Simulation zweispurigen Verkehrsflusses auf der Basis zellularer Automaten

Diplomarbeit im Fachbereich Physik
vorgelegt von

Marcus Rickert[1]

aus Bergisch Gladbach

Referent: Prof. Dr. Achim Bachem
Koreferent: Priv. Doz. Dr. Michael Schreckenberg

Mathematisch–Naturwissenschaftliche–Fakultät
Universität zu Köln

2. Mai 1994

[1]e-mail: mr@zpr.uni-koeln.de

Inhaltsverzeichnis

Ich versichere, daß ich die vorliegende Arbeit selbständig verfaßt und nur die angegebenen Hilfsmittel verwendet habe.

Marcus Rickert, April 1994

Danksagung

An dieser Stelle möchte ich Prof. Dr. Achim Bachem und Priv. Doz. Dr. Michael Schreckenberg danken, unter deren Obhut diese Diplomarbeit entstanden ist.

Grundlegenden Anteil am Gelingen hat mein Betreuer Kai Nagel, der trotz seines Aufenthaltes in den U.S.A. immer für mich ansprechbar war und mir mit Rat und Tat beiseite gestanden ist.

In Köln hat diese Aufgabe Thomas Pfenning übernommen, durch dessen technisch fundiertes und breit gefächertes Wissen ich Zugriff auf die neuesten Methoden und Software–Produkte hatte.

Für die Benutzung des Parallelrechners Parsytec GCel–3 im Zentrum für Paralleles Rechnen in Köln bedanke ich mich nochmals bei Herrn Bachem, für die Benutzung des Paragon XP/S–5 und XP/S–10 in Jülich bei der KFA–Jülich.

Desweiteren möchte ich denjenigen danken, die in Köln und Jülich die Rechnersysteme warten. Zu nennen sind da Jutta Docter, Stefan Esser, Wolfgang Form, Michael Greven, Renate Knecht, Rainer Kowalik, Harald Mika, W. Vermoehlen und Michael Wottawa sowie alle anderen Systembetreuer aus dem Rechenzentrum Köln und dem Rechenzentrum ZAM in Jülich, deren Namen ich leider nicht kenne.

Dank gebührt ebenso Steen Rasmussen, der mich überzeugt hat, daß diese Arbeit doch etwas mit Physik zu tun hat.

Für die Durchsicht meines Scripts und für die daraus resultierenden Anregungen bedanke ich mich bei meinem Vater, Anja Hamacher, Michael Kramer und Thomas Renard. Besonders mein Vater hat versucht, in meiner Arbeit unverständliche Formulierungen und Begriffe auszumerzen. Dadurch sollten weite Abschnitte auch für den Laien verständlich werden.

Ich danke den Mitarbeitern der PROFI.S, die mich in den Monaten der Niederschrift mit Arbeit weitgehend verschont haben.

Meinen Eltern danke ich dafür, daß sie mir das Studium und damit diesen Abschluß ermöglicht haben.

Kapitel 1

Einleitung

1.1 Vorwort

In den letzten beiden Jahrzehnten ist die Verkehrsdichte auf bundesdeutschen Straßen im allgemeinen und auf Autobahnen im besonderen drastisch[1] angestiegen. Daran konnte auch der fortwährende Ausbau des Autobahnnetzes nichts ändern. Wartezeiten in Staus verursachen inzwischen der Volkswirtschaft Verluste in Milliardenhöhe sowie eine Reduzierung der Freizeit und damit eine Verminderung der Lebensqualität von Millionen Autofahrern. Doch der zunehmende Verkehr ist auch ein ökologisches Problem. Durch Versäumnisse der Automobilindustrie, Motoren mit umweltschonender Schubabschaltung zu forcieren, wird besonders bei Verweilzeiten im Stau die Umwelt mit schädlichen Abgasen belastet. Die geringe Akzeptanz von umweltfreundlichem Fahrverhalten durch Abschalten des Motors bzw. durch Vermeidung von staugefährdeten Gebieten und Zeiträumen tut ihr Übriges, um die Situation zu verschärfen.

Ein Ziel der Verkehrspolitik in den nächsten Jahren muß sein, die Engpässe nicht durch Ausbau des Straßennetzes zu lindern, sondern vielmehr die vorhandene Infrastruktur durch zunehmende Verkehrsbeeinflussung optimal auszulasten und zusätzlich das Verkehrsmittel Auto als Bestandteil in ein umfassendes Verkehrskonzept einzugliedern, das alle Möglichkeiten des Individual– und öffentlichen Verkehrs umfaßt.

Mit zunehmender Leistungsfähigkeit der Computer wurde versucht, die Problematik der Verkehrsnetzplanung in rechnergestützten Simulationen naturgetreu zu modellieren, um damit ein Instrumentarium an der Hand zu haben, im Planungsstadium verschiedene Lösungen für staugefährdete Gebiete zu vergleichen. Dadurch kann kostengünstig die optimale Strategie ausgewählt werden. Dies ist jedoch die unbedeutendere Möglichkeit, mit der der Computer den Verkehrsplanern helfen kann, das Verkehrs–Chaos auf unseren Straßen zu vermeiden. Als viel interessanter ist die Verkehrsbeeinflussung anzusehen. Dabei soll bundesweit an allen Autobahnauffahrten, –dreiecken und –kreuzen sowie an einer ausreichenden Anzahl von Zwischenpunkten das Verkehrsaufkommen ständig *online* erfaßt und an Zentralrechner weitergeleitet werden. Dort wird versucht, durch Simulation die Verkehrsverteilung für die nächsten Minuten vorauszuberechnen und die kritischen Punkte und Strecken im Netz zu ermitteln, an denen innerhalb dieses Planungshorizontes Staus entstehen könnten. Werden Schwachstellen entdeckt, so kann ohne Zeitverzögerung eine Umleitung des Verkehrs

[1] Die Beförderungsleistung durch PKWs stieg im Zeitraum 1976 bis 1991 von 415 auf 703 Milliarden Personenkilometern. Für das Jahr 2010 wird ein Aufkommen von 838 Milliarden Personenkilometern geschätzt ([12] und [13]).

durch variable Verkehrszeichen eingeleitet werden. Genau nach diesem Prinzip funktioniert ein Pilotprojekt im Rhein-Main-Gebiet.

Bei genauerer Prüfung der Problematik stellt man fest, daß das Stauproblem oft nicht *lokal* zu meistern ist, sondern *regionale* Lösungen fordert. Sobald sich das betrachtete Gebiet vergrößert, steigt jedoch proportional die Länge der Strecken und die Anzahl der Anschlußstellen, Dreiecke und Kreuze, die von der Simulation zu bearbeiten sind. Im Augenblick sind herkömmliche Computer und Simulationstechniken dadurch noch hoffnungslos überfordert. In der vorliegenden Arbeit wird nun gezeigt, daß durch Anwendung von *Parallel–Computern* und *zellularen Automaten* reale Problemgrößen in angemessener Zeit zu simulieren sind. Parallel–Computer haben heute eine bis zu 1000–fach größere Rechenleistung als normale Computer und werden deshalb als das vielversprechendste Hardware–Konzept für die nächsten Jahrzehnte angesehen. Zellulare Automaten werden in der Physik, aber auch in anderen wissenschaftlichen Bereichen wie zum Beispiel der Genetik als Modell verwendet, um mit einfachen Regeln Abläufe schnell und wirklichkeitsnah zu simulieren.

Neben einer potentiellen Anwendung auf den realen Verkehr ist aber auch die Verwendung des Modells in der theoretischen Physik als Weiterentwicklung des eindimensionalen zellularen Automaten zu sehen. Grundsätzliche Untersuchungen können zeigen, ob Netze intrinsische Schwachstellen haben und wie die zeitliche Entwicklung des Verkehrsflusses innerhalb von Netzen abläuft.

1.2 Aufbau der Arbeit

Im zweiten Kapitel der Arbeit werden zunächst die Grundlagen des zellularen Automaten für einspurigen Verkehr erläutert. Sämtliche Begriffe und Meßgrößen, die im weiteren verwendet werden, sind dort erklärt. Es folgt im dritten Kapitel die Erweiterung des einspurigen Modells auf zwei Spuren und die Untersuchung der dadurch neu auftretenden Phänomene. Der eigentliche Kern der Arbeit liegt im vierten Kapitel, in dem das Simulationsmodell *PAMINA* vorgestellt wird, das die einspurigen und zweispurigen Modelle der vorangehenden Kapitel durch Definition von Schnittstellen an Autobahnkreuzen und –dreiecken zu einem Netzmodell erweitert. Das fünfte Kapitel dient der Präsentation der Simulationsergebnisse, die mit Hilfe des Netzmodells erarbeitet wurden. Im sechsten Kapitel werden Einzelheiten zur Implementation aller erwähnten Modelle erläutert. Dabei wird jeweils auf die Abhängigkeiten von der zur Verfügung stehenden Hardware eingegangen. Im siebten und letzten Kapitel werden die Ergebnisse der Arbeit diskutiert und ein Ausblick auf mögliche zukünftige Aktivitäten in diesem Bereich gegeben.

Kapitel 2

Grundlagen

Wegen der Ähnlichkeit zum Verhalten von Flüssigkeiten wird schon seit den 50er Jahren versucht, den Verkehrsfluß als hydrodynamisches Modell zu formulieren [16]. Dabei wird der Übergang vom frei fließenden zum zähflüssigen Verkehr mit dem Übergang von laminarer zur turbulenten Strömung verglichen. Trotz der Gemeinsamkeiten treten beim realen Verkehr Phänomene auf, die nur schwer oder gar nicht auf ein kontinuierliches Modell zu übertragen sind. Dazu gehören zum Beispiel das Auftreten von verschiedenen Fahrzeugtypen oder Überholvorgänge.

Mit Hilfe der Repräsentation von Fahrzeugen durch Einzelbits wurde der erste Schritt getan, die Problematik als zellulares Automatenmodell zu beschreiben [29]. Dabei können die Fahrzeuge im einfachsten Fall nur die Geschwindigkeiten 0 und 1 annehmen. Obwohl dieses Modell schon interessante Ergebnisse liefert [20] und auch analytisch exakt gelöst werden kann [28], so enthält es jedoch Symmetrien, die zu qualitativen Ergebnissen führen, welche nicht in Einklang mit dem realen Verkehr zu bringen sind. Als Haupteinwand ist zu nennen, daß der maximale Verkehrsfluß bei halber Belegung der Strecke eintritt und nicht bei ca. 10% bis 15% wie in der Realität. Dies ist auf die Äquivalenz von Beschleunigung und Verzögerung zurückzuführen, da nur eine Geschwindigkeit ungleich null existiert[1].

Durch Nagel und Schreckenberg [25] wurde das Verfahren modifiziert, indem statt zwei nun sechs verschiedene Geschwindigkeiten erlaubt sind. Der Vergleich von Meßdaten aus dem Straßenverkehr und Simulationen dieses Modells zeigte zum ersten Mal verblüffende qualitative Übereinstimmung. Dieses Verfahren war die Grundlage für die Untersuchungen dieser Arbeit. In den nächsten Abschnitten folgt eine Beschreibung des Begriffs des zellularen Automaten sowie des einspurigen Modells von Nagel und Schreckenberg.

2.1 Zellulare Automaten

Die meisten in der Physik auftretenden Probleme sind über einen Satz von Differentialgleichungen definiert, die Teilchen und Felder miteinander verknüpfen. Im Normalfall ist durch die hohe Anzahl der Teilchen und die große Reichweite der Felder der zu lösende Satz von Gleichungen so umfangreich, daß heutige und zukünftige Rechnersysteme noch nicht einmal

[1] Die Symmetrie von Meßgrößen bezüglich der Dichte 0.5 (enspricht 50% Belegung) ist bei Maximalgeschwindigkeit 1 nicht weiter verwunderlich. Denn der normale Fluß der Fahrzeuge bei Dichte ϱ ist vergleichbar dem Fluß der *Löcher*, also der Fahrzeugabstände, bei Dichte $1-\varrho$. Dies ist als Analogie zur Elektronen– und Löcherleitung in der Festkörperphysik zu sehen.

in Ansätzen in der Lage sind, den Rechenaufwand zu bewältigen. Als Beispiel seien hier die Berechnungen der Molekülbewegungen in einem Kubikmeter Luft mit 10^{23} Teilchen und 10^{46} Wechselwirkungen genannt.

Zur Lösung muß das Problem sowohl in der räumlichen als auch in der zeitlichen Auflösung stark vergröbert werden. Dabei wird von einem kontinuierlichen Raum zu einem äquidistanten Raumgitter und von kontinuierlichen Feldgrößen zu einer beschränkten Anzahl an Zuständen, die jeder Gitterpunkt annehmen kann, übergegangen. Zusätzlich wird die Zeit durch einen für alle Gitterpunkte gültigen Zeittakt diskretisiert. Damit ist die Grundidee des zellularen Automaten beschrieben. Wolfram [31] hat die wichtigsten Definitionen sinngemäß wie folgt zusammengefaßt:

- *Zellulare Automaten sind diskret bezüglich Raum (Gitter), Zeit (Zeittakt) und Anzahl der Zustände eines Raumpunktes.* Dadurch kommen sie der internen Repräsentation von numerischen Größen in Computern sehr nahe. Rundungsfehler, die ein großes Problem bei numerischen Simulationen ausmachen, entfallen. Durch die geringe Anzahl von Zuständen sinkt zudem der erforderliche Speicherbedarf.

- *Alle Raumpunkte sind äquivalent bezüglich ihrer möglichen Zustände und der Regeln, die die Ermittlung des neuen Zustandes festlegen.* Dies erlaubt eine einfache Implementation des zellularen Automaten ohne Berücksichtigung von Sonderfällen[2] durch spezielle Eigenschaften mancher Gitterpunkte.

- *Die Regeln beziehen sich nur auf Gitterpunkte der nächsten Nachbarschaft und auf den unmittelbar letzten Zustand (Markov–Eigenschaft).* Die geringe Anzahl von Regeln trägt ebenfalls zur Erhöhung der Geschwindigkeit bei. Auch wird die Komplexität des Problems vereinfacht. Bei Vielteilchen–Problemen wächst in der exakten Lösung die Anzahl der Gleichungen oft quadratisch mit der Anzahl der Teilchen. Bei Einschränkung auf eine konstant große Nachbarschaft steigt die Anzahl jedoch linear, wodurch die Berechnung von großen Systemen erst möglich wird.

- *Die Regeln sind deterministisch.*

- *Die Berechnung der neuen Zustände erfolgt synchron für alle Gitterpunkte.* Diese Eigenschaft hat besonderen Einfluß bei der Verteilung des Gitters auf einen Parallel–Computer. Nur durch sie wird die zeitgleiche Berechnung auf getrennten Einheiten möglich.

Trotz der Einfachheit ihrer Regeln zeigen viele zellularen Automaten bei ausreichend großen Systemen nicht–triviales Verhalten. Modelle mit zellularen Automaten sind heute in vielen Bereichen Gegenstand der Forschung. Dazu zählen zum Beispiel die selbstorganisierende Kritikalität[3], das Oberflächenwachstum, Spinmodelle, neuronale Netzwerke, Wolkenbildung in der Meterologie, das Zellwachstum und die Seuchenausbreitung in der Immunologie sowie die Vererbungslehre in der Genetik. Ein Review der zellularen Automaten ist in [30] zu finden.

[2] Sonderfälle treten an den Rändern des Systems auf, falls keine periodischen Randbedingungen gewählt werden. Diese sind hier nicht gemeint.

[3] SOC Selforganized criticality

Beschleunigung (Regel $R1$)

$$v_i \leftarrow \min(v_i + 1, v_{max})$$

Abstand halten (Regel $R2$)

$$v_i \leftarrow \min(v, gap(succ))$$

Nichtdeterministische Verzögerung (Regel $R3$)

if $rand() < prob_d$ then $v_i \leftarrow \max(v_i - 1, 0)$

Update

Das Fahrzeug i wird um v_i Sites vorwärtsbewegt.

Tabelle 2.1: *Regeln des einspurigen Modells*

2.2 Einspuriges Modell

Das Modell für den Verkehrsfluß von Nagel und Schreckenberg [25] basiert auf einem eindimensionalen, linearen Gitter der Länge L mit Gitterelementen x_i, den sogenannten *Sites*, die einen von sieben Zuständen einnehmen können: Entweder das Site ist leer oder von einem Fahrzeug mit Geschwindigkeit 0,1,2,3,4 oder $v_{max} = 5$ belegt. Die Fahrzeuge bewegen sich grundsätzlich nur in eine Richtung. Die Dynamik der Fahrzeuge ist so definiert, daß die Anzahl (bei periodischen Randbedingungen) erhalten bleibt, das heißt sie löschen sie sich gegenseitig nicht aus. Es ist eine Weiterentwicklung des Modells zur Beschreibung von Asymmetric–Exclusion–Prozessen, bei denen die Geschwindigkeiten auf 0 und 1 beschränkt sind.

2.2.1 Regelwerk

Mit der Definition

$$gap(pred) \quad := \quad \text{Anzahl freier Sites zum Vorgänger}^4$$

werden für alle Fahrzeuge die Geschwindigkeiten v_i und damit die neuen Positionen der Fahrzeuge über die Regeln in Tab. 2.1 aktualisiert. Dabei hat $gap(pred)$ den Wert 0, falls sich unmittelbar vor dem betrachteten Fahrzeug ein weiteres befindet. Die Größe v_{max} hat den Wert 5. Die Regel $R3$ bewirkt, daß die Geschwindigkeit eines Fahrzeugs mit der Wahrscheinlichkeit $prob_d$ um eine Einheit vermindert wird. In Diagrammen wird diese Größe auch mit *ProbDec* bezeichnet.

[3] *pred* steht für Englisch *predecessor* = Vorgänger. Diese Bezeichnung ist konträr zu der üblichen Nomenklatur in Listen, die in einer Richtung durchlaufen werden. Dort heißt *predecessor* dasjenige Element, das in der Reihenfolge vorher durchlaufen wird. Hier im Text ist die Bezeichnung als *zeitlich* anzusehen, da der *predecessor* eines Fahrzeugs einen gegebenen Ort vor diesem Fahrzeug erreicht.

2.2.2 Meßgrößen

Bei der quantitativen Betrachtung des Modells sind drei Meßgrößen wichtig. Die **Dichte** ϱ der Fahrzeuge auf dem Gitter (in Diagrammen auch mit *rho* bezeichnet) ergibt sich aus dem Verhältnis von belegten Sites N zu vorhandenen Sites L:

$$\varrho := \frac{N}{L}.$$

Die **Durchschnittsgeschwindigkeit** v_{mean} ist der arithmetische Mittelwert der Geschwindigkeiten aller im System vorhandenen Fahrzeuge:

$$v_{mean} := \frac{1}{N}\sum_{i=1}^{N} v_i$$

Der **Fluß** q ergibt sich aus dem Produkt von Dichte und Durchschnittsgeschwindigkeit

$$q := \varrho v_{mean} = \frac{1}{L}\sum_{i=1}^{N} v_i.$$

Er ist ein Maß für die Kapazität eines Straßenabschnitts. Dies wird deutlich bei Betrachtung der äquivalenten Formulierung für den lokalen Fluß durch einen Punkt eines Straßenabschnittes

$$q = \frac{N(\Delta t)}{\Delta t}$$

wobei $N(\Delta t)$ die Anzahl an Fahrzeugen ist, die in Δt Zeitschritten den Meßpunkt überquert haben.

Autofahrer sind an einer möglichst hohen Durchschnittsgeschwindigkeit interessiert, da dann ihre Fahrzeit kurz ist. Verkehrsplaner sind eher an einem hohen Fluß interessiert, um die Kapazität einer Straße (z.B. zur Rush–Hour) zu steigern.

2.2.3 Fundamentaldiagramm

Die elementare Messung für einen Straßenabschnitt besteht darin zu überprüfen, wie sich Durchschnittsgeschwindigkeit und Fluß in Abhängigkeit der Dichte verhalten. Zu diesem Zweck wählt man für das System *periodische Randbedingungen*[5] $x_1 = x_{L+1}$ und große Systemgrößen $L > 10000$. Die erhaltenen Diagramme (siehe Abb. 2.1) werden als *Fundamentaldiagramme* bezeichnet. Die Abhängigkeit zwischen Fluß und Dichte zeigt den für die Verkehrsproblematik typischen Verlauf: Der Fluß steigt bei geringer Belegung zuerst linear an, da alle Fahrzeuge ungehindert ihre Maximalgeschwindigkeit einhalten können. Für die Verzögerungswahrscheinlichkeit $prob_d = 0.5$ erreicht der Fluß bei der kritischen Dichte $\varrho_c = 0.086 \pm 0.002$ das Maximum $q_{max} = 0.318 \pm 0.001$ [22] (im Diagramm `rho_c` bzw. `q_max`). Bei höheren Dichten nimmt der Fluß durch zunehmende gegenseitige Behinderung immer mehr ab bis er bei $\varrho = 1$ den Wert 0 annimmt. In Abb. 2.2 ist eine schematisches Fundamentaldiagramm abgebildet. Man erkennt drei Bereiche. Der *Free–Flow–Bereich* erstreckt sich von Dichte 0 bis Dichte ϱ_a. Die Steigung der Strecke $\overline{AB}$ gibt die Free–Flow–Geschwindigkeit wieder. Es folgt ein Bereich fast konstanten Flusses von Dichte ϱ_a bis Dichte ϱ_b. Im dritten Bereich, genannt *Congested–Flow–Bereich*, nimmt der Fluß kontinuierlich ab. Die Steigung der Geraden $\overline{CD}$ gibt die Rückflußgeschwindigkeit von Stauwellen an. Im

[5] Das System wird zu einem Kreis geschlossen.

