of the direct road from ca. 11% to around 42%. Moreover, the *day-to-day* fluctuations also seem to go up in all measurements.

All this is in agreement with our intuition that traffic management can indeed make traffic more efficient, but may in addition lead to higher fluctuations and, as a consequence, lower predictability, since the system is driven closer to capacity and thus to the edge of chaos.

One should distinguish between two different kinds of fluctuations: Fluctuations due to the dynamics, and fluctuations due to the learning. The fluctuations in the latter might be due to the specifics of the chosen learning scheme, especially the lack of historic information beyond the last try. More realistic assumptions about the learning and en-route information are claimed to avoid that [6, 38]. However, the results for the vehicle-to-vehicle fluctuations (i.e. the σ as defined in the text) only depend on the fact that the traffic density is driven towards the critical value. A less fluctuating learning scheme should therefore even *increase* our values for σ.

It should be emphasized that the quantity of the vehicle-to-vehicle and therefore the prediction error as defined above can only be found in a *microscopic* model.

9.4 Summary and discussion

The predictability of travel times sharply decreases when the density increases above the density of maximum throughput (near-critical regime). Therefore, traffic management will not only make traffic flow more efficient, but will also drive large portions of the system towards the critical regime. The main reason for this is that the most efficient use of a traffic system takes place when all parts operate at densities near capacity flow. Systems designed for the management of traffic flows will reroute traffic from overcrowded roads to under-crowded ones, thus driving both closer to criticality.[1] Once traffic is near the critical region, further control inputs will have unpredictable consequences.

This implies that the approximation of deterministic, predictable traffic patterns will be less and less correct the more one approaches high performance of the traffic system. In consequence, traffic assignment methods based on relaxation to (time dependent) equilibrium would no longer be meaningful: The changes in the traffic patterns due to one relaxation step would get lost in the changes due to the inherently fluctuating dynamics, and the algorithm would never converge. An open question is in how far one can replace the equilibrium quantities by statistical averages (e.g. many Monte-Carlo runs); this is a topic of future research.

[1] Technically, it should be remembered that traffic at maximum flow is not exactly critical, but shows critical properties up to a certain cut-off, provided by the fluctuations at free driving, as discussed in Chapter 7.

One envisaged way [102] of pushing drivers towards system-optimal behavior is to give each driver only individual route guidance instead of complete traffic information. If one doubts that this will lead to high user acceptance, then congestion pricing seems to be the only alternative. Our simulation results support the idea that already locally operating agents can achieve this in an efficient way.

In the text, the case of tolling on a specific road segment upstream of "the" bottleneck is discussed. However, this demands prior knowledge about the system. Yet, one can imagine a completely local algorithm in the following way (see also [101]): Assume that *every* road segment in the system is operated by a simple economic agent. This agent wants to keep the operation of the segment as efficient as possible, and the only measure she has is to go up or down with the toll. The agent knows the performance characteristics (i.e. throughput q as a function of density ρ) of her segment, and from this she obtains the density which corresponds to maximum flow and therefore to maximum road performance. The agent then tries to keep the density on her segment at this particular density, increasing the toll when the density becomes too high, and else decreasing it. In a real network, we would expect that the toll for most segments turns out to be zero.

This tolling scheme gives the impression that every agent locally drives her segment towards criticality (= maximum flow), but the situation is more complicated. In most cases, it is not the traffic *inside* bottlenecks which is tolled, but the overcrowded segments *upstream of* the bottlenecks. But because of the bottleneck, these upstream segments usually cannot operate at maximum throughput: As soon as the incoming flow is more than the bottleneck capacity, dense traffic builds up, and the segment switches from operation far below to far above the critical point (see above). Nevertheless, our results show that this still leads to having more parts of the network near criticality, as a result of collective effects.

In an economic context, we therefore have a local aiming for high performance, which happens to coincide with criticality. But even though the criticality very often cannot be reached locally by this mechanism, it drives the *global* system closer towards criticality: Local maximization of efficiency leads to global criticality [8].

Or in short: The fact that, in a complex system, high performance often has the downside of high variability seems also to be true in transportation systems.