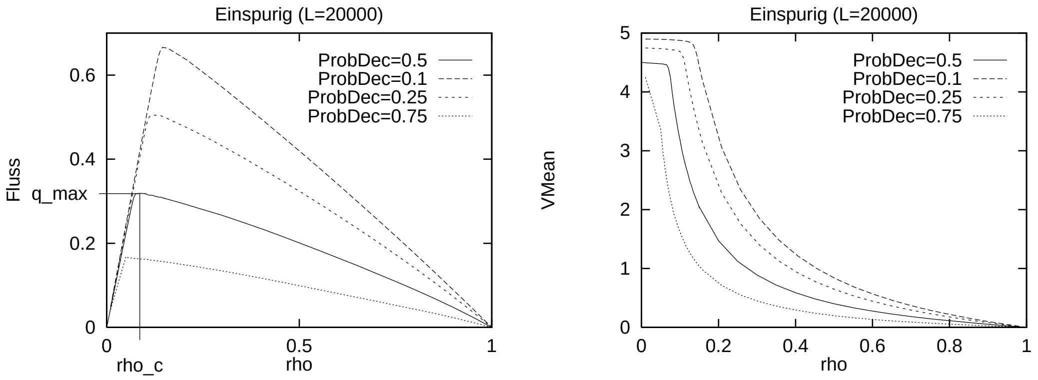

Abbildung 2.1: *Fundamentaldiagramme des einspurigen Verkehrsflusses*

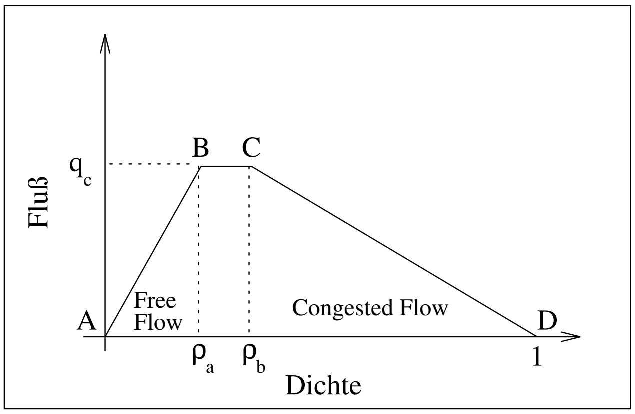

Abbildung 2.2: *Schematisches Fundamentaldiagramm*

Vergleich zu Abb. 2.1 stellt man fest, daß für die Ergebnisse der Simulation $\varrho_a = \varrho_b = \varrho_c$ gilt. Für das theoretische Modell lassen sich die Steigungen von $\overline{AB}$ in Punkt A und $\overline{CD}$ in Punkt D explizit angeben. Die erste ist gleich der Free–Flow–Geschwindigkeit

$$v_{free} = \lim_{\varrho \to 0+} v_{mean}(\varrho) \quad = \quad v_{max} - prob_d.$$

Für die zweite betrachtet man ein fast komplett gefülltes System, in dem nur noch wenige Sites frei sind. Praktisch alle Fahrzeuge haben die Geschwindigkeit null und nur die wenigen $L(1-\varrho)$ Fahrzeuge vor einem freien Site haben die Möglichkeit auf $v = 1$ zu beschleunigen. Sie tun dies mit einer Wahrscheinlichkeit $1 - prob_d$, so daß sich $q = (1-\varrho)(1-prob_d)$ als Zusammenhang zwischen Fluß und Dichte ergibt mit einer Steigung von $prob_d - 1$.

2.2.4 Einflüsse des Zufalls

Im Gegensatz zur ursprünglichen, deterministischen Definition des zellularen Automaten wird in diesem Modell der Zufall durch die Regel $R3$ mitberücksichtigt, die das Verhalten der Fahrzeuge entscheidend beeinflußt (*probabilistische zellulare Automaten*). Ohne diese Regel ordnen sich alle Fahrzeuge nach einer gewissen Relaxationsphase so an, daß ihre Geschwindigkeiten ein Muster bilden, das sich entgegen der Fahrrichtung fortbewegt [25]. Dieses bleibt dann für beliebig lange Zeit erhalten. Erst die Einführung des Zufalls ergibt durch lokale Fluktuationen ein sich ständig änderndes Muster.

Der Wert für die Verzögerungswahrscheinlichkeit kann variiert und damit eine Verlagerung des Maximums erreicht werden. In Abb. 2.1 sind die Fundamentaldiagramme für verschiedene Verzögerungswahrscheinlichkeiten $prob_d$ (ProbDec) dargestellt. In Anhang B.2 befindet sich eine kurze Erläuterung der verwendeten Zufallsgeneratoren.

2.2.5 Eichung

Durch Vergleich des berechneten Fundamentaldiagramms mit realistischen Diagrammen aus dem Straßenverkehr ist eine Eichung der Längen– und Zeitskala des Modells möglich. Dabei kann nach [25]

- die Durchschnittsgeschwindigkeit bei geringen Dichten v_{free},
- der maximale Fluß oder
- die Rücklaufgeschwindigkeit der Stauwellen

als Eichparameter verwendet werden. Aus allen drei Angaben ergibt sich als Zeitschritt ca. 1 [sec], als Länge eines Sites ca. 7.5 [m] und als Geschwindigkeitseinheit ca. 27 [km/h]. Demnach entspricht die Maximalgeschwindigkeit $v_{max} = 5$ einer realen Geschwindigkeit von 135 [km/h]. Die Free–Flow–Geschwindigkeit liegt bei $v_{free} = 122$ [km/h]. Nimmt man die Länge eines Fahrzeugs (PKW) mit 4 Metern an, so entspricht die Site-Länge dem Abstand zweier Fahrzeuge bei Verkehrsstillstand (mit 3.5 Metern Abstand von Stoßstange zu Stoßstange). Die Dichte 1 ist dann einem Stau gleichzusetzen, in dem Fahrzeug an Fahrzeug steht. In der Realität treten solch große Dichten jedoch nur selten und unter extremen Bedingungen auf (z.B. Rückstau vor Unfällen, bei denen mehr als eine Fahrspur wegfällt). Viel interessanter ist der Dichtebereich bis 0.3, weil sich dort der größte Teil der Dynamik abspielt. In den berechneten Fundamentaldiagrammen werden deswegen auch in den meisten Fällen die Dichten über 0.3 vernachlässigt.

2.2.6 Update–Verfahren

Durch die Regeln $R1$ bis $R3$ ist die Dynamik des Modells noch nicht erschöpfend beschrieben. Es fehlt die Angabe, in welcher Reihenfolge die Regeln auf die einzelnen Fahrzeuge des Systems angewandt werden. Man unterscheidet hier 3 Fälle:

Parallel Update: Hier wird die Konfiguration der Fahrzeuge im System vor Beginn des Updates zwischengespeichert. Anschließend werden für alle Fahrzeuge die neuen Geschwindigkeiten anhand dieser Konfiguration ermittelt. Da bei allen Fahrzeugen die gleiche Datengrundlage vorliegt, ist keines ausgezeichnet und das Update kann — trotz der sequentiellen Bearbeitung im Rechner — als *parallel* angesehen werden.

Random Sequential Update: Bei diesem Update wird das Fahrzeug, auf das die Regeln angewandt werden, durch eine Zufallszahl bestimmt. Anschließend wird es direkt weiterbewegt, so daß sich für das nächste Fahrzeug die Konfiguration geändert hat. Wie im ersten Fall ist hier kein Fahrzeug ausgezeichnet.

Sequential Update: Hier werden die Fahrzeuge mit entweder steigenden oder fallenden Indizes nacheinander bearbeitet. Bei periodischen Randbedingungen ergibt sich dabei das Problem, daß zwar im Prinzip kein erster bzw. letzter Index vorhanden ist, durch die Implementation jedoch im Rechner ein Fahrzeug ausgezeichnet ist, das als erstes bzw. letztes im Speicher abgelegt ist. Wegen dieses Schwachpunktes ist dieses Update–Verfahren bisher kaum untersucht worden.

2.2.7 Lösungen

Wie in der Einleitung erwähnt, entwickeln zellulare Automaten trotzt ihrer einfachen Regeln eine bisweilen hochkomplizierte Dynamik, die sich in den meisten Fällen einer analytischen Behandlung entzieht. Für den Sonderfall $v_{max} = 1$ kann jedoch für das Verkehrsmodell eine analytische Lösung angegeben werden. Für Maximalgeschwindigkeiten über 1 gibt es zumindest theoretische Näherungslösungen.

Fall $v_{max} = 1$

Für deterministisches Random Sequential Update mit $prob_d = 0$ ergibt sich die Durchschnittsgeschwindigkeit aus der Wahrscheinlichkeit, daß ein Fahrzeug ein freies Site vor sich hat

$$v_{mean}(\varrho) = 1 - \varrho$$

und damit der Fluß zu:

$$q_{rs}(\varrho) = \varrho(1 - \varrho). \tag{2.1}$$

Der maximale Fluß von $q_{max} = 0.25$ wird daher bei der kritischen Dichte $\varrho = 0.5$ erreicht.

Beim nichtdeterministischen (probabilistischen) Random Sequential Update konnten Nagel und Schreckenberg [25] zeigen, daß die Verzögerungswahrscheinlichkeit nur die Zeitachse skaliert und deshalb im Gleichgewicht ebenfalls Lösung (2.1) mit Korrekturfaktor $\overline{p} := 1 - prob_d$ gilt:

$$q_{rsp}(\varrho) = \overline{p}\varrho(1 - \varrho). \tag{2.2}$$

Beim nichtdeterministischen Parallel Update haben Schadschneider und Schreckenberg [28] gezeigt, daß sich die exakte Lösung aus der Molekularfeld–Näherung mit zusätzlicher Berücksichtigung von Nächsten–Nachbarn–Beziehungen[6] ergibt. Man erhält für den Fluß:

$$q_p(\varrho) = \frac{1 - \sqrt{1 - 4\overline{p}\varrho(1 - \varrho)}}{2}.$$

Dieser ist höher als im Random–Sequential–Update, was mit größeren Korrelationen erklärt wird, die sich in gegenseitiger Abstoßung zwischen Fahrzeugen äußern.

Fall $v_{max} > 1$

Für $v_{max} > 1$ gibt es bisher keine exakten Lösungen. Es wurden jedoch die Molekularfeld–Näherungen für Cluster–Größen bis $n = 5$ berechnet, die von den Simulationen weniger als ein Prozent abweichen [28].

2.2.8 Phänomene

Das einspurige Modell wurde von mehreren Gruppen eingehend untersucht. Dabei wurden verschiedene Phänomene entdeckt. Sie werden an dieser Stelle nur kurz erwähnt, da

[6] Cluster aus zwei Sites

in Kapitel 3 jeweils eine kurze Zusammenfassung bei Vorstellung der Ergebnisse für den entsprechenden zweispurigen Fall folgt.

In [22] wurden Systeme mit *offenen Randbedingungen* betrachtet, wobei am linken äußeren Ende Fahrzeuge der Geschwindigkeit 0 eingefügt (sobald dieser Site frei wird) und am rechten äußeren Ende auf einer Breite von v_{max} ankommende Fahzeuge gelöscht wurden. Bei Messung des Flusses, der sich im Gleichgewicht einstellt, wurde festgestellt, daß nicht der Maximalfluß der periodischen Randbedingungen erreicht wird. In [3] werden die Ergebnisse für verschiedende Einlaß– und Auslaßbedingungen untersucht. In [22] wird der *Ausfluß am Stauende* gemessen. Dabei konnte der Maximalfluß der periodischen Randbedingungen reproduziert werden, sobald die Anfangsdichte des Staubereiches oberhalb der kritischen Dichte ϱ_c lag.

Der bereits in [21] kurz erwähnte *Finite–Size–Effekt* eines erhöhten Flußmaximums bei kleinen Systemen mit $L < 10000$ wurde in [15] genauer untersucht.

In [23] wird demonstriert, daß in einem geringfügig abgewandelten Modell ein interessantes Verhalten bei Betrachtung des *Bremsverhaltens* einer Fahrzeugkolonne auftritt, die einem Fahrzeug mit geringfügig verminderter Geschwindigkeit folgt.

In [5] wird die Kolonnenbildung von Fahrzeugen mit ballistischer Dynamik in einem verwandten kontinuierlichen Modell untersucht und asymptotisch exakte Lösungen für die Überlebenswahrscheinlichkeit einzelner Fahrzeuge und die Verteilung der Kolonnenmassen gefunden.

Kapitel 3

Zweispuriges Modell

Zweispurige Modelle sind in den letzten beiden Jahren von mehreren Gruppen untersucht worden. Schadschneider und Schreckenberg [28] erwähnen ein Modell mit Geschwindigkeiten $v_{max_1} = 1$ und $v_{max_2} = 2$, die in Dichten ϱ_1 und ϱ_2 vorliegen. Sie finden eine Verschiebung des Maximums zu größeren Dichten. Nagatani [20] hat ein deterministisches Zweispur–Modell mit $v_{max} = 1$ vorgestellt und stellt eine Abhängigkeit des Übergangs von Free–Flow zu Congested–Flow und dem Austausch der Fahrzeuge zwischen den Fahrbahnen her. Latour [15] untersuchte kürzlich verschiedene Regelwerke für den nichtdeterministischen zweispurigen Verkehr.

Das vorliegende zweispurige Modell ensteht aus dem einspurigen durch Betrachtung zweier einspuriger Fahrbahnen, für die Wechselwirkungen durch Austausch von Fahrzeugen definiert werden. Der Wechsel zwischen Fahrbahnen dient dabei als Simulation von Überholvorgängen, die im realen Verkehr von Bedeutung sind. Desweiteren wurde ein zusätzlicher Satz von asymmetrischen Regeln untersucht, der das Rechtsüberholverbot auf Autobahnen modelliert.

3.1 Wunschgeschwindigkeiten und Fahrzeugtypen

Im einspurigen Modell ist es üblich, nur eine Maximalgeschwindigkeit bei Fahrzeugen zu betrachten. Die Einführung von mehreren Geschwindigkeiten führt i.A. nur zur Ausbildung von Fahrzeugkolonnen[1]. Der maximale Fluß orientiert sich bei diesen Modellen immer an dem des Fahrzeugs mit der geringsten Geschwindigkeit. Einen Ausweg aus diesem Verhalten bietet zum Beispiel die Simulation von einspurigem Verkehr mit Gegenverkehr, wobei Überholen auf der Gegenspur erlaubt ist.

Beim zweispurigen Verkehr ist die Einführung von verschiedenen Maximalgeschwindigkeiten wegen der Möglichkeit von Überholvorgängen durchaus sinnvoll. Diese Maximalgeschwindigkeiten werden im weiteren als *Wunschgeschwindigkeiten* v_d bezeichnet[2]. Sie drücken demnach eher eine empfohlene Geschwindigkeit für eine Fahrzeuggruppe und deren Fahrer aus als deren tatsächliche Höchstgeschwindigkeit, die normalerweise beträchtlich höher liegt. Im folgenden wird als Maximalgeschwindigkeit v_{max} die maximale Wunschgeschwindigkeit aller Fahrzeugtypen bezeichnet, wobei der Wert wie vorher bei 5 liegt. Bei den meisten Simulationen wurden zwei Fahrzeugtypen gewählt: schnelle (PKWs repräsentierende) Fahrzeuge mit $v_d = v_{max} = 5$ und langsame (LKWs repräsentierende) Fahrzeuge

[1] in der englischsprachigen Literatur als *Platoon* bezeichnet
[2] der Index steht für englisch *desired*

mit $v_d = 3$. Dabei werden andere typische Eigenschaften eines LKWs wie zum Beispiel eine geringere Beschleunigung oder eine geringere Staudichte (durch die größere Fahrzeuglänge) vernachlässigt. Trotzdem zeigt schon ein einfacher, in Abschnitt 3.2.4 präsentierter asymmetrischer Regelsatz, daß damit realistisches Verhalten nachempfunden werden kann. Der Anteil der langsamen Fahrzeuge an der Gesamtanzahl wird im folgenden mit *RatioS* bezeichnet.

3.2 Regelwerk

3.2.1 Spurwechsel und Überholen

Die Regeln, die den Spurwechsel definieren, lassen sich in zwei Untergruppen zerlegen. Die einen drücken den *Wunsch* des Fahrzeugs[3] aus, die Spur zu wechseln, um seine eigene Geschwindigkeit zu erhöhen. Die anderen gewährleisten, daß andere Fahrzeuge auf der Spur, auf die gewechselt wird, (genannt *Zielspur*) nicht zu stark beeinflußt werden, stellen also eine *Erlaubnis* dar. In dieser Arbeit wurde nicht versucht, viele verschiedene Regelsätze vergleichend darzustellen, sondern möglichst einen zu finden, der die gewünschten Resultate liefert und später als Grundlage für die Netzsimulation in Kapitel 4 dienen kann. Eine Zusammenstellung von fünf Modellen befindet sich in [15].

Bei den Regeln, die die Erlaubnis betreffen, sollte im Vordergrund stehen, daß der nachfolgende Verkehr auf der Zielspur zumindest noch den *unmittelbar nächsten* Zeitschritt ausführen kann, bevor eine Änderung des Fahrverhaltens durch den erfolgten Spurwechsel notwendig wird. Daher wird immer auf einen Abstand von v_{max} zum nachfolgenden Fahrzeug auf der Zielspur geachtet.

Ein *Überholvorgang* besteht normalerweise aus zwei Spurwechseln, die relativ kurz hintereinander liegen und zwischen denen mindestens ein Fahrzeug passiert wurde. Dies ist jedoch eine sehr komplexe Regel, denn sie würde ein Gedächtnis des Fahrzeugs über seine alte Spur und über die Anzahl der passierten Fahrzeuge voraussetzen. Dies widerspräche aber der Idee des zellularen Automaten.

Demnach gibt es in diesem Modell keine echten Überholvorgänge. Ein Zurückwechseln auf die "alte" Spur ist nur die Folge der gleichen Regeln, die vorher gespiegelt den Wechsel auf die vermeintliche Überholspur ausgelöst haben.

3.2.2 Asymmetrie

Zusätzlich zu den symmetrischen Regeln wird ein asymmetrischer Regelsatz vorgestellt, der das Verhalten vieler Autofahrer auf bundesdeutschen[4] Autobahnen modelliert, sich mit einer größeren Wahrscheinlichkeit auf der linken Spur aufzuhalten in der Hoffnung, dort schneller voran zu kommen. Das wird dadurch erreicht, daß Fahrzeuge so lange links bleiben, bis sie zum einen ihre Wunschgeschwindigkeit erreicht haben und zum anderen in großen Abstand vor sich keine weiteren Fahrzeuge auszumachen sind. Desweiteren werden Fahrzeuge auf der rechten Spur abgebremst, sobald sie Gefahr laufen ein Fahrzeug auf der linken Spur zu überholen.

[3] diese Personifizierung sei erlaubt

[4] und bestimmt auch manch anderer Länder

3.2.3 Symmetrisches Modell

Für die Überprüfung der Positionen von Vorgänger und Nachfolger werden die Funktionen

$$\begin{aligned} gap(pred) &:= \text{Anzahl freier Sites zum Vorgänger, und} \\ gap(succ) &:= \text{Anzahl freier Sites zum Nachfolger}^5 \end{aligned}$$

definiert. Bei *pred* und *succ* können die Indizes l, r und s angefügt werden, um die linke, rechte oder eigene Spur zu spezifizieren. Zur Beschreibung der Regeln für den Spurwechsel ist zusätzlich die Definition der Geschwindigkeit v_{hope} notwendig. Sie stellt diejenige Geschwindigkeit dar, die das Fahrzeug hätte, wenn es auf seiner jetzigen Spur bliebe ohne Berücksichtigung des Abstandes zum Vorgänger ($R2$) und des Zufalls ($R3$):

$$v_{hope} \quad := \quad min(\quad v+1 \quad , \quad v_d \quad)$$

Im symmetrischen zweispurigen Modell wechselt ein Fahrzeug nach $\left\{ \begin{matrix} \text{links} \\ \text{rechts} \end{matrix} \right\}$, falls die Bedingungen in Tab. 3.1 gleichzeitig zutreffen.

Regel $S1$	v_{hope}	$>$	$gap(pred_s)$	Ist aktuelle Spur ungünstig?
Regel $S2$	v_{hope}	$\leq$	$\left\{ \begin{matrix} gap(pred_l) \\ gap(pred_r) \end{matrix} \right\}$	Ist andere Spur besser?
Regel $S3$	$\left\{ \begin{matrix} gap(succ_l) \\ gap(succ_r) \end{matrix} \right\}$	$>$	5	Genügend Abstand zum Nachfolger?

Tabelle 3.1: *Regeln des zweispurigen, symmetrischen Modells*

In Abb. 3.1 ist ein Fahrzeug mit seiner berechneten Geschwindigkeit v_{hope} dargestellt. Es kann im nächsten Zeitschritt diese Geschwindigkeit nicht halten, da das nächste Hindernis nur vier Sites Abstand hat (Regel $S1$). Die grau schraffierten Felder geben die Sites wieder, die für den Spurwechsel frei sein müssen (Regeln $S2$ und $S3$).

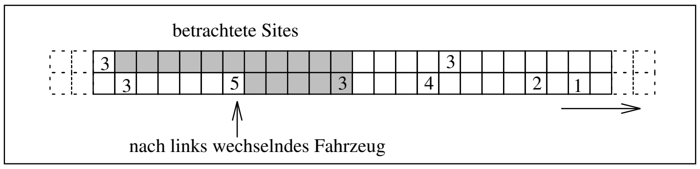

Abbildung 3.1: *Spurwechsel nach links*

Ein Update–Schritt ist in zwei Teilschritte geteilt: im ersten werden die Regeln der Wechselwirkungen überprüft und entsprechend Fahrzeuge ausgetauscht, jedoch ohne diese vorwärtszubewegen. Im zweiten Teilschritt werden die beiden Einzelfahrbahnen gemäß des einspurigen Modells unabhängig voneinander aktualisiert. Wenn man die Breite der Fahrbahn mit ca. 3.75 Metern (also ein halbes Site) annimmt, sind spurwechselnde Fahrzeuge genau genommen etwas schneller als spurtreue Fahrzeuge: $v_{eff} = \sqrt{(3.75/7.5)^2 + v^2} = \sqrt{0.25 + v^2}$.

[5] Englisch: *successor*

Dieser Effekt wurde jedoch nicht weiter untersucht, da für die Ermittlung der Größen Fluß und Durchschnittsgeschwindigkeit immer nur die Projektion der Geschwindigkeit parallel zur Fahrtrichtung verwendet wurde.

3.2.4 Asymmetrisches Modell

Im Rahmen dieser Arbeit wurden zwei asymmetrische Modelle untersucht. Das erste, im folgenden *einfach asymmetrisch* genannte, Modell wurde für Untersuchungen des Einflusses von langsamen Fahrzeugen (3.5) und für den Ausfluß am Stauende (3.7) verwendet, das andere, im folgenden *erweitert asymmetrisch* genannte, Modell, um den Effekt der Flußerhöhung durch Geschwindigkeitsbeschränkung (3.6) zu erzielen.

Das asymmetrische Modell unterscheidet sich durch zwei grundlegende Aspekte vom symmetrischen Modell:

- Der Regelsatz für den Wechsel der Fahrzeuge zwischen den Spuren ist unterschiedlich für die linke und die rechte Spur.
- Es wird ein Rechtsüberholverbot eingeführt, das die Geschwindigkeit der Fahrzeuge auf der rechten Spur in Abhängigkeit der Fahrzeuge auf der linken Spur beeinflußt.

Einfach asymmetrisches Modell

Die Regeln für den Wechsel nach links sind identisch mit denen des symmetrischen Modells ($A1_l = S1$, $A2_l = S2$ und $A3_l = S3$). Bei den Regeln für den Wechsel nach rechts unterscheidet sich nur Regel $S1$ von $A1_r$. Der Vollständigkeit halber sind dennoch alle aufgeführt. Ein Fahrzeug wechselt nach rechts, wenn die Regeln in Tab. 3.2 erfüllt sind.

Regel $A1_r$	$gap(pred_s)$	$>$	$2v_{hope}$	Großer Abstand zum Vorgänger?
Regel $A2_r$	v_{hope}	$\leq$	$gap(pred_r)$	Rechte Spur frei?
Regel $A3_r$	$gap(succ_r)$	$>$	5	Genügend Abstand zum Nachfolger?

Tabelle 3.2: *Regeln des zweispurigen, einfach asymmetrischen Modells*

Erweitert asymmetrisches Modell

Das erweitert asymetrische Modell unterscheidet sich vom einfachen Modell zum einen durch die Regel $A1_r^*$, die in Tab. 3.3 zu sehen ist. Zum anderen erfolgt die Berechnung der

Regel $A1_r^*$	v_{hope}	$=$	v_d	Wunschgeschwindigkeit erreicht und
	$gap(pred_s)$	$>$	$2v_d$	großer Abstand zum Vordermann?

Tabelle 3.3: *Ersatzregel des zweispurigen, erweitert asymmetrischen Modells*

Geschwindigkeit v_{hope} für Fahrzeuge auf rechten Spur in Abhängigkeit der linken Spur:

$$v_{hope,asym} = \min(\quad v_{hope,sym} \quad , \quad v_{left} \quad)$$

wobei v_{left} die Geschwindigkeit des Vorgängers auf der linken Spur ist, falls dieser weniger als $2v_{max}$ Sites entfernt ist. Die Regeln $A2_r$ und $A3_r$ des einfachen Modells bleiben erhalten.

Eigenschaften beider asymmetrischer Modelle

In Abb. 3.2 ist ein Fahrzeug mit seiner berechneten Geschwindigkeit v_{hope} dargestellt. Für einen Spurwechsel nach rechts müssen alle grau schraffierten Sites frei sein.

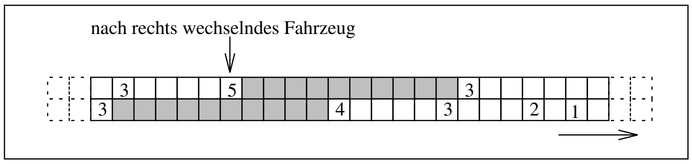

Abbildung 3.2: *Spurwechsel nach rechts im asymmetrischen Modell*

Im Gegensatz zum symmetrischen Modell werden im asymmetrischen Modell abwechselnd der linke und rechte Fahrstreifen aktualisiert, um das Rechtsüberholverbot zu modellieren. Im Algorithmus 1 sind die Schritte aufgelistet. Dabei bezeichnet L die Länge des Systems in Sites.

FOR Position L **DOWNTO** Position 1 **DO**
    BEGIN
    IF Linkes Site belegt **THEN**
        BEGIN
        Speichere Fahrzeuggeschwindigkeit $v_{left} := v$
        Aktualisiere (linkes) Fahrzeug
        END
    IF Rechtes Site belegt **THEN**
        BEGIN
        IF $gap(pred_l) < 2v_{max}$ **THEN**
            $v := v_{left}$
        Aktualisiere (rechtes) Fahrzeug
        END
    END

Algorithmus 1: Teilschritt im asymmetrischen Modell

In Abb. 3.3 ist ein Fahrzeug abgebildet, das beim Update aufgrund des Überholverbotes auf die Geschwindigkeit seines linken Vorgängers abgebremst wird.

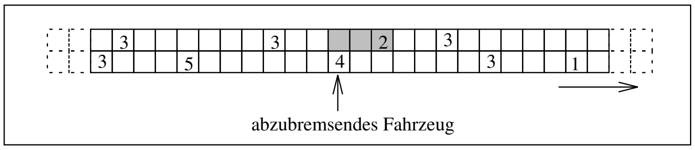

Abbildung 3.3: *Überholverbot im asymmetrischen Modell*

3.3 Fundamentaldiagramme

Bei der ersten Messung wurde ein Vergleich der Fundamentaldiagramme von einspurigem und zweispurigem Verkehr für einen Fahrzeugtyp mit $v_d = 5$ durchgeführt (siehe Abb. 3.4). Dabei ist zu beachten, daß der Fluß *pro Spur* abgetragen ist. Es fällt auf, daß im symme-

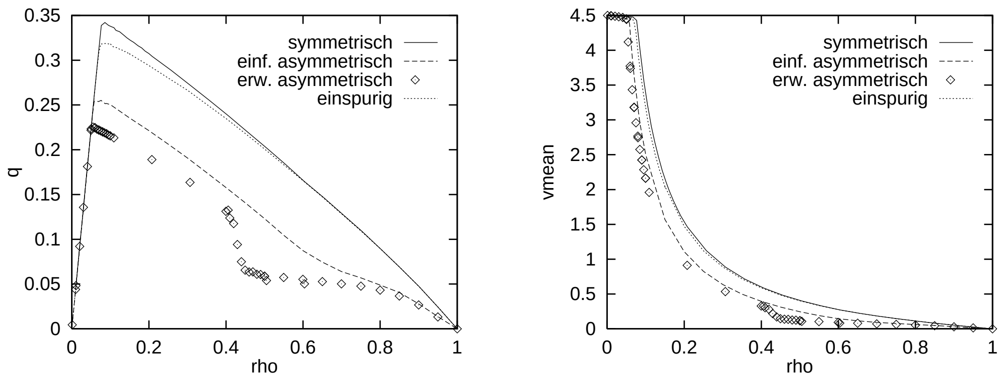

Abbildung 3.4: *Fundamentaldiagramme für zweispurigen Verkehr*

trischen Modell der Fluß tatsächlich höher ist als im einspurigen Fall. Das bedeutet, daß kleine lokale Dichteerhöhungen auf einer Spur schneller ausgeglichen werden, indem ein Teil der beteiligten Fahrzeuge auf die andere, noch freie Spur wechselt. Im asymmetrischen Fall jedoch kommt es zu einer erheblichen Flußreduzierung durch den umgekehrten Effekt: Gibt es auf der linken Spur eine Dichteerhöhung und eine Verminderung der Durchschnittsgeschwindigkeit, so kommt es automatisch zu einer Anpassung und damit Verminderung des Flusses auf der rechten Spur. Besonders bei Dichten ab 0.5 nimmt im erweiterten Modell der Fluß rapide ab. Auf diesen Effekt wird ausführlicher in Abschnitt 3.6 eingegangen.

Die Lage ϱ_c des Flußmaximums wird durch die unterschiedlichen Regeln geringfügig beeinflußt (siehe Tab. 3.4).

Regeln	ϱ_c	$q_{max}(\varrho_c)$
einspurig	0.086 ± 0.002	0.318 ± 0.001
symmetrisch	0.086 ± 0.002	0.341 ± 0.001
einfach asymmetrisch	0.076 ± 0.002	0.255 ± 0.004
erweitert asymmetrisch	0.055 ± 0.002	0.225 ± 0.002

Tabelle 3.4: *Vergleich der Maxima der Regelwerke*

3.4 Eichung

Die Eichung für den zweispurigen, symmetrischen Fall ist dem einspurigen Fall sehr ähnlich, da sich wie oben gesehen das Fundamentaldiagramm nur geringfügig verändert. Laut [4] können die in Tab. 3.5 aufgeführten Richtwerte für eine zweispurige Fahrbahn bei optimalen äußeren Bedingungen angenommen werden. Verwendet man v_{free} als Eichparameter

Name	Zeichen	Wert	Einheit
Flußmaximum	q_{max}	1800	[Fahrzeuge/h]
Free–Flow–Geschwindigkeit	v_{free}	115	[km/h]
Geschwindigkeit am Flußmaximum	$v_0 = v(q_{max})$	90	[km/h]

Tabelle 3.5: *Richtwerte für zweispurige Fahrbahnen bei optimalen Bedingungen*

und übernimmt die Site–Länge von 7.5 [m], so ergibt sich die Zeiteinheit zu 0.95 [s]. Der maximale Fluß beträgt dann 1297 [Fahrzeuge/h], ist also erheblich unter dem Richtwert. v_0 ergibt sich zu 102 [km/h]. Das Problem des zu geringen Flusses ist schon aus dem einspurigen Fall bekannt. Bei Messungen des dynamischen Fundamentaldiagramms (siehe 3.9) stellt sich heraus, daß der maximale Fluß bei kleinen Systemen tatsächlich größer sein kann als das Maximum des idealisierten unendlichen Systems. Deswegen ist die hier aufgezeigte Diskrepanz nicht so störend, wie vermutet werden könnte.

Benutzt man q_{max} als Eichparameter, so verringert sich die Zeiteinheit auf 0.68 [s]. v_0 und v_{free} erhöhen sich auf 158 [km/h] bzw. 179 [km/h]. Diese Werte sind für realistischen Straßenverkehr zu hoch.

3.5 Einfluß langsamer Fahrzeuge

Als nächstes wurde der Einfluß langsamer Fahrzeuge untersucht. Dazu wurden im symmetrischen und einfach asymmetrischen Fall jeweils ein gewisser Anteil an Fahrzeugen auf Wunschgeschwindigkeit $v_d = 3$ gesetzt. Der Rest der Fahrzeuge behielt die Wunschgeschwindigkeit $v_d = 5$ (siehe Abb. 3.5 und Abb. 3.6). Der Verlauf von q und v_{mean} ist

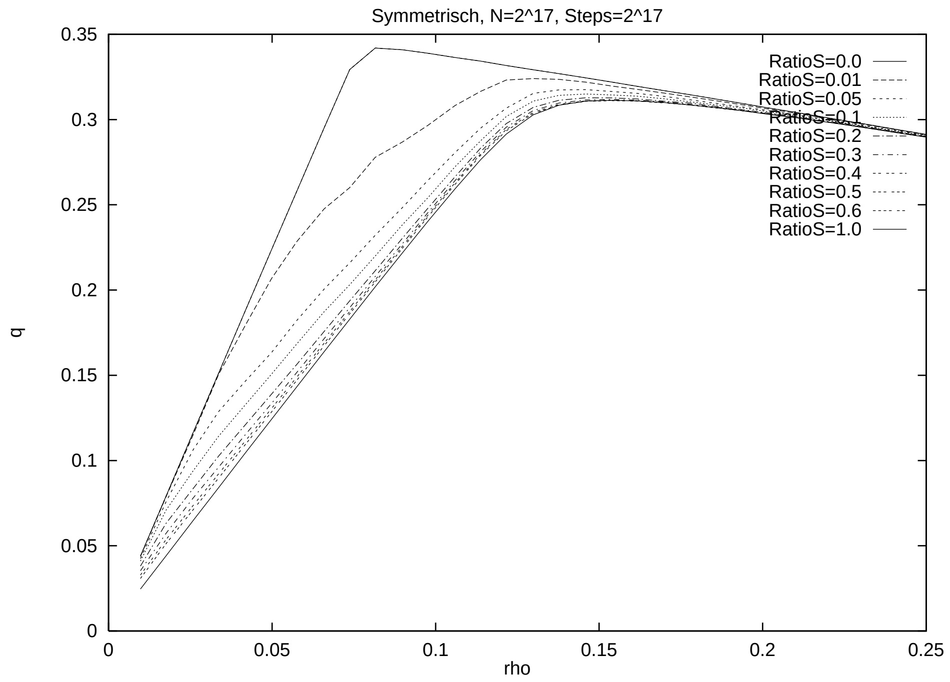

Abbildung 3.5: *Einfluß langsamer Fahrzeuge auf den Fluß im symmetrischen Modell*

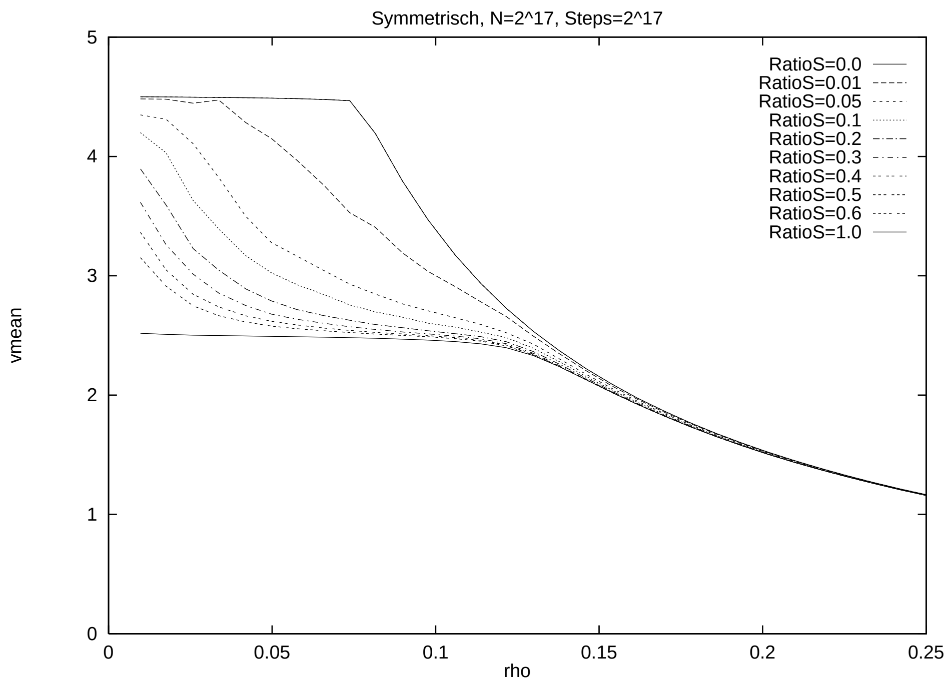

Abbildung 3.6: *Einfluß langsamer Fahrzeuge auf die Durchschnittsgeschwindigkeit im symmetrischen Modell*

überraschend. Für geringe Dichten erwartet man, daß sich schnelle und langsame Fahrzeuge kaum gegenseitig beeinflußen. Die Meßgrößen sollten sich daher aus dem arithmetischen Mittel der Kurven für $q(RatioS = 0.0)$ und $q(RatioS = 1.0)$ ergeben. Man sieht jedoch, daß das nicht der Fall ist. Für $RatioS = 0.01$ ist schon bei $\varrho = 0.03$, also weit unterhalb der kritischen Dichte ein Abknicken von v_{mean} zu erkennen. Für höhere $RatioS$ setzt der Abfall sogar noch früher ein. In Abb. 3.7 ist eine Übersicht über die Abhängigkeit der Free–Flow–Geschwindigkeit v_{free}, der Geschwindigkeit am Flußmaximum v_{ϱ_c} (hier mit $\varrho_c = 0.08$ angenommen) und Flußmaximum q vom Anteil langsamer Fahrzeuge zu sehen. $v_{free}(0.01)$

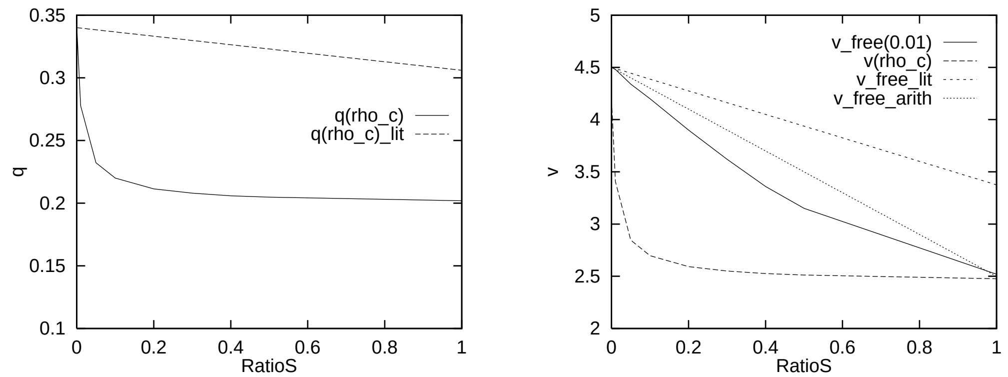

Abbildung 3.7: *Vergleich Einfluß langsamer Fahrzeuge*

gibt den simulierten Verlauf der Free–Flow–Geschwindigkeit wieder. Dem gegenübergestellt ist der aus [4] entnommene Verlauf $v\_free\_lit = v_{free}(1 - 0.25 RatioS)$, der für ein Steuerungssystem im Rhein–Main–Gebiet entwickelt wurde (siehe Tab. 3.6 nach [4]). Das arithmetische Mittel aus beiden Free–Flow–Geschwindigkeiten ist unter v_free_arith abgetragen. Auch die Geschwindigkeit am Flußmaximum fällt für kleine Dichten stark ab, bleibt aber ab $\varrho = 0.2$ konstant. Die Abhängigkeit des Flusses ist in $q(rho\_c)$ abgetragen. Man erkennt auch dort die zu starke Abhängigkeit von $RatioS$ gegenüber den Literaturwerten $q(rho\_c)\_lit$.

Parameter	Korrektur durch LKWs
q_{max}	$1 - 0.1\ RatioS$
v_{ϱ_c}	1
v_{free}	$1 - 0.25\ RatioS$

Tabelle 3.6: *Einflüsse von LKWs*

Die langsamen Fahrzeuge haben demnach einen nicht zu unterschätzenden Einfluß. Durch die Symmetrie der Regeln sind die langsamen Fahrzeuge auf beide Spuren gleich verteilt. Sie befinden sich mit einer gewissen Wahrscheinlichkeit direkt oder leicht versetzt nebeneinander. Dadurch wird aber ein Überholen durch die schnelleren Fahrzeuge unmöglich, so daß sich auch im zweidimensionalen Fall lange Kolonnen hinter den langsamen Fahrzeugen bilden. In Abb. 3.8 ist ein solcher Fall abgebildet. Die grau schraffierten Sites geben die Positionen an, auf denen der obere LKW (zusammen mit dem unteren) die Durchfahrt für nachkommende Fahrzeuge versperrt. Die Länge d_{mean} gibt den mittleren Abstand

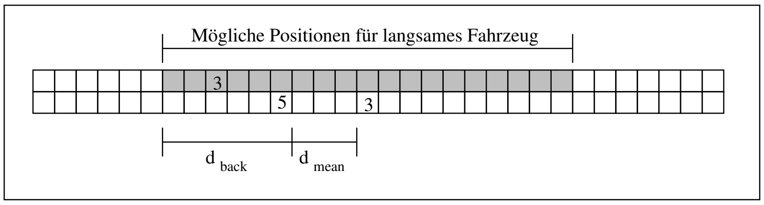

Abbildung 3.8: *Blockade durch langsame Fahrzeuge*

vom schnellen Fahrzeug zum langsamen an[6]. Zum Spurwechsel müssen zusätzlich d_{back} Sites frei sein, so daß sich insgesamt 9 Sites ergeben. Aus Symmetrie muß das langsame Fahrzeug den gleichen Abstand auch haben, wenn es sich *vor* dem unteren befindet, also insgesamt 18 Sites. Für ein Verhältnis von $RatioS = 0.05$ und einer Dichte von $\varrho = 0.03$ ergibt sich eine Wahrscheinlichkeit von $18 RatioS^2 \varrho^2 = 0.000405$, daß an einer Stelle die freie Durchfahrt blockiert ist. Bei $N = 2^{17} = 131072$ Sites ist das bereits für ca. 5 Stellen der Fall. Zwei langsame Fahrzeuge, die sich unmittelbar nebeneinander auf freier Strecke befinden, müssen sich mindestens 9 Sites voneinander entfernen. Dazu ist es notwendig, daß neun mal *hintereinander* das eine Fahrzeug verzögert wird, während das andere seine Geschwindigkeit beibehält. Demnach ergibt sich bei $prob_d = 0.5$ eine Wahrscheinlichkeit von $p = 0.5^9 \sim 0.002$ für die Freigabe der Strecke.

[6]im Durchschnitt ist $d_{mean} \sim 2.5 = v_{slow,free}$

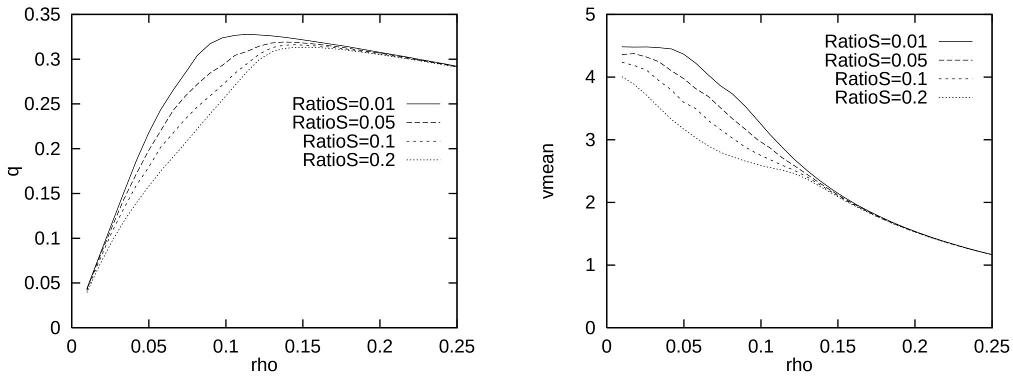

Abbildung 3.9: *Fundamentaldiagramme bei asymmetrischer Verzögerungswahrscheinlichkeit*

Der Fluß wird also durch wenige Paare von langsamen Fahrzeugen beträchtlich vermindert. Deshalb wurden Messungen mit geringfügig unterschiedlichen Verzögerungswahrscheinlichkeiten $prob_{d,l} = prob_d - \Delta prob_d$ für die linke und $prob_{d,r} = prob_d + \Delta prob_d$ für die rechte Spur durchgeführt. Das Resultat für $\Delta prob_d = 0.05$ ist in Abb. 3.9 zu sehen. Man erkennt, daß die Kurven von Fluß und Durchschnittsgeschwindigkeit zur Dichte für unterschiedliche *RatioS* einen klaren Abstand zueinander haben. Erst bei Dichte $\varrho = 0.15$ laufen sie wieder zusammen.

Im Gegensatz dazu ist im asymmetrischen Modell der Einfluß der langsamen Fahrzeuge nicht überproportional stark, wie Abb. 3.10 zeigt.

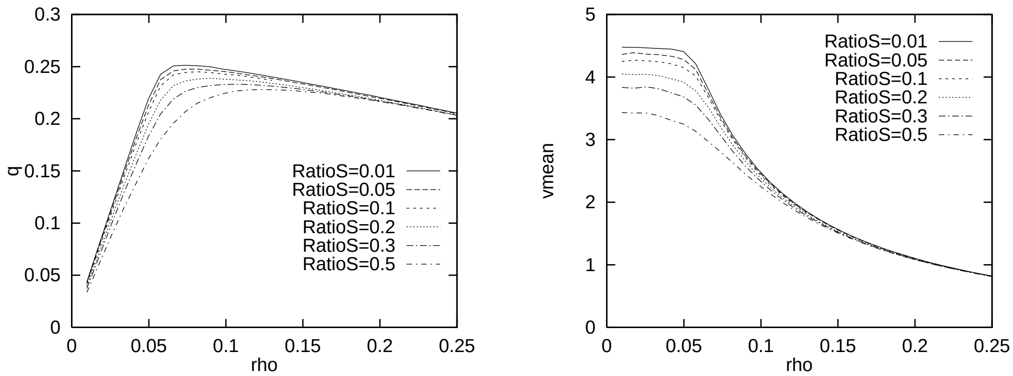

Abbildung 3.10: *Einfluß langsamer Fahrzeuge im einfach, asymmetrischen Modell*

3.6 Geschwindigkeitsbeschränkung

Nach Messungen von Zackor [33] kann durch Einführung einer Geschwindigkeitsbeschränkung der Fluß einer zweispurigen Straße erhöht werden. In Abb. 3.11 sind die Meßresultate für eine Geschwindigkeitsbeschränkung auf der Autobahnstrecke München—Salzburg zu sehen. Man erkennt eine deutliche Zunahme des Flusses für 80 [km/h] und 100 [km/h].

Diagramm Zackor

Abbildung 3.11: *Flußmessung bei verschiedenen Höchstgeschwindigkeiten*

Für 60 [km/h] ist der Maximalfluß wieder kleiner. Es war Ziel bei Aufstellung des einfach asymmetrischen Regelsatzes, diesen Effekt zu reproduzieren. Durch Änderung der Regel $A1_r^*$ gegenüber der symmetrischen Regel $S1$ verbleiben Fahrzeuge solange auf der linken Spur, bis sie ihre Wunschgeschwindigkeit erreicht und in absehbarer Entfernung keinen Vordermann sehen. Die Messung wurde für einen Anteil an langsamen Fahrzeugen von $RatioS = 0.15$ durchgeführt. Es wurde einmal gemessen ohne Beschränkung der Wunschgeschwindigkeit, einmal mit Beschränkung auf $v_{max} = 4$ und einmal mit Beschränkung auf $v_{max} = 3$. Im letzten Fall sind langsame und schnelle Fahrzeuge ununterscheidbar. Das Ergebnis ist in Abb. 3.12 zu sehen. Es zeigt sich eine Flußerhöhung für $v_{max} = 3$ und $v_{max} = 4$ gegenüber dem nicht beschränkten Fall. Außerdem sind die Maxima nach rechts verlagert. Damit ist der Effekt von Abb. 3.11 qualitativ reproduziert worden. Leider ist er quantitativ nicht so stark ausgeprägt. Der Effekt der Flußerhöhung basiert in diesem Modell auf der

Quelle	$v_{max}[km/h]$	$\Delta\varrho_c$ [%]	Δq [%]
Messung	100	40	15.0
Simulation	$\sim$ 108	43	2.7
Messung	80	50	21.0
Simulation	$\sim$ 81	100	2.3

Tabelle 3.7: *Vergleich von Simulationen*

gleichmäßigeren Auslastung der beiden Fahrspuren. Ohne Geschwindigkeitsbeschränkung sind die schnellen Fahrzeuge durch Regel $A1_r^*$ gezwungen auf der linken Spur zu bleiben, bis sich vor ihnen eine Lücke der Länge $2v_{hope}$ ergibt. Für kleine Dichten stellt das kein

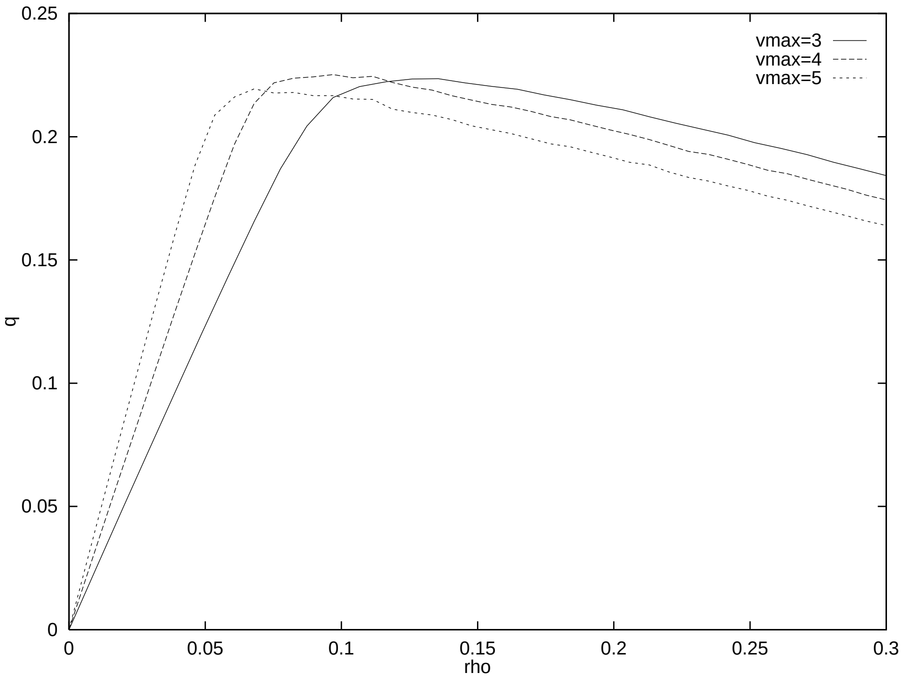

Abbildung 3.12: *Flußerhöhung durch Geschwindigkeitsbeschränkung im erweitert asymmetrischen Modell*

Problem dar, denn der durchschnittliche Abstand zweier Fahrzeuge liegt bis 0.09 oberhalb von $2v_{hope}$. Dann jedoch kommt es zu einem vermehrten Verbleiben auf der linken Spur: die rechte Spur verarmt an schnellen Fahrzeugen. Es verbleiben hauptsächlich die langsamen Fahrzeuge, die sowieso nur einen geringen Anteil zum Gesamtfluß beitragen. Durch die Herabsetzung der Wunschgeschwindigkeit werden die Spurwechsel nach rechts wieder wahrscheinlicher und es kommt zu einer homogenen Auslastung.

Es bleibt zu vermerken, daß nur das erweitert asymmetrische Modell diesen Effekt hervorrufen kann. Der Hauptunterschied zum einfachen Modell besteht darin, daß schon bei der Überprüfung, ob auf die linke Spur gewechselt werden soll, die spätere Geschwindigkeitsreduzierung durch das Überholverbot mittels v_{hope} berücksichtigt wird. Dadurch ist die Tendenz, nach links zu wechseln, noch größer als im einfachen Modell.

Bezüglich der Realität wird in [4] und [8] noch eine andere Ursache für die Erhöhung des Flusses genannt. Durch die Geschwindigkeitsbeschränkung kommt es nicht nur zu einem Cut–Off bei hohen Geschwindigkeiten, sondern langsame Fahrer werden darüberhinaus ermuntert ihre Geschwindigkeit (nach oben) der Beschränkung anzupassen. Sie fungiert also gleichzeitig als Geschwindigkeitsgebot. Die Folge ist eine Abnahme der Varianz der Geschwindigkeitsverteilung und damit auch Abnahme der Fluktuationen im Bereich kritischer Dichten, die als Ursache für Verkehrsstaus angesehen werden.

3.7 Ausfluß am Stauende

Nagel [22] hat festgestellt, daß sich der Ausfluß von Fahrzeugen aus einem Gebiet hoher Dichte $\varrho > \varrho_c$ auf das Flußmaximum des Systems mit periodischen Randbedingungen einstellt. Diese Messung wurde hier auch für den zweispurigen Verkehr durchgeführt. Es wurde folgendermaßen vorgegangen:

1. Ein zweispuriges System der Länge L wird komplett mit Fahrzeugen gefüllt $\varrho = 1$. Es wurden Anteile von langsamen Fahrzeugen $RatioS = 0.0, 0.05$ und 0.15 verwendet.
2. Die Simulation wird entweder mit symmetrischen oder asymmetrischen Regeln gestartet und die ersten $L/20$ Zeitschritte ohne Aufnahme von Statistik durchgeführt. Das rechte Ende des Systems ist offen in dem Sinne, daß alle Fahrzeuge, die sich in den letzten v_{max} Sites befinden, gelöscht werden.
3. Ab dem Zeitschritt $L/20$, werden alle Fahrzeuge, die das System am rechten Ende verlassen, gezählt. Die Summe der Geschwindigkeiten dieser Fahrzeuge seit Start der Statistik bis zum Zeitschritt t sei mit $\sum v_{boundary}(t)$ bezeichnet.
4. Die Simulation wird so lange fortgesetzt, bis alle Fahrzeuge das System verlassen haben.

Während der Simulation wird der bis dahin durchschnittliche Fluß durch das rechte Systemende über

$$q(i) \quad = \quad \frac{1}{2t} \sum v_{boundary}(t)$$

berechnet. Es zeigt sich, daß auch im zweispurigen Fall der Ausfluß sich dem Maximalfluß eines entsprechenden Systems periodischer Randbedingungen annähert. Dies gilt sowohl für die symmetrischen als auch für die einfach asymmetrischen Regeln. In Abb. 3.13 ist der Maximalfluß und in Abb. 3.14 der durchschnittlich akkumulierte Ausfluß und in für ein symmetrisches System mit Länge $L = 262144$ Sites und Anteilen an langsamen Fahrzeugen von $RatioS = 0.0,\ 0.05,\ 0.15$ dargestellt. Dahinter folgen die beiden entsprechenden Abbildungen für den asymmetrischen Fall. Dieses Verhalten wird als selbstorganisierende Kritikalität [2] bezeichnet, bei der ein oder mehrere Meßgrößen eines Systems (hier der Fluß) selbsttätig kritische Werte annehmen. In Tab. 3.8 sind alle Ergebnisse des Ausflusses

S/A	System	RatioS	Simulation	L	Schritte	Fluß
S	NEC	0.00	Ausfluß	2^{18}	-	0.341 ± 0.06
S	NEC	0.00	periodisch	2^{18}	2^{17}	0.341 ± 0.001
S	NEC	0.05	Ausfluß	2^{18}	-	0.317 ± 0.07
S	NEC	0.05	periodisch	2^{18}	2^{17}	0.317 ± 0.002
S	NEC	0.15	Ausfluß	2^{18}	-	0.313 ± 0.05
S	NEC	0.15	periodisch	2^{18}	2^{17}	0.313 ± 0.001
S	Parsytec	0.00	Ausfluß	2^{18}	-	0.342
S	Paragon	0.00	periodisch	2^{17}	2^{16}	0.343
S	Parsytec	0.05	Ausfluß	2^{18}	-	0.317
S	Paragon	0.05	periodisch	2^{17}	2^{16}	0.318
S	Parsytec	0.15	Ausfluß	2^{18}	-	0.313
S	Paragon	0.15	periodisch	2^{17}	2^{16}	0.314
A	Parsytec	0.0	Ausfluß	2^{18}	-	0.257
A	Paragon	0.0	periodisch	2^{17}	2^{15}	0.255
A	Parsytec	0.05	Ausfluß	2^{18}	-	0.247
A	Paragon	0.05	periodisch	2^{17}	2^{15}	0.248
A	Parsytec	0.15	Ausfluß	2^{18}	-	0.238
A	Paragon	0.15	periodisch	2^{17}	2^{15}	0.242

Tabelle 3.8: *Übersicht über Ausfluß am Stauende*

am Stauende ausgeführt. Die Übereinstimmung zum jeweiligen Vergleichssystem mit periodischen Randbedingungen ist sehr gut. Nur im asymmetrischen Modell bei $RatioS = 0.15$ gibt es eine nennenswerte Abweichung von 0.004, die aber immer noch ganz im Bereich der Fehlertoleranz liegt.

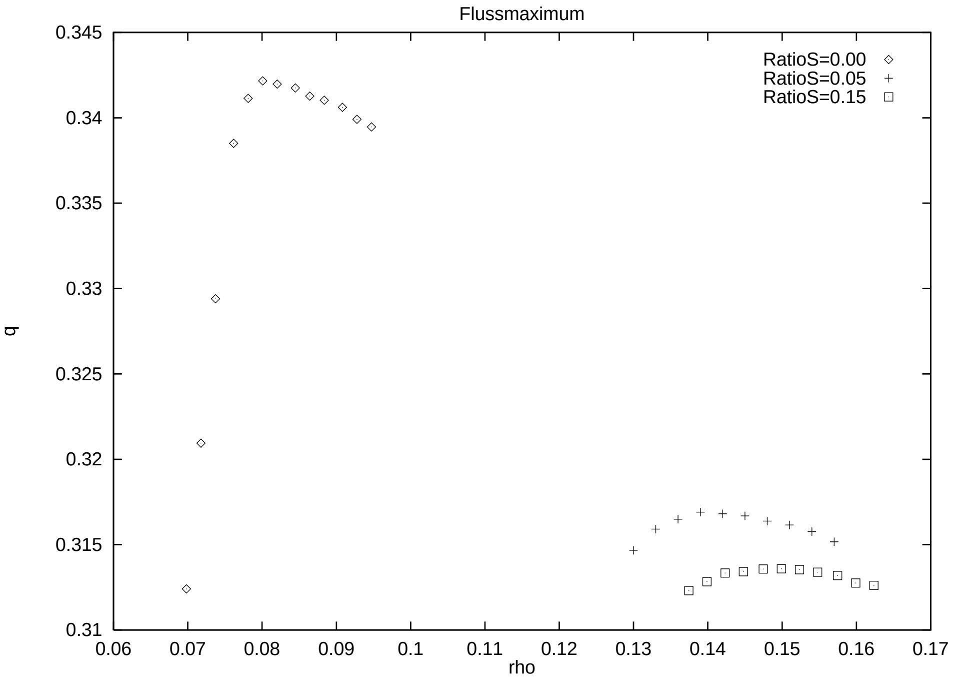

Abbildung 3.13: *Maximaler Fluß im symmetrischen Modell*

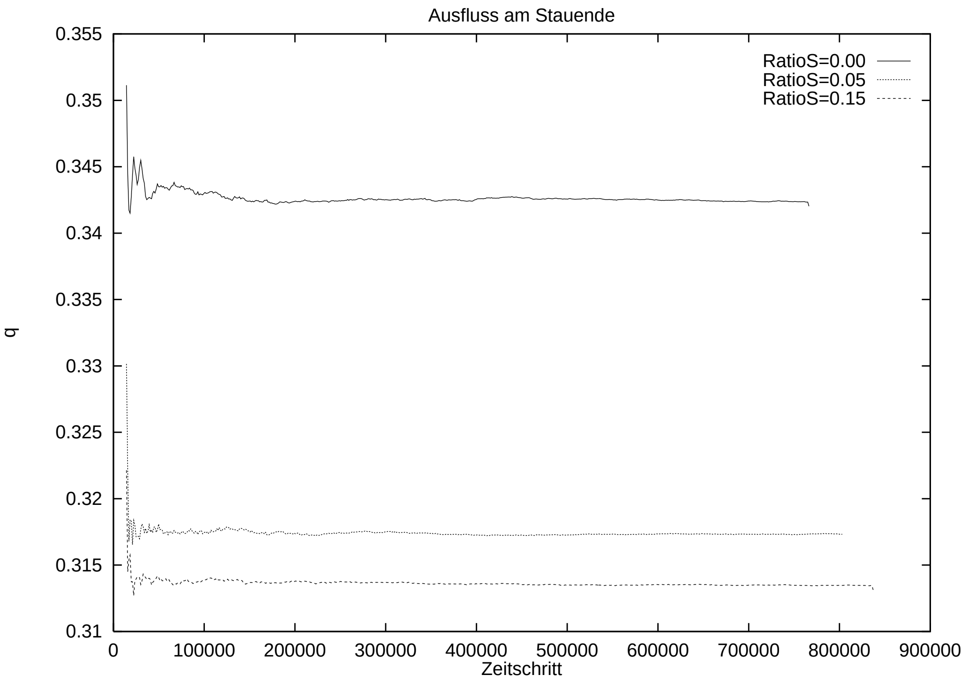

Abbildung 3.14: *Ausfluß im symmetrischen Modell*

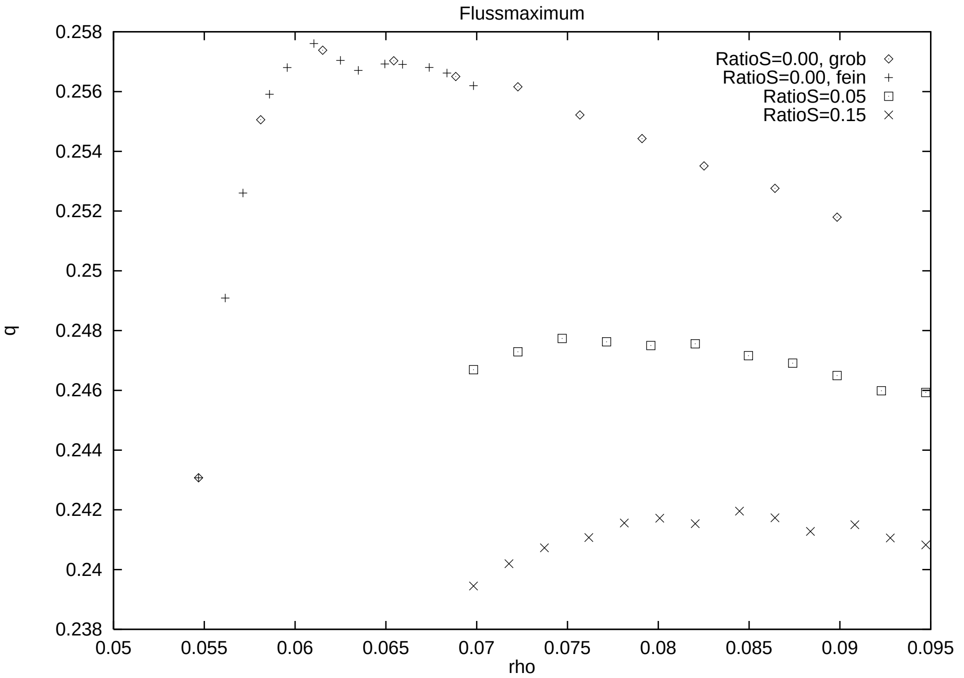

Abbildung 3.15: *Maximaler Fluß im einfach asymmetrischen Modell*

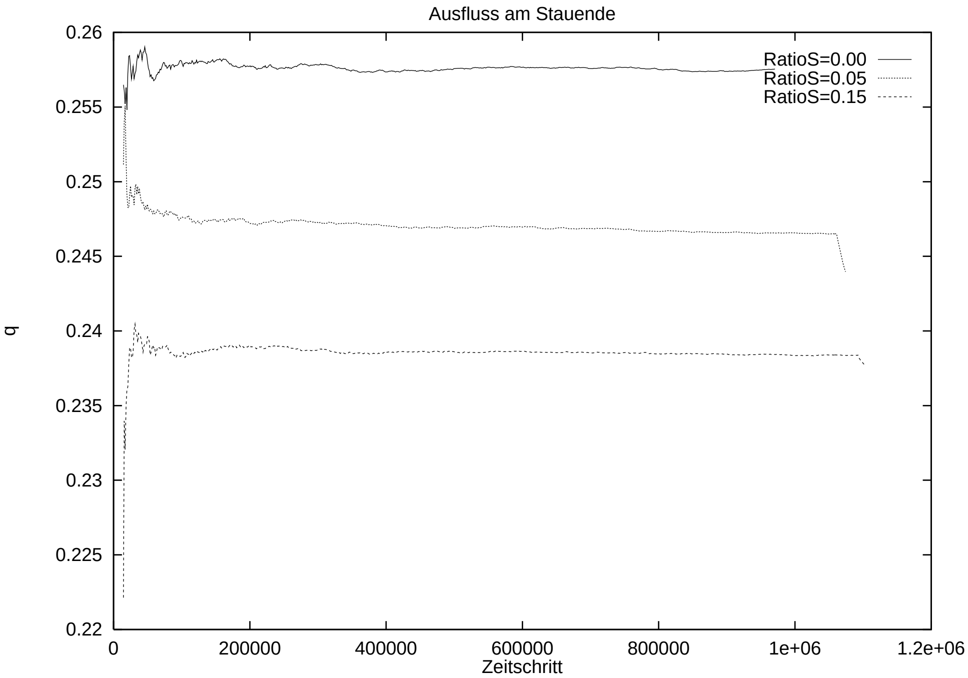

Abbildung 3.16: *Ausfluß im einfach asymmetrischen Modell*

3.8 Finite–Size–Effekt

Bei Simulation von kleinen Systemen mit periodischen Randbedingungen tritt gegenüber großen Systemen eine Abweichung des Flußverlaufs in der Nähe des Maximums auf. Dabei handelt es sich nicht um eine schlechte Statistik durch die geringe Anzahl von Sites, sondern um einen systematischen Fehler, der auch durch die Erhöhung der Simulationsschritte nicht behoben werden kann. Latour hat in [15] das Phänomen für den einspurigen Fall mit periodischen Randbedingungen, Baldus hat in [3] das Phänomen für offene Randbedingungen untersucht. Auch in dieser Arbeit wurde das Problem für den zweispurigen symmetrischen Fall im besonderen Hinblick auf den Anteil langsamer Fahrzeuge untersucht. Im folgenden sind für die Anteile $RatioS = 0.0$ (Abb. 3.17) bis $RatioS = 1.0$ (Abb. 3.23) die Fundamentaldiagramme abgebildet. Man erkennt eine deutliche Verlagerung des Effektes von

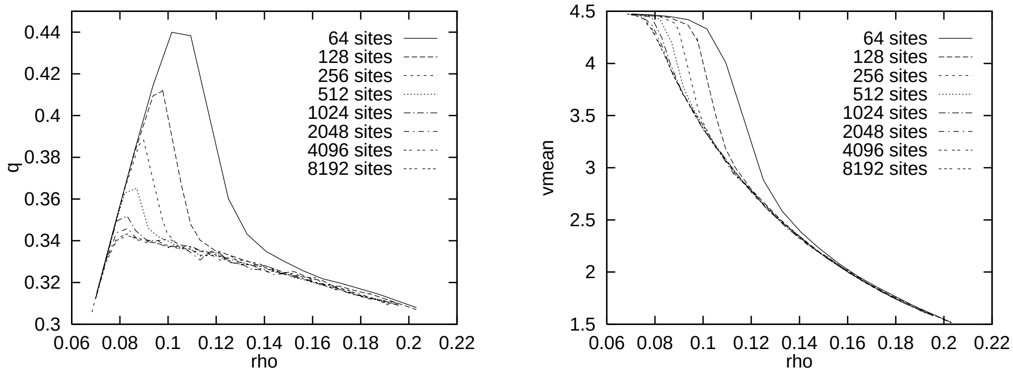

Abbildung 3.17: *Finite–Size–Effekt bei* $RatioS = 0.0$

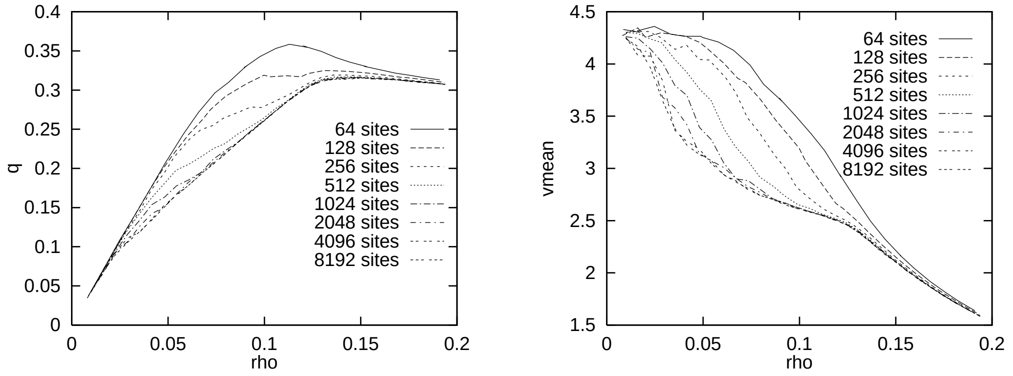

Abbildung 3.18: *Finite–Size–Effekt bei* $RatioS = 0.1$

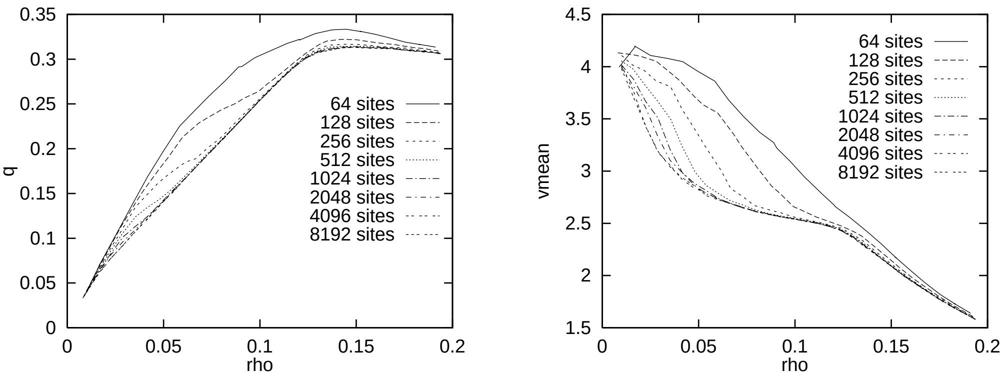

Abbildung 3.19: *Finite–Size–Effekt bei RatioS* $= 0.2$

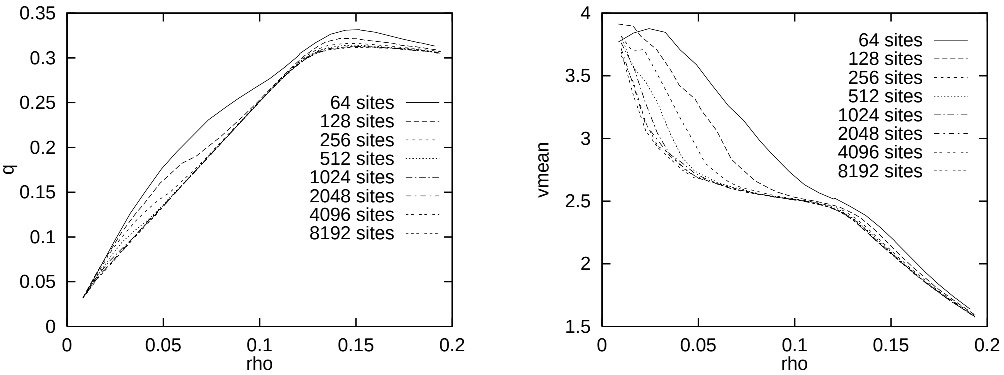

Abbildung 3.20: *Finite–Size–Effekt bei RatioS* $= 0.3$

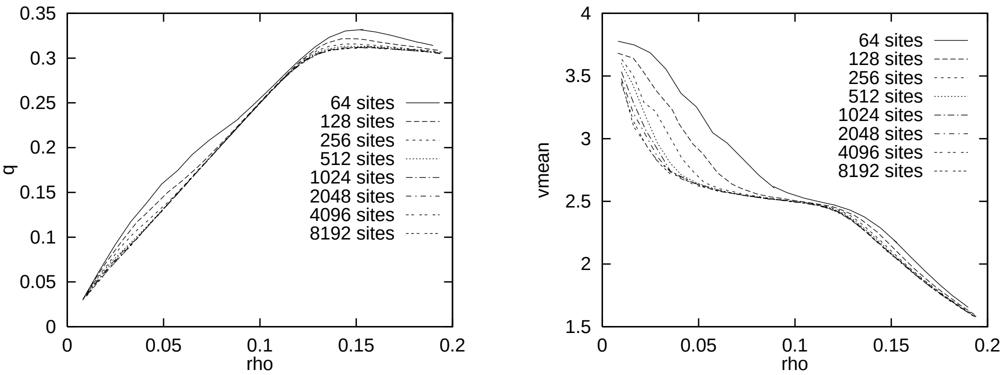

Abbildung 3.21: *Finite–Size–Effekt bei RatioS* $= 0.4$

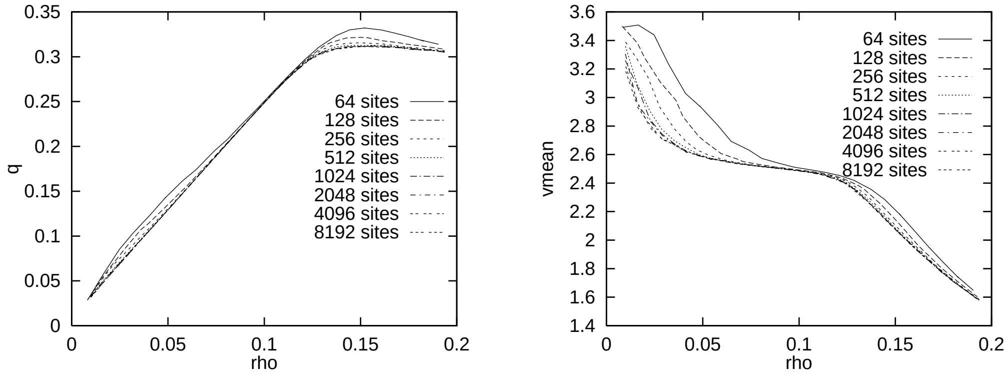

Abbildung 3.22: *Finite–Size–Effekt bei* $RatioS = 0.5$

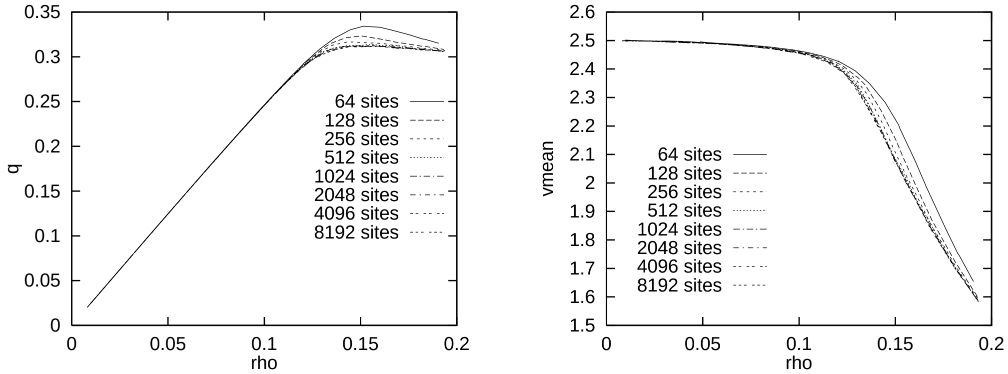

Abbildung 3.23: *Finite–Size–Effekt bei* $RatioS = 1.0$

Dichten rechts von ϱ_c zu Dichten links von ϱ_c. Gleichzeitig wird die Erhöhung abgeschwächt: während bei $RatioS = 0.0$ das Verhältnis der Flußmaxima zwischen dem kleinsten (64 Sites) und dem größten (8192) System bei ca. 1.3 lag, ist es bei $RatioS = 0.1$ schon auf ca. 1.1 gesunken.

Bei den kleinen Systemem mit $L = 64, 128$ fällt besonders die starke Abstufung der Kurven auf, obwohl alle Systeme unabhängig von der Größe mit 30 Meßpunkten im Dichteintervall $0 \ldots 0.2$ berechnet worden sind. Dies liegt an der Tatsache, daß die vorgegebene Dichte nur mit Genauigkeit $1/(2L)$ erreicht werden kann.

Um so schwerer tritt das Problem bei der Festlegung des Anteils an langsamen Fahrzeugen auf. Bei Dichte $\varrho = 0.1$ sind bei $L = 64$ nur 12 Fahrzeuge vorhanden, so daß sich für $RatioS$ ein Fehler von $1/12$ ergibt. In der Simulation wurde dieses Problem folgendermaßen gelöst: wenn N die Gesamtanzahl von Fahrzeugen im System ist, so liegt die korrekte nichtganzzahlige Anzahl der langsamen Fahrzeuge zwischen $\lfloor N RatioS \rfloor$ und $\lceil N RatioS \rceil$. Bei den mehrfach ausgeführten Simulationen wird nun mit der Wahrscheinlichkeit $N RatioS - \lceil N RatioS \rceil$ die höhere Anzahl genommen und sonst die geringere.

In Abb. 3.23 ist der Finite–Size–Effekt für ein System zu sehen, das nur aus langsamen Fahrzeugen besteht. Die Maxima liegen wie im Fundamentaldiagramm für große Systeme der gleichen Maximalgeschwindigkeit bei Dichten um $\varrho = 0.15$. Die Erhöhung für kleine Systeme ist nicht so stark ausgeprägt wie für die schnellen Fahrzeuge (29%), sondern beträgt nur 7%. Durch die unterschiedliche Lage der Maxima bei langsamen und schnellen

Fahrzeugen bildet sich bei gleichmäßiger Mischung der Typen ein Zwischenbereich aus, in dem kein Finit–Size–Effekt zu erkennen ist (siehe Abb. 3.22).

3.9 Dynamisches Fundamentaldiagram

Sehr eng verwandt mit dem Finite–Size–Effekt ist das *dynamische Fundamentaldiagramm*. Es gibt den Zusammenhang zwischen Fluß und Dichte wieder, wie ihn die Verkehrswissenschaftler bei Feldversuchen ermitteln. Sie haben im Gegensatz zu Simulationen kein *beliebig großes* System zur Verfügung, sondern müssen sich mit Meßabschnitten von ca. 500 Metern bis wenigen Kilometern[7] begnügen. Also tritt in gewissem Maße der Finite–Size–Effekt auf. Außerdem ist die Statistik durch die geringe Anzahl der Sites und durch die kleine Anzahl an Stichproben sehr schlecht. Sie erhalten i.A. statt einer glatten Linie eine Ansammlung von Punkten, die grob in drei Teile zerfällt. In Abb. 3.24 ist eine Messung [1] abgebildet, die auf einem Freeway in Ontario, Kanada, durchgeführt worden ist.

Kanadisches Paper

Abbildung 3.24: *Dynamisches Fundamentaldiagramm aus Verkehrsmessungen*

- Für kleine Dichten sieht man einen klar definierten linearen Anstieg, der dem des theoretischen Fundamentaldiagramms enstpricht.
- Für große Dichten ist eine verwaschene Wolke von Punkten zu erkennen, die ungefähr den gleichen mittleren Verlauf hat wie der rechte Teil des Fundamentaldiagramms.
- Dazwischen befindet sich ein Bereich, der geringfügig schwächer mit Meßpunkten bevölkert ist.

Das vorliegende Modell kann — wie auch das einspurige vorher [25] — dieses Verhalten reproduzieren. Dazu wird absichtlich die Statistik verschlechtert, indem nur eine geringe Anzahl von Sites (z.B. 1000) für die Mittelwertberechnung verwendet werden. In Abb. 3.25 ist ein Diagramm für ein System mit symmetrischen Regeln und Größe $L = 20000$ zu sehen, wobei jeweils Teilabschnitte von Länge 500 ausgewertet wurden. Allerdings ist der freie

[7] bis zur größten Entfernung zwischen zwei Netzpunkten

Bereich zwischen Free–Flow und Congested–Flow nicht so deutlich zu erkennen wie im im einspurigen Fall. Das Maximum liegt bei ca. $q_{max} = 0.38$, also ca. 12% oberhalb des

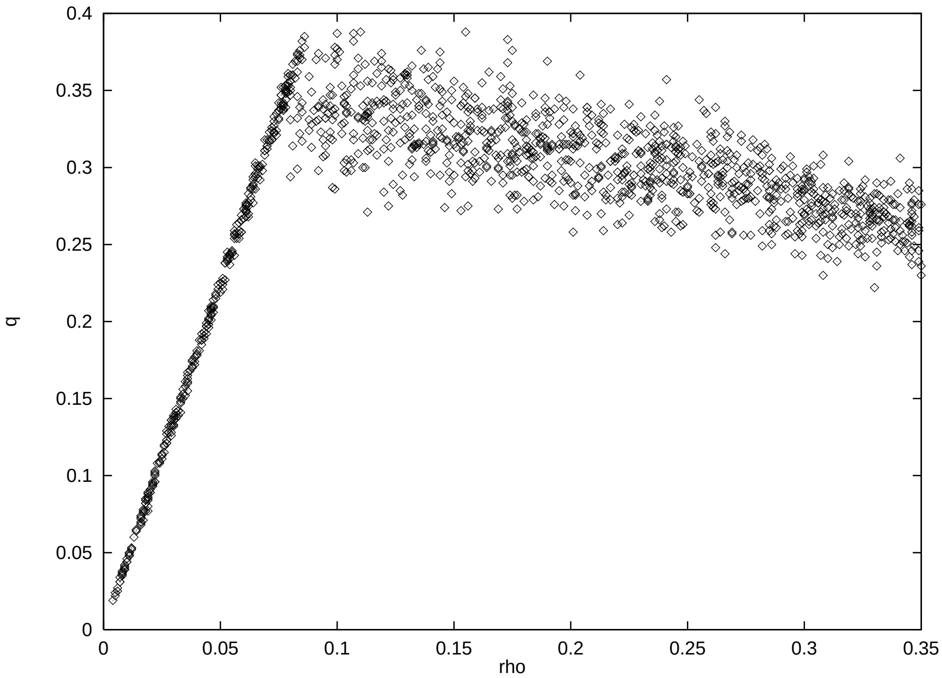

Abbildung 3.25: *Dynamisches Fundamentaldiagramm* $RatioS = 0.0$

Maximums des normalen Fundamentaldiagramms. In Abb. 3.26 und Abb. 3.27 sind die dymanischen Fundamentaldiagramme für das symmetrische Modell mit $RatioS = 0.1$ und

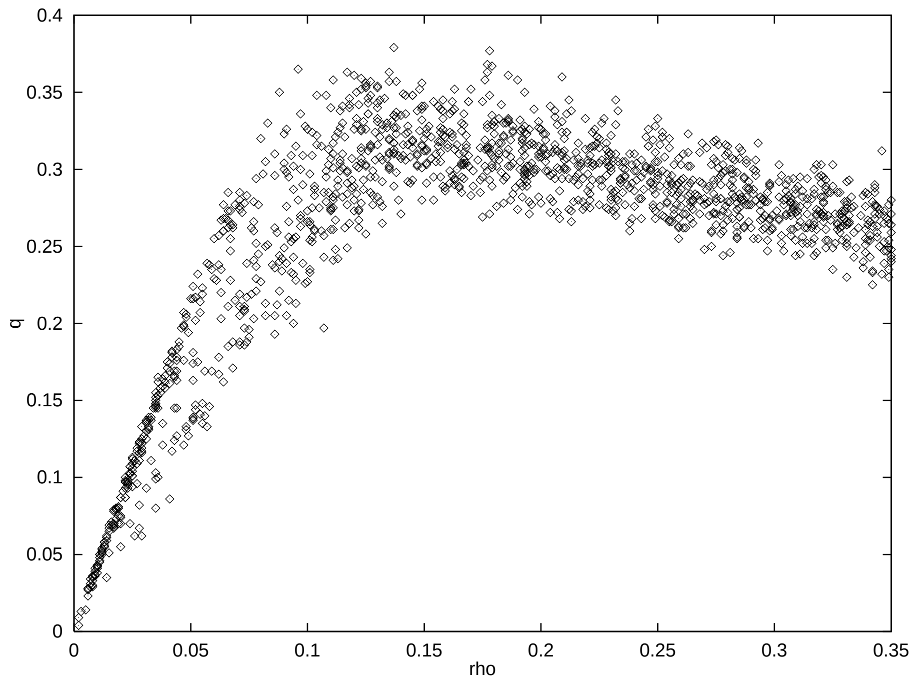

Abbildung 3.26: *Dynamisches Fundamentaldiagramm symmetrisch* $RatioS = 0.1$

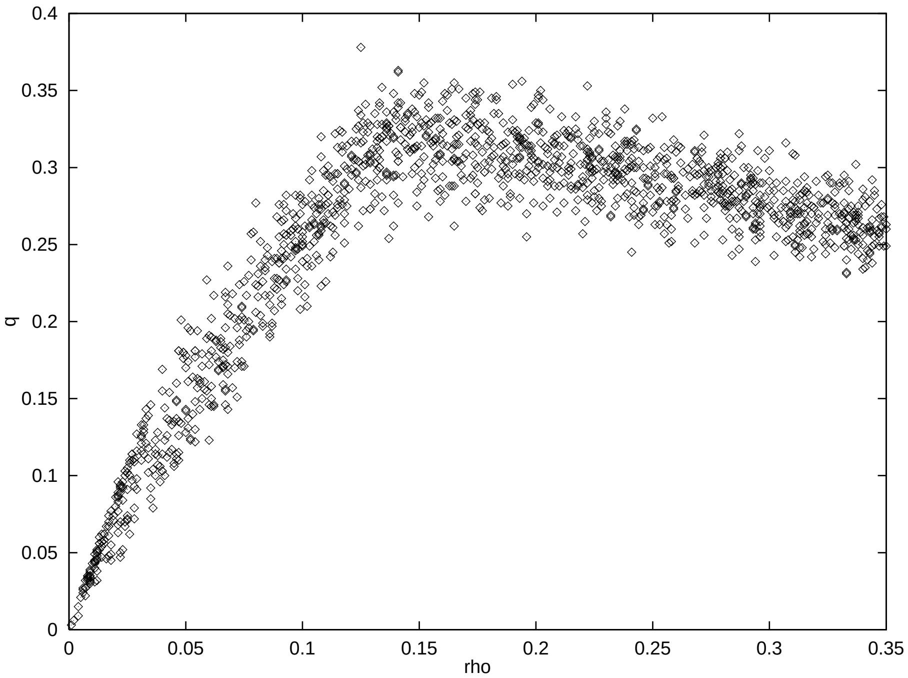

Abbildung 3.27: *Dynamisches Fundamentaldiagramm symmetrisch* $RatioS = 0.2$

$RatioS = 0.2$ abgetragen. Bei $RatioS = 0.1$ erkennt man die leichte Auffächerung des linearen Anstiegs durch den Anteil an langsamen Fahrzeugen. Gleichzeitig kommt es zu einer y–Verbreiterung der Verteilung, besonders im Bereich des Flusmaximums um $\varrho = 0.1$. Bei $RatioS = 0.2$ nimmt die Auffächerung noch stärker zu. Der schwach besiedelte Bereich ist nicht mehr erkennbar.

3.10 Geschwindigkeit

Die rechenintensive Simulation zur Ermittlung der Maxima der Fundamentaldiagramme wurden auf der NEC SX–3 (auf diesem Rechner lief die Simulation in ihrer bitorientierten Version, siehe 6.1.2) im Rechenzentrum Köln, auf dem Parsytec GCel–1024 im Zentrum für Paralleles Rechnen (ZPR) in Köln und auf der Paragon in Jülich durchgeführt. Dabei wurden Systeme mit Länge $L = 2^{17} \dots 2^{18}$ und Zeitschritte im Bereich[8] von $T = 2^{15} \dots 2^{17}$ betrachtet. Die Dichten (Anzahl n_ϱ) lagen im Bereich von $\varrho = 0.07 \dots 0.15$. Die tatsächliche Anzahl von Updates ergibt sich aus $2L(T + L/20)(n_\varrho + 1)$. Ein Vergleich der Geschwindigkeiten ist in Tab. 3.9 zu sehen. Die Einheit MUPS[9] gibt die Anzahl von Sites in Millionen an, für die pro Sekunde ein kompletter Zeitschritt durchgeführt wird. Eine intensive Un-

System	CPN	L	Schritte	Dichten	time [h]	MUPS/CPN	MUPS
NEC	(64)	2^{18}	2^{17}	10	0.93	(3.2)	204
Paragon	32	2^{17}	2^{16}	10	2.61	0.69	22.1
Paragon	64	2^{17}	2^{15}	30	2.51	0.62	39.9
Parsytec	128	2^{16}	2^{10}	10	0.46	0.029	3.74
Parsytec	256	2^{17}	2^{10}	10	0.88	0.027	6.93

Tabelle 3.9: *Rechengeschwindigkeit des zweispurigen Modells*

tersuchung der Abhängigkeit der Rechengeschwindigkeit von Länge und Anzahl der CPN wurde von Nagel und Schleicher [24] durchgeführt. Diese Ergebnisse können für den zweispurigen Fall übernommen werden, da der Ansatz der Parallelisierung identisch mit dem des einspurigen Falls ist.

[8] Beim Parsytec wurde die Geschwindigkeit in Testläufen gesondert gemessen. Daraus folgt die geringe Anzahl an Zeitschritten.

[9] Englisch: mega updates per second

Kapitel 4

Das Netzmodell

4.1 Problematik

Simulationen auf Systemen mit periodischen Randbedingungen liefern — wie in den beiden vorangegangenen Kapiteln erläutert — interessante theoretische Resultate, die sich zu einem guten Teil auch direkt mit den in der Realität beobachteten Phänomen vergleichen lassen. Dennoch ist dieses Modell in den folgenden Aspekten weit von den tatsächlichen Gegebenheiten im Straßenverkehr entfernt:

- Ein realistisches Straßennetz besteht aus einer erheblichen Anzahl von Abschnitten, deren Längen beträchtlich kürzer sind als die für die Ermittlung der theoretischen Ergebnisse verwendeten. Letztere waren gewählt worden, um die Statistik der Messung zu verbessern, indem der Finite–Size–Effekt unterdrückt wird (siehe 3.8). In der Realität ist der Finite–Size–Effekt also kein systematischer Fehler wie aus Sicht der statistischen Physik, sondern ein unverzichtbarer Teil des Modells. Mit seiner Hilfe können die Diskrepanzen in der Eichung der Fundamentaldiagramme des zellularen Automaten mit denen des Straßenverkehrs zum Teil erklärt werden (siehe dazu 3.8 und 3.9).

- Bei einem Straßennetz unterscheiden sich Einzelabschnitte nicht nur durch die Länge, sondern auch durch individuelle Eigenschaften, die die Kapazität der Straße mit beeinflussen. Streckenverläufe mit Geschwindigkeitsbeschränkungen, starken Krümmungen oder Steigungen sind anders zu bewerten als freie gerade Streckenabschnitte. Dabei müssen die beeinflußenden Faktoren nicht unbedingt so offentlichtlich sein wie die gerade genannten. Es gibt durchaus subtile Eigenschaften, die sich der oberflächlichen Betrachtung entziehen. Da ist zum Beispiel die Breite der Fahrbahnen zu nennen. Zwar ist auch bei engen Fahrbahnen für jedes Fahrzeug genügend Raum zum Nachbarn vorhanden, jedoch führt der subjektive Eindruck von Platzmangel zu einer direkten Beeinträchtigung der Fahrweise. Weitere Aspekte sind die Beschaffenheit (bzw. das Vorhandensein) eines Seiten– oder Mittelstreifens oder die umliegende Bebauung.

- Beim Modell mit periodischen Randbedingungen wird das System am Anfang *einmal* mit den jeweils gültigen Anfangsbedingungen initialisiert. Nach Start der Simulation evolviert das System frei von jeglichen äußeren Einflüssen lediglich der eigenen Dynamik unterworfen. Der Straßenverkehr in einem Netz ist jedoch nicht nur abhängig von Anfangsbedingungen zu einem gewissen Zeitpunkt, sondern wird *fortwährend* durch äußere und innere Phänome beeinflußt. Zu den von außen kommenden Einflüssen

May p.59

May p.57

Abbildung 4.1: *Einflüsse der Tageszeit und des Wochentags*

gehören Fahrzeuge, die an gewissen Punkten[1] in das Netz eingespeist (emittiert) werden bzw. die Vorschriften, nach denen Fahrzeuge aus dem Netz entfernt (absorbiert) werden. Dies hat als direkte Konsequenz, daß die Gesamtanzahl an Fahrzeugen nicht mehr konstant, sondern auf vielfältige Weise parametrisiert ist. Als wichtigste Parameter des Verkehrsaufkommens sind die *Tageszeit* und der *Wochentag* zu nennen (siehe Abb. 4.1 aus [17]).

Zu den inneren Einflüssen gehören zum Beispiel zeitliche Veränderungen in der Beschaffenheit der Straßen und deren Umgebung. Eine schneebedeckte Straße bei Nebel hat eine erheblich geringere Kapazität als eine trockene Straße bei sonnigem Wetter. Bei dieser Beeinflußung durch Witterung ist eine Unterteilung des Straßennetzes in Großräume ausreichend. Anders sieht es bei lokalen Störungen wie zum Beispiel Tagesbaustellen und Unfällen aus. Dort werden nur einzelne Punkte des Netzes für kurze Zeit in ihrer Dynamik geändert, was aber in der Regel zu nicht–lokalen Auswirkungen führt.

Es sollte erwähnt werden, daß außer dem Ansatz, das eindimensionale Modell auf ein Netz zu erweitern, noch die Möglichkeit besteht, ein zweidimensionales Gitter zu definieren, auf dem der Verkehrsfluß horizontal und vertikal fließen kann ([6], [10] und [19]), wobei für beide Dimensionen im Normalfall periodische Randbedingungen angesetzt werden. Diese Modelle eignen sich eher für die Simulation von Stadtverkehr, da bei ihnen die Dynamik fast ausschließlich durch das Verhalten an den Kreuzungen festgelegt ist. Es gibt im Gegensatz zum Netzmodell keine langgestreckten Abschnitte mit ungestörtem Verkehrsfluß.

4.2 Das Modell

Im Rahmen dieser Diplomarbeit wurde ein mikroskopisches Modell zur Verkehrssimulation in Netzen erstellt. Es trägt den Namen *PAMINA* für *Paralleler Mikroskopischer Netzwerk–Algorithmus*[2]. Mit seiner Hilfe ist es möglich, reale Netze auf gängiger Hardware in Echtzeit

[1] Endstücke, Anschlüsse, Dreiecke, Kreuze

[2] Praktischerweise ist diese Abkürzung auch in der englischen Übersetzung sinnvoll

oder schneller zu berechnen. Für die meisten Berechnungen diente das Autobahnnetz von Nordrhein-Westfalen als Datengrundlage (siehe 4.4.2). Es hat eine Gesamtlänge von 1950 Kilometer pro Fahrrichtung. Auf einem Cluster[3] von vier Workstations des Typs Sparc10 erreicht man einen Echtzeitfaktor von sechs bei einer Belegung von ca. 110000 Fahrzeugen. Das bedeutet, daß sechs Sekunden Simulation in einer Sekunde Echtzeit ablaufen. Diese hohe Geschwindigkeit wird durch eine aufwendige Aufbereitung der Ausgangsdaten und durch eine hardwarenahe Implementation erreicht.

Im folgenden Abschnitt werden die einzelen Elemente, aus denen ein komplettes Netz aufgebaut ist, erläutert. Es folgt eine kurze Beschreibung der Schritte, die notwendig sind, um aus Rohdaten, die in einer Datei auf einem externen Speichermedium vorliegen, ein Netz aufzubauen und dieses im Arbeitsspeicher eines Rechners abzulegen. Anschließend wird darauf eingegangen, auf welche Weise dieses Netz auf die zur Verfügung stehenden Recheneinheiten eines Parallelrechners verteilt wird. Im letzten Abschnitt folgt eine kurze Erläuterung der verschiedenen Modi, in denen die Simulation betrieben werden kann.

4.3 Netzelemente

Bei der Implementation von *PAMINA* wurde darauf geachtet, daß das gesamte Netz aus einer möglichst kleinen Anzahl von Grundelementen aufgebaut wird. Dies hat zum einen den Vorteil eines geringeren Programmieraufwandes, zum anderen kommt man damit der Philosophie der zellularen Automaten nahe (siehe 2.1), bei denen gerade eine geringe Anzahl von Regeln angestrebt wird. Im folgenden werden die vier Grundelemente beschrieben, anschließend drei weitere aus Grundelementen zusammengesetzte Kombinationselemente. Dabei werden Begriffe aus der Graphentheorie verwendet, die in Anhang A.2 erläutert werden.

4.3.1 Grundelemente

Einspuriger Transferabschnitt

Der *einspurige Transferabschnitt*, der dem Modell von Nagel und Schreckenberg (siehe 2.2) enstpricht, wird verwendet, um Fahrzeuge

- von einem zweispurigen Straßenabschnitte zu einem anderen,
- von einem Endstück zu einer Auffahrt oder
- von einer Abfahrt zu einem Endstück zu transportieren.

Seine Länge kann variiert werden, was besonders für seine Aufgabe als Verbindung zwischen zweispurigen Straßenabschnitten wichtig ist: bei ausreichender Länge dient er als Puffer zwischen Quell– und Zielstraßenabschnitt. Dadurch werden Rückstaus wie in der Realität erst mit einer gewissen Zeitverzögerung an den Quellabschnitt weitergegeben.

Die dynamischen Eigenschaften des Transferabschnittes können individuell beeinflußt werden. Eine Zusammenstellung befindet sich in Tab. 4.1. Diese Eigenschaften werden in der bisherigen Version *global* für alle Transferstrecken im Netz festgesetzt.

[3] hier: Zusammenschaltung von Computern

Symbol	Beschreibung
l_t	Länge der Transferstrecke in Sites
d_p	Verzögerungswahrscheinlichkeit
v_{max}	Höchstgeschwindigkeit

Tabelle 4.1: *Eigenschaften des Transferabschnitts*

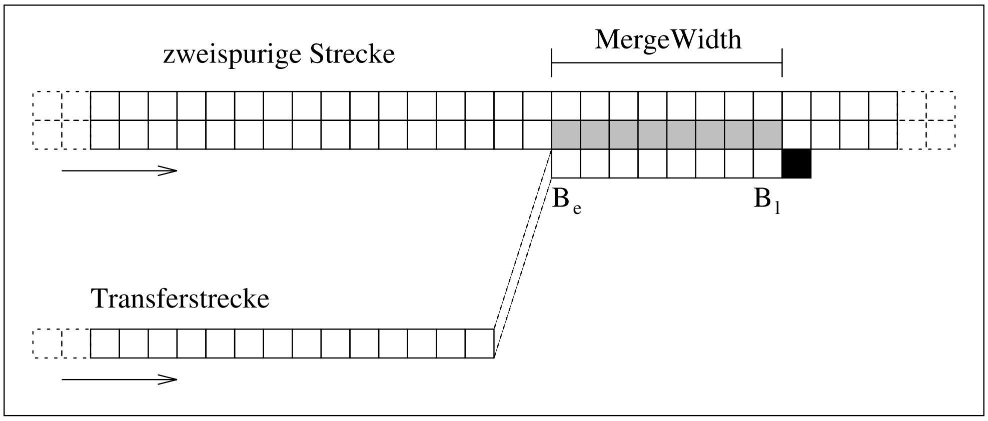

Abbildung 4.2: *Geometrie eines Beschleunigungsstreifens* — Der Wechsel der Fahrzeuge vom Beschleunigungsstreifen auf die zweispurige Fahrbahn findet in den grau schraffierten Sites auf einer Länge von *MergeWidth* statt. Die Fahrzeuge werden durch eine Blockade (schwarzes Site) an der Weiterfahrt auf der Transferstrecke gehindert.

Zweispuriger Straßenabschnitt

Die *zweispurigen Straßenabschnitte* entsprechen der symmetrischen Version des in 3.2 beschriebenen Modells. Auf ihnen bewegen sich die Fahrzeuge, bis sie entweder an Dreiecken, Kreuzen oder Abfahrten von einer Transferstrecke absorbiert werden (siehe unten) oder ein Endstück erreichen. Den zweispurigen Streckenelementen können alle Eigenschaften der einspurigen Transferabschnitte zugewiesen werden. Zusätzlich sind folgende Eigenschaften möglich:

- Ein oder mehrere Sites eines Abschnitts können blockiert werden. Dadurch ist die Modellierung eines Unfalls oder einer Baustelle möglich.
- Ein Abschnitt kann mit einem Überholverbot versehen werden.

Beschleunigungs– und Verzögerungsstreifen

Die *Beschleunigungs– und Verzögerungsstreifen* übernehmen im Modell die Emission von Fahrzeugen von Transferabschnitten in zweispurige Straßenabschnitte bzw. umgekehrt die Absorption von Fahrzeugen von zweispurigen Straßenabschnitten in Transferabschnitte.

Der Beschleunigungsstreifen stellt jeweils das Ende eines Transferabschnittes dar. Auf einer Breite von $MergeWidth$ Sites kann der Austausch zwischen Transferabschnitt und zweispurigem Straßenabschnitt stattfinden. Die Geometrie ist Abb. 4.2 zu entnehmen: Das Programm verwaltet während der Simulation eine Tabelle aller Beschleunigungsstreifen und

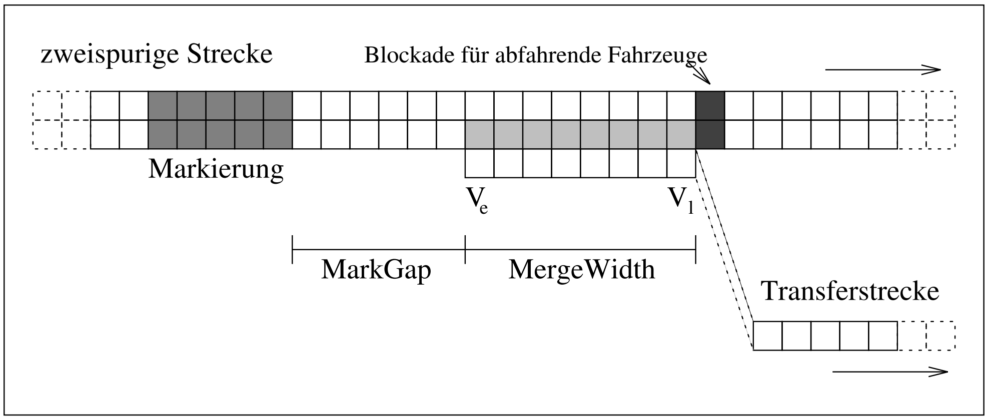

Abbildung 4.3: *Geometrie eines Verzögerungsstreifens* — Der Wechsel der Fahrzeuge vom zweispurigen Abschnitt zum Transferabschnitt findet in den hellgrau schraffierten Sites auf einer Länge von *MergeWidth* statt. Die Markierung des Abbiegewunsches wird im Abstand *MarkGap* durchgeführt. Abbiegende Fahrzeuge werden durch eine Blockade an der Weiterfahrt auf der zweispurigen Strecke gehindert. Sie ist unsichtbar für alle anderen Fahrzeuge.

arbeitet diese pro Zeitschritt einmal ab. Dabei wird für jeden einzelnen Streifen folgendermaßen verfahren: Betrachte den ersten Site des Streifens (B_e). Falls sich dort ein Fahrzeug befindet, überprüfe, ob die linken hinteren 5 Sites und der unmittelbar linke Nachbar auf der zweispurigen Fahrbahn frei sind. Wenn ja, entferne das Fahrzeug aus dem Beschleunigungsstreifen und füge es in das benachbarte Site der zweispurigen Bahn ein. Verfahre anschließend genau so für alle weiteren Sites des Beschleunigungsstreifens bis B_l. Die Überprüfung der hinteren 5 Sites entspricht genau der Wechselregel $S3$:*Genügend Abstand zum Nachfolger?* für den Wechsel nach links im zweispurigen Modell.

Der Verzögerungsstreifen stellt jeweils den Anfang eines Transferabschnittes dar. Auf einer Breite von $MergeWidth$ Sites kann der Austausch zwischen Transferabschnitt und zweispurigem Straßenabschnitt stattfinden. Die Geometrie ist Abb. 4.3 zu entnehmen. Das Programm verwaltet während der Simulation eine Tabelle aller Verzögerungsssstreifen und arbeitet diese pro Zeitschritt einmal ab. Dabei wird für jeden einzelnen Streifen folgendermaßen verfahren: Betrachte den ersten Site des Streifens (V_e). Falls dieser Site leer ist, überprüfe, ob der unmittelbar linke Nachbar auf der zweispurigen Fahrbahn belegt ist und dieser die Markierung `LeftRoute` oder `RightRoute` trägt (siehe 4.3.3). Wenn ja, entferne das Fahrzeug aus der zweispurigen Bahn und füge es in das benachbarte Site des Verzögerungsstreifens ein. Verfahre anschließend genau so für alle weiteren Sites des Verzögerungsstreifens bis V_l.

Bei beiden Streifentypen werden durch den Austausch der Fahrzeuge zwischen benachbarten Sites die Fahrzeuge nicht tatsächlich vorwärtsbewegt. Dies geschieht nur in den Bewegungsschritten für zweispurige Strecken und Transferstrecken (siehe 4.6.6).

Endstück (Terminal)

Ein *Endstück* dient als Abschluß von zweispurigen Straßenabschnitten bei Autobahnendpunkten und von einspurigen Transferstrecken bei Auf– und Abfahrten. Mit seiner Hilfe werden dem System Randbedingungen aufgeprägt. Man unterscheidet dabei zwischen *Quellen*, an denen Fahrzeuge ins Netz emittiert, und *Senken*, an denen Fahrzeuge aus dem Netz absorbiert werden. Jeweils eine Kombination aus Quelle und Senke bilden ein Endstück.

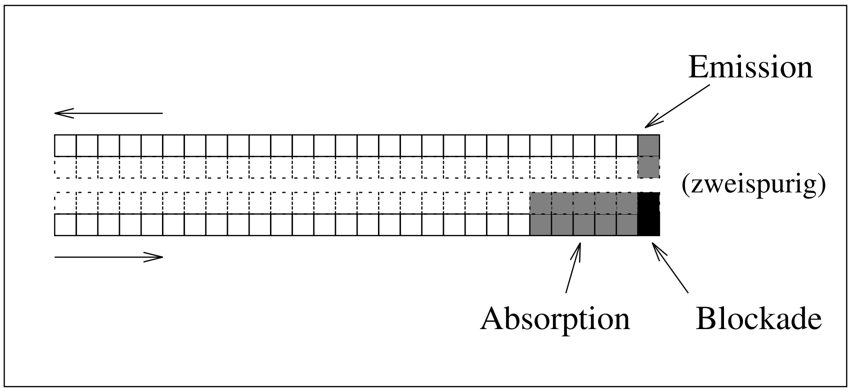

Abbildung 4.4: *Geometrie eines Endstücks* — Im oberen Abschnitt wird im grau schraffierten Site emittiert (Quelle), im unteren Abschnitt absorbiert (Senke). An der Senke werden die Fahrzeuge durch eine Blockade an der Weiterfahrt gehindert. Die gestrichelten Fahrbahnen gelten für Endstücke an zweispurigen Abschnitten.

Wie viele Fahrzeuge absorbiert und emittiert werden, wird durch Raten festgelegt, die in 4.3.3 erläutert werden. In Abb. 4.4 ist ein Endstück eines zweispurigen Straßenabschnittes zu sehen. Eine Beeinflussung der Fahrzeuge findet jeweils nur in den grau schraffierten Bereichen statt. Bei den Quellen werden Fahrzeuge eingesetzt bis entweder die gewünschte durch Emissionsraten vorgebene Anzahl erreicht wird oder der Bereich komplett gefüllt ist. Bei den Senken werden die Fahrzeuge mit einer durch die Absorptionsraten definierten Wahrscheinlichkeit absorbiert. Am Ende einer Senke befindet sich eine Blockade (schwarze Felder), die einen Rückstau erzeugt, falls Fahrzeuge nicht schnell genug das System verlassen.

4.3.2 Kombinationselemente

Die *Kombinationselemente* setzen sich jeweils aus mehreren Grundelementen zusammen. Nachdem das Netz im Arbeitsspeicher aufgebaut worden ist, verlieren sie im Prinzip ihre Bedeutung, d.h. die Dynamik der gesamten Simulation setzt sich nur noch aus den dynamischen Eigenschaften der Grundelemente zusammen. Dennoch werden sie nicht komplett gelöscht, denn für Ausgabe von Statistiken ist ein Bezug auf die ursprünglichen Eigenschaften (insbesondere Namen) der Kombinationselemente sinnvoll.

Autobahnanschluß

Ein Netzknoten mit Valenz zwei, der zusätzlich einen Namen hat, wird zu einem *Autobahnanschluß* (siehe 4.4.3). Er wird überführt in zwei Endstücke (mit jeweils einer Quelle und einer Senke), zwei Verzögerungsstreifen, zwei Beschleunigungsstreifen und vier Transferstrecken. Die Geometrie ist Abb. 4.5 zu entnehmen. Die Quellen sind über Transferstrecken der Länge t_l mit Autobahnauffahrten, die Senken über Transferstrecken gleicher Länge mit Autobahnabfahrten verbunden.

Autobahndreieck

Ein Netzknoten mit Valenz drei wird zu einem *Autobahndreieck*. Er wird überführt in zwei Auffahrten, zwei Abfahrten und vier Transferstrecken. Die Geometrie ist Abb. 4.6 zu

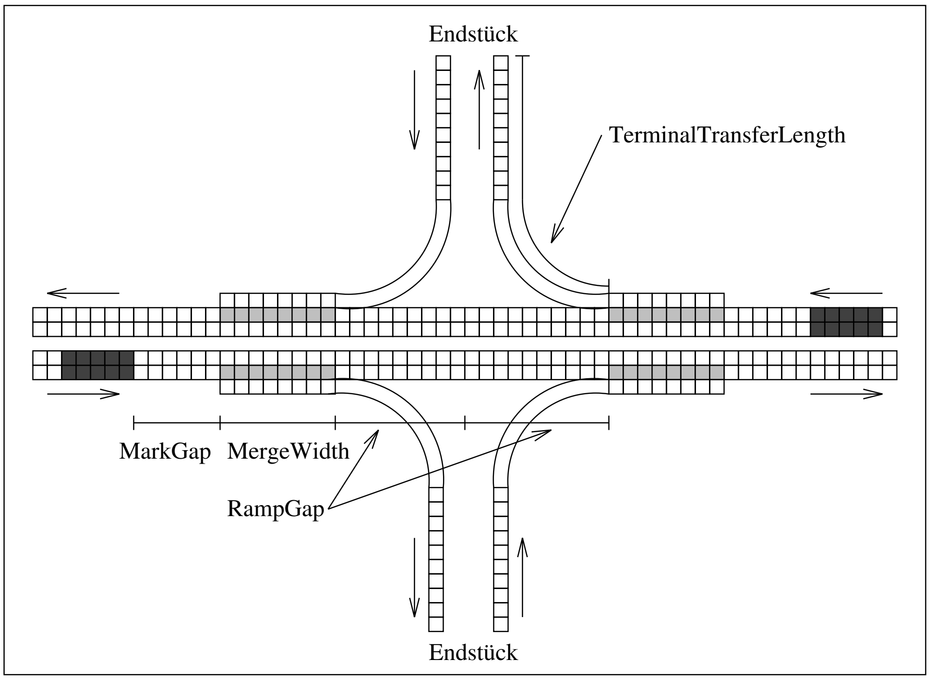

Abbildung 4.5: *Geometrie eines Autobahnanschlusses* — Der Autobahnanschluß ist aus zwei Endstücken, zwei Verzögerungsstreifen, zwei Beschleunigungsstreifen und vier Transferstrecken aufgebaut.

entnehmen. Im Gegensatz zu den Autobahnkreuzen, gibt es bei Autobahndreiecken in der Realität sehr viele Möglichkeiten, den Spurverlauf beim Übergang zwischen den Abschnitten zu regeln. Die beiden Extremfälle sind folgende (neben beliebigen Abstufungen dazwischen):

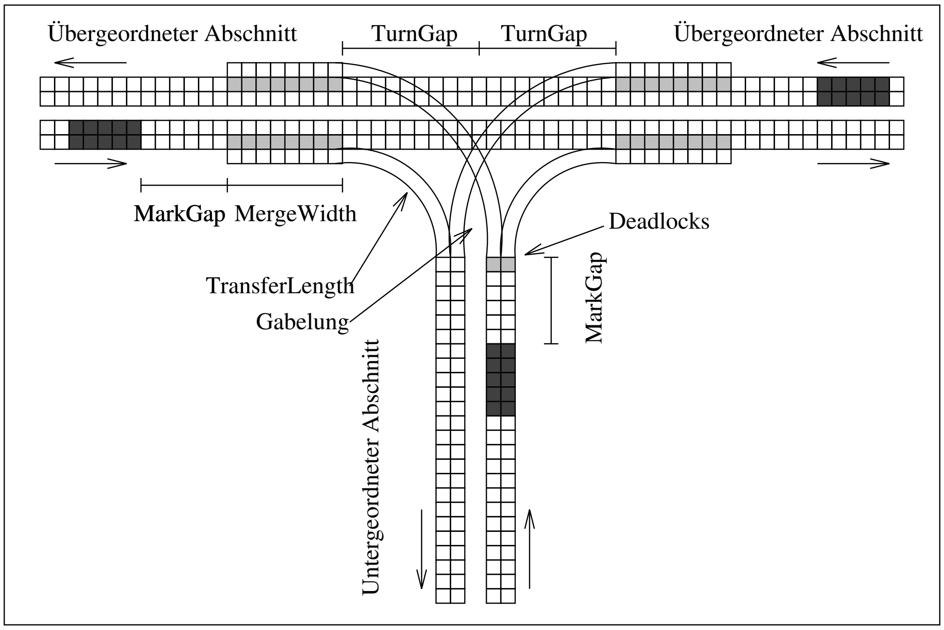

Abbildung 4.6: *Geometrie eines Autobahndreiecks* — Ein Autobahndreieck ist aus zwei Verzögerungstreifen, zwei Beschleunigungsstreifen und 4 Transferstrecken aufgebaut. An der Gabelung des untergeordneten Abschnitts können Deadlocks entstehen (siehe 4.3.2)

- Zwei der drei Straßenabschnitte sind *übergeordnet*, d.h. auf diesen Strecken kann der Verkehr ohne Beeinflußung durch die dritte *untergeordnete* Strecke fließen. Ein Austausch an Fahrzeugen findet lediglich über Beschleunigungs– und Verzögerungstrecken statt. Dieser Typ ist am häufigsten in realen Netzen vorhanden, da meist die dritte Strecke zu einem späteren Zeitpunkt errichtet wurde und somit die vorhandenen Strecken durchgehend blieben.
- Alle drei Straßenabschnitte sind *gleichberechtigt*. Der Austausch der Fahrzeuge findet über Aufspaltung bzw. Zusammenführung der Fahrbahnen statt. Diese Version hat im Gegensatz zur ersten den Vorteil, daß sie *symmetrisch*, also keine der drei Strecken ausgezeichnet ist.

In *PAMINA* wurde bis jetzt nur der erste Typ realisiert. Der andere läßt sich jedoch genau so auf Grundelemente zurückführen und könnte Teil einer späteren Erweiterung sein. Dadurch, daß der asymmetrische Typ gewählt wurde, tritt das Problem auf, einen der drei adjazenten Abschnitte als *untergeordnet* einstufen zu müssen. Im Prinzip sollte diese Information Teil der Input–Daten sein. Leider ist das aber de fakto selten bei digitalisierten Karten der Fall. Auch eine visuelle Überprüfung anhand von Straßenkarten bietet keinen Ausweg, da diese den genauen Verlauf der Dreiecke im allgemeinen nicht darstellen. Also bleibt nur die Möglichkeit, eine der drei Strecken automatisch über ein einfaches Kriterium bestimmen zu lassen. Dies geschieht folgendermaßen:

1. Ermittle für jedes Paar der drei Abschnitte den eingeschlossenen Winkel.
2. Deklariere die beiden Abschnitte, deren eingeschlossener Winkel die kleinste Differenz zu π hat, als übergeordnet.

Dabei wird also angeommen, daß diejenige Strecke durchgehend ist, die den "kleinsten Knick" im Verlauf hat.

Deadlocks an Autobahndreiecken

Beim Aufteilen des Fahrzeugstromes des untergeordneten Abschnitts an Dreiecken kann es vorkommen, daß dicht nebeneinanderfahrende Fahrzeuge für den Wechsel in entgegensetzte Richtungen markiert werden, so daß das *beide* Fahrzeuge auf die Spur des Nachbarn wechseln müssen. Dies ist jedoch aufgrund der Regeln, die den Spurwechsel festlegen, nicht immer direkt möglich. Im ungünstigsten Fall verbleiben beide Fahrzeuge bis zum letzten Site des Abschnitts auf ihrer alten Fahrbahn und sind so gezwungen auf diesem anzuhalten, da sonst eine falsche Richtung eingeschlagen würde. Diese Situation wird *Deadlock* genannt (Abb. 4.6). Sie wird in der Simulation durch Vertauschen der beiden Fahrzeuge aufgelöst.

Autobahnkreuz

Ein Netzknoten mit Valenz vier wird zu einem Autobahnkreuz. Er wird überführt in vier Auffahrten, vier Abfahrten und acht Transferstrecken. Die Geometrie ist Abb. 4.7 zu entnehmen. Beim Autobahnkreuz wird ebenfalls die Lage der vier adjazenten Abschnitte zueinander untersucht. Dabei werden verständlicherweise jeweils die Strecken, die sich gegenüber liegen, als durchgehend angenommen.

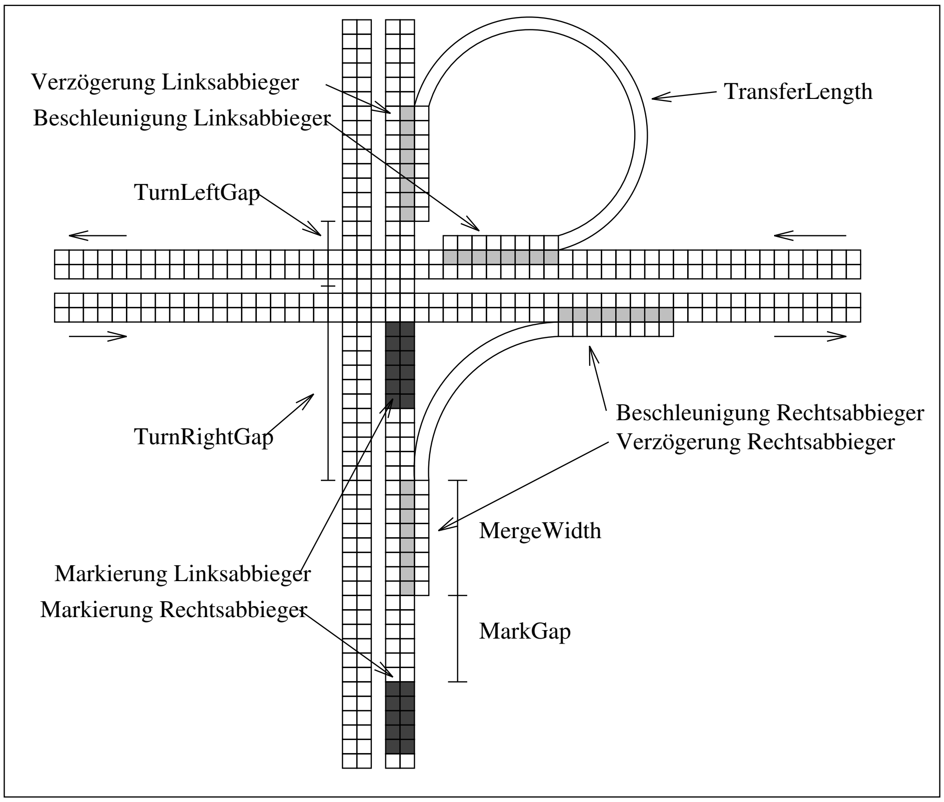

Abbildung 4.7: *Geometrie eines Autobahnkreuzes* — Die Abbildung zeigt exemplarisch eine Links– und eine Rechtsabbiegerspur. Auf den hellgrau schraffierten Sites findet der Austausch der Fahrzeuge zwischen Transferstrecken und zweispurigen Strecken auf einer Länge *MergeWidth* statt. Die dunkelgrau schraffierten Sites zeigen die Markierungsbereiche, in denen den Fahrzeugen das Abbiegemerkmal aufgeprägt wird. Sie haben die Länge $v_{max} = 5$ und den Abstand *MarkGap* vom Verzögerungsstreifen.

4.3.3 Randbedingungen und Parameter

Im folgenden werden alle Parameter beschrieben, die den Ablauf der Simulation beeinflußen. Dabei handelt es sich zum einen um zeitlich konstante, den Straßenabschnitten zugeordnete als auch um zeitabhängige meist Netzknoten zugeordnete Eigenschaften.

Auswahl der Streckeneigenschaften

Die *PAMINA* zugrundeliegenden Modelle erlauben nur eine eingeschränkte Beeinflussung ihrer Dynamik. Deswegen muß bei der Simulation eines realen Netzwerks eine Möglichkeit gefunden werden, die Vielzahl von in der Realität auftretenden Parameter auf die wenigen in zellularen Automaten verfügbaren Parameter zu übertragen (siehe dazu Tab. 4.1 und die Aufzählung in 4.3.1). Da die Parameter Höchstgeschwindigkeit und Überholverbot, sowie das Sperren von Sites ein direktes Gegenstück in der Realität besitzen, kann darüberhinaus kaum eine weitere Parametrisierung erfolgen. Es bleibt als einziger Parameter die Verzögerungswahrscheinlichkeit p_d, um alle weiteren Parameter (z.B. der Witterungseinflüsse) abzudecken.

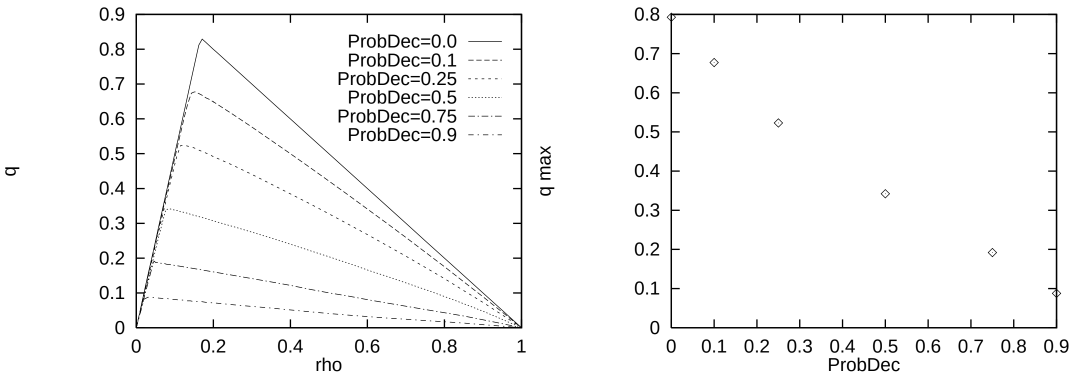

Abbildung 4.8: *Fluß im 2–spurigen symmetrischen Modell in Abhängigkeit von $Prob_d$*

Verzögerungswahrscheinlichkeit

Im einspurigen und zweispurigen Modell wurde bis jetzt jeweils nur ein Wert (0.5) als Verzögerungswahrscheinlichkeit benutzt. In der Netzsimulation ist es möglich, acht verschiedene Wahrscheinlichkeiten in einer Tabelle zu definieren und den Straßenabschnitten einen Index auf diese Tabelle zuzuordnen. In [4] sind einfache Korrekturfaktoren für q_{max}, v_{ϱ_c} und v_{free} in Abhängigkeit der Wetter– und Lichtbedingungen gegeben (siehe Tab. 4.2). Natürlich sind innerhalb des Modells des zellularen Automaten diese Parameter nicht un-

Parameter	dunkel	naß	dunkel + naß
q_{max}	1.00	0.85	0.85
v_{ϱ_c}	1.00	0.85	0.85
v_{free}	0.92	0.92	0.85[4]

Tabelle 4.2: *Korrekturfaktoren für äußere Bedingungen*

abhängig voneinander justierbar. Deswegen sollte am besten die Anpassung durch Vergleich von q_{max} mit Hilfe der Veränderung von $prob_d$ durchgeführt werden, da die Abhängigkeit $q_{max} = q_{max}(prob_d)$ sehr einfach zu handhaben ist. In Abb. 4.8 sind die Fundamentaldiagramme für verschiedene Verzögerungswahrscheinlichkeiten aufgetragen (links). Der Anteil an langsamen Fahrzeugen ist in allen Fällen null (einspurig siehe Abb. 2.1). Im rechten Diagramm ist die Abhängigkeit der Flußmaxima von $prob_d$ dargestellt.

Emissions– und Absorptionsraten

Für alle Endstücke läßt sich die Emissions– und Absorptionsrate individuell festlegen. Dies geschieht über einen Zeittakt von 5 [min], genannt *Endstück–Zeitschritt.* Jeweils am Anfang eines solchen Endstück–Zeitschrittes werden die aktuellen Raten aus einer Tabelle eingelesen. Sie gibt für jede Wunschgeschwindigkeit an, wie viele Fahrzeuge innerhalb der nächsten 5 Minuten emittiert bzw. absorbiert werden sollen.

[4] An dieser Stelle erwartet man einen Wert unter 0.85. Es handelt sich hierbei *nicht* um einen Übertragungsfehler. Vielmehr zeigt es, daß die Ermittlung von Flußdiagrammen in Abhängigkeit von Umwelteinflüssen bis jetzt stark vernachlässigt wurde. Alle Meßdaten beziehen sich auf ungefähre Wetterangaben und berücksichtigen nicht, daß sich die Wetterverhältnisse unter Umständen ebenso schnell entwickeln können wie die Verkehrsverhältnisse.

Es kann nun innerhalb der Simulation der Fall auftreten, daß die Strecke, die von einer Quelle wegführt, über einen längeren Zeitraum gesättigt ist, und deswegen de fakto weniger Fahrzeuge eingespeist werden, als die Emissionsrate vorgibt. Dies wird dadurch gelöst, daß am Ende des Endstück–Zeitschrittes die Zähler der noch nicht emittierten Fahrzeuge zu den Raten des neuen Zeitschrittes hinzuaddiert werden.

Transferwahrscheinlichkeit und Markierung des Transferwunsches

Wie schon beschrieben, können sich Fahrzeuge auf den zweispurigen Streckenabschnitten frei bewegen. Dies gilt jedoch nur, solange sie nicht in unmittelbarer Nachbarschaft eines Netzknotens sind. Nähert sich nun ein Fahrzeug einem solchen Punkt, so muß zu einer gewissen Zeit entschieden werden, wie sich das Fahrzeug im Bezug auf seine zukünftige Fahrtrichtung verhalten wird. Man unterscheidet drei Fälle:

Fall 1 An Ausfahrten und an Abzweigungen übergeordneter Strecken zur untergeordneter Strecke bei Dreiecken (siehe Abb. 4.6) muß festgelegt werden, welcher Anteil der Fahrzeuge abfährt. In diesem Fall reicht *eine* Transferwahrscheinlichkeit.

Fall 2 An Gabelungen der untergeordneten Strecke bei Dreiecken muß festgelegt werden, welcher Anteil nach links bzw. nach rechts fährt. Hier reicht ebenfalls ein Wert aus, der die Transferwahrscheinlichkeit nach rechts[5] angibt.

Fall 3 An Kreuzen müssen die Transferwahrscheinlichkeiten für beide Richtungen angegeben werden. Hierbei muß beachtet werden (siehe Abb. 4.7), daß die Markierung der rechts abbiegenden Fahrzeuge räumlich *vor* der Markierung der links abbiegenden Fahrzeuge erfolgt. Demnach ist eine Markierung für das Abbiegen nach links nur dann sinnvoll, wenn diese noch nicht für das Abbiegen nach rechts markiert worden sind oder bereits nach rechts abgefahren sind[6]. Also ist die Wahrscheinlichkeit für das Abbiegen nach links auf eine einfache Weise zu korrigieren: Wenn t_l der Anteil ist der links und t_r der Anteil der rechts abfährt, so ergeben sich die korrigierten Wahrscheilichkeiten T_l und T_r zu

$$T_r \quad = \quad t_r \qquad \text{und} \qquad T_l \quad = \quad \frac{t_l}{1 - t_r}.$$

Die eigentliche Markierung der Fahrzeuge läuft bei allen Fällen gleich ab. Dazu ist jedem Beschleunigungsstreifen und jeder Gabelung ein *Markierungsbereich* vorgeschaltet, der *MarkGap* entfernt liegt und selbst die Länge v_{max} hat. Da die maximale Wunschgeschwindigkeit der Fahrzeuge ebenfalls v_{max} beträgt, muß jedes Fahrzeug beim Vorbeifahren mindestens ein Site des Markierungsbereiches für einen Zeitschritt belegen. Kommt nun ein Fahrzeug auf ein solches Site, so wird es mit der vorgegebenen Transferwahrscheinlichkeit markiert. Diese Markierung heißt *Transferwunsch* und wird mit `LeftRoute` oder `RightRoute` bezeichnet. Damit aber überhaupt sichergestellt ist, daß die Fahrzeuge absorbiert werden können, muß ihr Spurwechselverhalten zusätzlich beeinflußt werden:

- Fahrzeuge der obigen Fälle 1 und 3 sowie des Falls 2 für den Transfer nach rechts müssen sich auf der *rechten* Spur befinden. Deswegen wird durch eine Markierung `ChangeRight` ein Spurwechsel nach links verboten und beim Spurwechsel nach rechts werden die Regeln $S1$:*Aktuelle Spur günstig?* und $S2$:*Ist andere Spur besser?* (siehe

[5] willkürliche Wahl

[6] Welcher der beiden Fälle zutrifft, hängt von der Größe *MarkGap* ab

3.2.3) ignoriert. Es ist möglich, auch zusätzlich die Regel $S3$ *Genügend Abstand zum Nachfolger?* so abzuschwächen, daß nur noch das benachbarte Site überprüft wird. Dieser Modus wird als *PanicLaneChange* bezeichnet.

- Fahrzeuge des Falls 2 für den Transfer nach links müssen sich auf der linken Spur befinden. Deswegen wird durch eine Markierung *ChangeLeft* ein Spurwechsel nach rechts verboten und beim Spurwechsel nach links werden die Regeln $S1$ und $S2$ (siehe 3.2.3) ignoriert. Wie oben ist auch hier der *PanicLaneChange* möglich.

Außer der Beeinflussung des Spurwechselverhaltens ist noch eine weitere Maßnahme notwendig. Es kann vorkommen, daß Fahrzeuge, die sich noch nicht auf der korrekten Spur befinden, wegen deren dichten Belegung nicht auf diese wechseln können. Nach dem normalen zweispurigen Modell würden sie solange auf ihrer Spur weiterfahren, bis sie den Transferbereich hinter sich gelassen haben, und somit trotz der Markierung `RouteLeft/RouteRight` tatsächlich nicht transferiert werden. Um Abhilfe zu schaffen, werden Fahrzeuge, die sich auf gleicher Höhe mit dem Ende eines Verzögerungsstreifens oder einer Gabelung auf der "falschen" Spur befinden, so abgebremst daß sie auf dem letzten Site zu stehen kommen, auf dem der Transfer noch möglich ist. Dort verbleiben sie so lange, bis sie die Spur wechseln können (siehe Schriftzug *Blockade für abfahrende Fahrzeuge* in Abb. 4.3).

4.4 Aufbereitung der Daten

Bevor die Simulation gestartet werden kann, müssen die Netzdaten, die als ASCII–Datei auf Platte liegen, im Rechner als Datenstruktur aufgebaut werden. Dazu sind mehrere Schritte notwendig, die im folgenden beschrieben werden.

4.4.1 Konsistenzüberprüfungen

Ein grundsätzliches Problem bei den heute vorliegenden Netzdaten ist die hohe Fehlerquote. Im allgemeinen sind die Daten Punkt für Punkt von Landkarten digitalisiert worden. Da diese Arbeit im hohen Maße die Konzentration, sind Tipp– und Positionierungsfehler nicht auszuschließen. Im günstigsten Fall handelt es sich nur um Buchstabenverdrehungen, die keine Beeinflußung des eigentlichen Netzes darstellen. Im Gegensatz dazu können Positionierungsfehler zu einem direkten Fehler im Netz führen, z.B. dann, wenn Straßenabschnitte an die falschen von dicht nebeneinander liegenden Netzpunkten angebunden werden. Später durchlaufen die Daten oft mehrere Selektions– und Konvertierungsschritte. Deswegen muß im allgemeinen angenommen werden, daß das Datenmaterial nicht konsistent ist und somit einer gründlichen Überprüfung bedarf. Nach dem Einlesen der Daten laufen nun im einzelnen die folgenden Schritte ab:

1. Überprüfung auf mehrfaches Auftreten der logischen Numerierung der Knoten und Kanten.
2. Überprüfung auf Fehlen von Knoten, die von Kanten referenziert werden.
3. Überprüfung auf Valenzen von null und Valenzen größer vier.
4. Überprüfung auf Null–Kanten (Startknoten ist gleich Zielknoten).
5. Überprüfung auf mehrfaches Auftreten von Kanten mit äquivalenten Start– und Zielknoten.

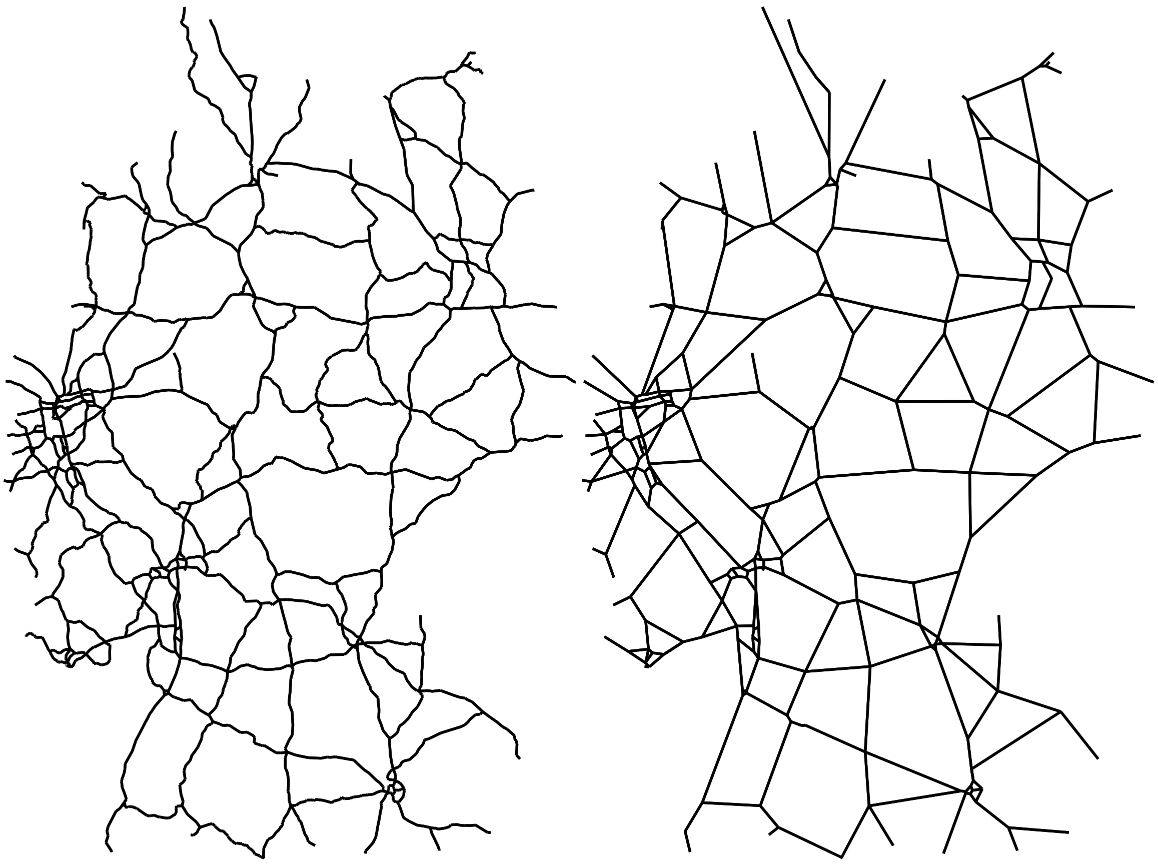

Abbildung 4.9: *Das Autobahnnetz der Bundesrepublik* — Auf der linken Seite ist das Originalnetz, auf der rechten Seite das reduzierte Netz abgebildet. Siehe dazu auch Abb. 4.10.

Fehlerhafte Datenelemente werden markiert und anschließend eine korrigierte, d.h. um die fehlerhaften Datenelemente gekürzte, Version des Datenbestandes abgespeichert. Die Simulation beginnt erst, wenn alle Konsistenzüberprüfungen erfolgreich verlaufen (siehe hierzu auch 4.4.3 und 4.4.4).

4.4.2 Datengrundlage

Die Netzdaten, die den Simulationen dieser Arbeit zugrunde liegen, wurden durch Extraktion aus dem Deutschlandnetz der Firma *Planungsbüro Transport und Verkehr GmbH* (kurz *PTV*), Karlsruhe gewonnen. Die Straßenabschnitte der Deutschlandkarte sind in 6 Ebenen (Level) eingeteilt, wobei ausgehend von der obersten (gröbsten) Ebene das Netz durch Hinzunahme von weiteren Ebenen immer feiner erweitert wird. Ausreichend für *PAMINA* war der Level 1. Er umfaßt sämtliche Autobahnen und zusätzlich Bundestraßen und Landstraßen, um den Zusammenhang des Netzes und eine homogene Abdeckung des Bundesgebietes zu garantieren. Damit in späteren Simulationen eine Anpassung der Straßeneigenschaften an Straßentypen möglich ist, sind die Straßenabschnitte der Inputdaten in Kategorien eingeteilt. Eine Übersicht befindet sich in Tab. 4.3. Die Eckdaten der Karten von Nordrhein–Westfalen und der Bundesrepublik sind in Tab. 4.4 zusammengefaßt. Die von PTV zur Verfügung gestellten Daten haben sich als so gut erwiesen, daß alle Kanten übernommen werden konnten.

Typ	Kategorie
0	keine Information verfügbar
1	Autobahn
2	Bundesstraße mit zweispuriger Fahrbahn
3	Bundesstraße
4	Landstraße

Tabelle 4.3: *Übersicht über Straßenkategorien*

Größe	NRW	BRD
Knoten	549	3307
Kanten	1,160	6,860
Endstücke	19	46
Autobahnanschlüsse	349	1568
Autobahndreiecke	39	176
Autobahnkreuze	21	58
Strecke [km] (pro Spur u. Richtung)	1,952	12,474
Sites einspurig	62,880	265,280
Sites zweispurig	1,041,204	6,653,016

Tabelle 4.4: *Umfang der Netze NRW und BRD*

4.4.3 Reduktion des Netzes

Die Knoten eines Straßennetzes lassen sich anhand ihrer Bedeutung für die Struktur in zwei Gruppen teilen. Siehe dazu Abb. 4.10.

- Ein Knoten, der entweder einen Autobahnanschluß darstellt (Valenz 2 mit Namen), oder lediglich den graphischen Straßenverlauf wiederspiegelt (Valenz 2 ohne Namen), kann einfach entfernt werden, indem die beiden adjazenten Kanten zu einer neuen Kante verschmolzen werden, deren Länge die Summe der Ausgangskanten ist. Das Netz hat nachher zwar einfachere Kanten–Verläufe zwischen den Netzknoten, die eigentliche Struktur ist jedoch gleichgeblieben.
- Ein Knoten, an dem Straßenabschnitte zusammentreffen (Valenz 3 und 4) oder aufhören (Valenz 1) ist für die Struktur des Netzes ausschlaggebend, denn nach Entfernen eines solchen Knotens — was automatisch die Entfernung aller adjazenten Kanten impliziert — hätte das Netz ein grundsätzlich anderes Aussehen. Es könnte sogar in mehrere Zusammenhangskomponenten zerfallen.

Durch sukzessives Löschen der Knoten der Valenz zwei ist somit eine Vereinfachung des Netzes möglich. Das neue daraus gewonnene Netz besteht dann aus *Superknoten* der Valenz 1, 3 und 4 und *Superkanten*, die nur zu Superknoten adjazent sind.

4.4.4 Zusammenhangstest

Nach der Konsistenzüberprüfung und der Netzreduktion wird ein Zusammenhangstest auf dem reduzierten Netz durchgeführt. Falls mehr als eine Zusammenhangskomponente gefun-

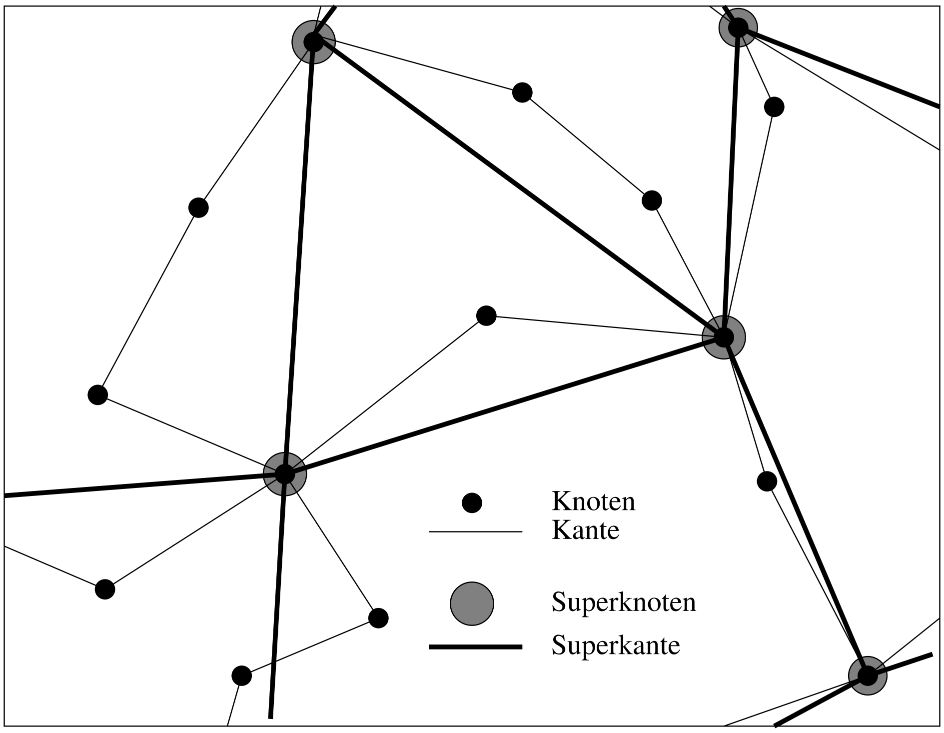

Abbildung 4.10: *Reduktion der Kanten zu Superkanten* — Es ist ein Teilausschnitt eines Netzes mit überlagertem reduziertem Netz zu sehen. Die Superknoten bilden eine Teilmenge der alten Knoten. Die Superkanten werden neu durch Verschmelzen der alten Kanten gebildet. Obwohl in diesem Bild nicht ersichtlich haben sie als Länge die Summe der ursprünglichen Einzellängen. Siehe dazu Abb. 4.9 als konkretes Beispiel.

den wird, werden alle Netzelemente, die nicht zur größten[7] Komponente gehören, markiert und bei Abspeichern des korrigierten Datenbestandes nicht mehr berücksichtigt. An dieser Stelle ist im Bezug auf die verwendeten PTV–Daten zu erwähnen, daß der Level 1 sich auch hier als fehlerfrei zusammenhängend erwiesen hat.

Durch iteratives Aufrufen des Programms und Ablauf der ersten drei Schritte *Konsistenzüberprüfung*, *Reduktion* und *Zusammenhangstest* wird schließlich eine Datengrundlage geschaffen, auf der die Simulation aufbaut.

4.4.5 Ermittlung der Transferwahrscheinlichkeiten

Die Ermittlung der Transferwahrscheinlichkeiten an Kreuzen, die von *PAMINA* als Input vorausgesetzt werden, erweist sich in der Realität als ernstzunehmendes Problem. Dies läßt sich am Beispiel eines Autobahnkreuzes demonstrieren. Oft liegen Meßstellen an Kreuzen nur für die einlaufenden und die auslaufenden Straßenabschnitte vor. Man erhält dadurch 4 Gleichungen (= Valenz des Knotens), hat jedoch 4×2 unbekannte Transferwahrscheinlichkeiten, insgesamt also ein unterbestimmtes Gleichungssystem. Im Beispiel in Abb. 4.11 sind die Verkehrsflüsse für einen Beobachter an den Positionen A, B, C und D äquivalent, obwohl sich die Ströme an der Kreuzung unterschiedlich aufteilen. Noch ungünstiger ist das Verhältnis von Unbekannten zu Gleichungen, wenn nur noch für Teilnetze die Daten über einfließenden und ausfließenden Verkehr verfügbar sind. Stellt man sich die Fläche F vor,

[7]berechnet aus der Gesamtlänge aller enthaltenen Kanten

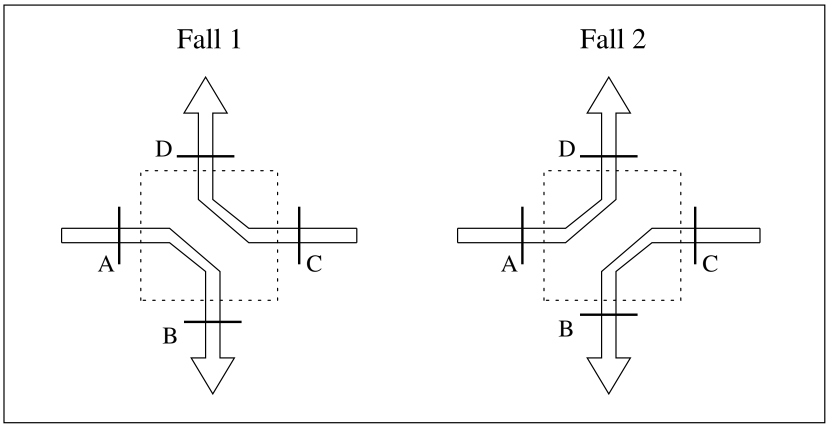

Abbildung 4.11: *Ermittlung der Transferwahrscheinlichkeiten* — Die Verkehrsflüsse sind für einen Beobachter an den Positionen A, B, C und D äquivalent für beide Fälle, obwohl sich die Ströme an der Kreuzung unterschiedlich aufteilen.

die von einem solchen Teilnetz eingenommen wird, so wächst bei angenommener homogener Struktur des Netzes die Anzahl der Netzknoten linear mit der Fläche. Die Anzahl der Bestimmungsgleichungen, die sich über Kanten durch den Rand der Fläche ermitteln lassen, jedoch nur mit $\sqrt{F}$. Die Anzahl der Gleichungen wächst also demnach viel langsamer als die Anzahl der Unbekannten. Deshalb ist es notwendig, sich eine wichtige Eigenschaft des Verkehrsflusses zunutze zu machen: das Verkehrsaufkommen ist stark zeitabhängig. Wenn man die zeitlichen Verzögerung der Fahrzeuge bei der Durchfahrt durch eine Teilnetz von einem Eintrittspunkt zu allen Austrittspunkten kennt, so kann ein *Mustervergleich* der Dichte–Zeit–Kurven stattfinden, wodurch sich zusätzliche Bestimmungsgleichungen ergeben (siehe z.B. [7] und [9]).

Da bei Abschluß dieser Arbeit noch keine flächendeckenden Daten über Abbiegewahrscheinlichkeiten und Fahrzeugaufkommen vorlagen, hat die besprochene Problematik keine direkten Folgen auf die in Abschnitt 5 besprochenen Ergebnisse der Netzsimulationen. Diese wurden mit Hilfe von kanonischen Randbedingungen und Parametern gewonnen, bei denen alle Knoten des Netzes gleichberechtigt waren.

4.5 Parallelisierung

Nach den Schritten *Konsistenzüberprüfung*, *Reduktion* und *Zusammenhangstest* werden die Netzelemente auf die zur Verfügung stehenden CPNs[8] des Parallel–Computers verteilt. Die Vorgehensweise wird in den nächsten Abschnitten erläutert.

4.5.1 Loadbalancing und Kommunikation

Die Verteilung der Knoten und Kanten auf ein System sollte möglichst einfach im Ansatz sein, denn die Hauptaufgabe der Simulation besteht in der schnellen Berechnung der Fahrzeuge im Netz und nicht in der optimierten Aufteilung an sich, was eher in den Bereich der Kombinatorik fällt. Andererseits sollte so viel Aufwand investiert werden, daß

[8] engl. für Computational Node, enspricht einer Recheneinheit eines Parallel–Rechners

die zur Verfügung stehende Hardware effizient genutzt wird. Zwei Faktoren beeinflußen hauptsächlich die Leistungsfähigkeit eines Programms auf Parallel–Rechnern:

Loadbalancing Die Dynamik der Simulation läuft in Zeitschritten ab. Eine CPN kann erst den Zeitschritt n bearbeiten, wenn sie alle Informationen von ihren Nachbarn über den Zeitschritt $n-1$ besitzt. Demnach nützt es nichts, wenn manche CPNs ihren Teilbereich des Netzes schneller bearbeiten als andere, denn das führt nur zu Wartezeiten am Ende von Zeitschritten. Die Gesamtheit aller CPNs richtet sich also nach der CPN, die für ihren Teilbereich die meiste Zeit braucht. Deswegen ist es wichtig zu erkennen, welche Komponenten der Simulation den größten Rechenaufwand darstellen, und die Verteilung auf CPNs bezüglich dieser Komponenten durchzuführen. Diese Aufgabe wird *Loadbalancing* genannt. Wird sie einmal vor Beginn der Simulation durchgeführt, so bezeichnet man sie als *statisches Loadbalancing* im Gegensatz zum *dynamischen Loadbalancing*, bei dem während der Simulation die Aufteilung geändert wird.

Kommunikation Während der Simulation werden zwischen den CPNs Nachrichten ausgetauscht, in denen der Status der Randbereiche (Boundaries) codiert enthalten ist. Um hohe Geschwindigkeiten zu erreichen, muß sowohl die Anzahl als auch das Volumen der Nachrichten minimiert werden.

4.5.2 Geometrische Verteilung

In *PAMINA* wurde als Ansatz für die Parallelisierung ein geometrisches, statisches Loadbalancing gewählt. Dabei werden zuerst jedem Superknoten die Summe der halben Längen aller adjazenten Kanten zugeordnet. Alle Superknoten bilden anfänglich ein Segment[9], das dem kleinsten, das Netz umfassenden Rechteck entspricht. Anschließend wird dieses Segment solange in Teilsegmente unterteilt, bis deren Anzahl gleich der Anzahl der zur Verfügung stehenden CPNs ist. Dazu werden für eine Teilung die folgenden Schritte durchgeführt:

1. Ermittle das Segment, das die größte Gesamtstraßenlänge hat. Dadurch wird gewährleistet, daß dasjenige Segment zerteilt wird, das den größten Anteil an der Berechnungszeit hat, und somit das Loadbalancing am ehesten verbessert wird.

2. Sortierung der Superknoten nach X–Koordinaten (bzw. Y–Koordinaten) dieses Segmentes. Hierdurch wird sichergestellt, daß im Netz zusammenliegende Knoten auch auf den CPNs zusammenliegen und somit die Anzahl der Boundaries reduziert wird, was seinerseits zu einer geringen Anzahl von zu verschickenden Nachrichten führt.

3. Unterteilung der Superknoten in zwei Teilmengen, so daß die absolute Differenz der Gesamtstraßenlängen der beiden Knotenmengen möglichst klein ist. Bei Testläufen hat sich herausgestellt, daß die Aktualisierung der Fahrzeuge auf den Fahrbahnen ca. 85% bis 95 % der gesamten Rechenzeit ausmacht. Der Rest entfällt auf die Verwaltung der Netzelemente und der Boundaries. Somit ist die Anzahl der Netzknoten, die einer CPN zugeordnet werden, nebensächlich. Ausschlaggebend ist die Gesamtstraßenlänge.

4. Bilde aus den beiden Knotenmengen die neuen Segmente. Falls eines der beiden Segmente zu einem späteren Zeitpunkt nochmal einmal geteilt wird, so wird nach der

[9] In einigen Publikationen werden *Kanten* als Segmente bezeichnet. Hier wird der Begriff Segment aber für eine *Fläche* verwendet

jeweils anderen Koordinate sortiert. Dadurch bleiben die Seitenlängen der Segmente ungefähr gleich lang und es kommt zu keiner Unterteilung in dünne Scheiben.

Man erhält als Ergebnis eine Parkettierung des Netzgebietes, wobei die einzelnen Kacheln unterschiedliche Ausmaße haben. Sie sind groß in Gebieten mit einer geringen Straßendichte und klein in Ballungszentren.

Jede CPN übernimmt die Verwaltung aller ihr zugeordneten Netzknoten und deren Funktionen. Diese bestehen aus

- dem Update der Fahrzeuge auf den Pfaden,
- dem Update der Fahrzeuge auf den Transferstrecken sowie
- der Emission and Absorption an Endstücken und den Beschleunigungs– und Verzögerungsstreifen.

Die Wahl eines statischen Loadbalancings läßt sich dadurch rechtfertigen, daß die gitterorientierte Implementation der Fahrzeugdynamik (siehe 6.1.1) im Vergleich zur listenorientierten relativ unabhängig von der Belegungsdichte der Straßen ist. Dadurch halten die CPNs, die Netzbereiche mit Ballungszentren bearbeiten, zu Stoßzeiten weitgehend ihre Geschwindkeit bei.

4.5.3 Verwaltung der Boundaries

Nachdem die Netzknoten auf die CPNs verteilt worden sind, müssen die Superkanten ebenfalls den CPNs zugeordnet werden. Dabei gibt es zwei Fälle:

- Beide Superknoten der Superkante liegen auf der gleichen CPN. In diesem Fall übernimmt diese CPN die Berechnung der kompletten Superkante.
- Die Superknoten der Superkante liegen auf unterschiedlichen CPNs. In diesem Fall wird die Superkante an derjenigen Kante geteilt, die möglichst genau in der Mitte zwischen den beiden CPNs liegt. Jede CPN übernimmt die jeweils näherliegende Hälfte. Dadurch entstehen an den Übergängen Boundaries, die als Nachrichten verschickt werden.

Falls ein Paar von Knoten mehrere Boundaries gemeinsam hat, so werden diese zu einem *BoundarySet* zusammengefaßt. Dadurch erhält jede CPN während der Simulation nur eine Nachricht pro Zeitschritt und pro Nachbar[10].

Durch die trotz Loadbalancing auftretende ungleichmäßige Belastung der einzelnen CPN kann es vorkommen, daß eine CPN sowohl die Boundaries für den aktuellen Zeitschritt als auch schon für den nächsten Zeitschritt von einem Nachbarn empfängt. Siehe dazu Abb. 4.12. In diesem Beispiel ist ein Netz mit drei CPNs dargestellt, wobei CPN A mit CPN B und CPN B mit CPN C verbunden ist. CPN C benötigt erheblich mehr Zeit als die anderen beiden für den ersten Zeitschritt. Dadurch wird CPN B blockiert. Einen Zeitschritt später wird auch CPN A blockiert. CPN B erhält zum Zeitpunkt t_1 eine Boundary von A, die noch nicht verwendet werden kann, weil die Boundary von CPN C fehlt. Zum Zeitpunkt t_2 erhält CPN B eine weitere Boundary. Also müssen Puffer für zwei Boundaries

[10] Zwei CPNs sind Nachbarn, wenn sie mindestens eine gemeinsame Boundary besitzen

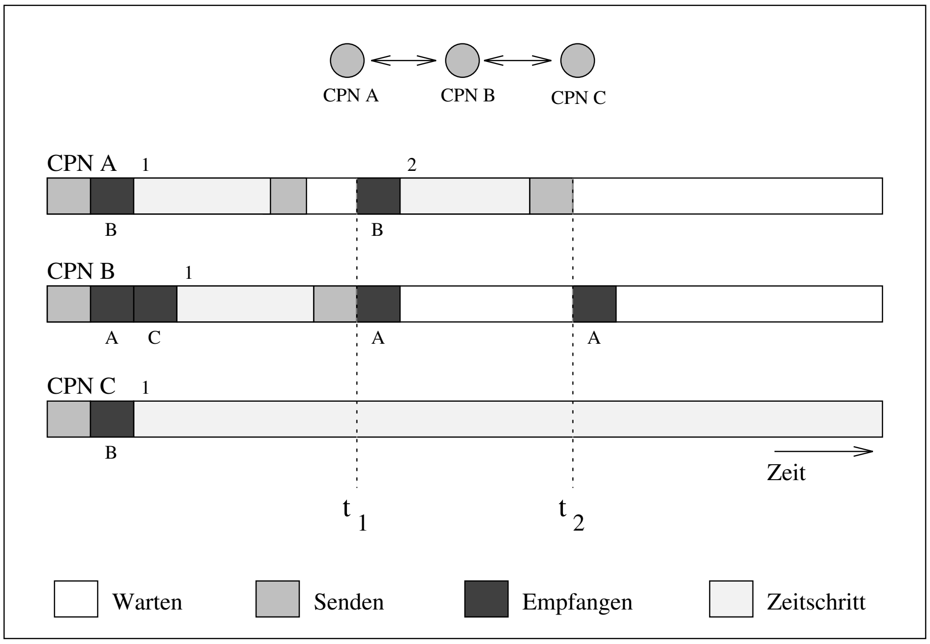

Abbildung 4.12: *Pufferung von Boundaries* — Das Diagramm zeigt das Entstehen eines Zeitschrittgefälles aufgrund einer langsamen Bearbeitung des Zeitschritts 1 (Ziffern oberhalb Balken) durch CPN C. Die Buchstaben unterhalb der Balken geben den Absender der empfangenen Nachricht an.

pro Nachbar eingerichtet werden. Der Effekt des Zeitschrittgefälles in verteilten Netzen wurde bereits von Nagel und Schleicher [24] erwähnt.

Unter Umständen muß also eine Generation von Boundaries zwischengespeichert werden, bis die letzte für den aktuellen Zeitschritt notwendige Boundary eintrifft, und der nächste Zeitschritt durchgeführt werden kann. Andererserseits brauchen auch nicht *mehr* als zwei Boundaries für einen Knoten A gespeichert werden, denn die dritte Boundary (vom gleichen Nachbarn B) kann erst dann ankommen, wenn B einen weiteren Zeitschritt durchgeführt hat, was aber nicht möglich ist, da A bereits blockiert ist und B die Boundary von A zum Zeitschritt braucht.

Bei Beginn der Simulation kennt jede CPN jeden ihrer Nachbarn $N_1 \dots N_n$ und damit die Anzahl n der zu empfangenden Boundaries pro Zeitschritt. Es kann nun auf die folgenden zwei Arten vorgegangen werden:

- Jede CPN begibt sich in eine Warteschleife, in der sie zuerst auf die Boundary von N_1 wartet, anschließend in diejenige für N_2 und so weiter. Dabei ist für die Bearbeitungsgeschwindigkeit hauptsächlich die Ankunft der Nachricht von N_1 ausschlaggebend. Dieses Verfahren ist starr gegenüber der unterschiedlichen Bearbeitungszeiten auf den CPNs, da eine CPN gegenüber den anderen Nachbarn ausgezeichnet ist. Es hat aber den Vorteil, daß durch die systematische Abarbeitung der Nachbarn ewaige schon empfangene Boundaries von zukünftigen Zeitschritten erst im jeweils nächsten Durchlauf berücksichtigt werden. Das Zeitschrittgefälle wird so automatisch gelöst.
- Um das Warten auf den ersten Nachbarn zu vermeiden, begibt sich jede CPN in eine Warteschleife, in der sie auf eine Boundary eines beliebigen Nachbarn wartet. Bei jedem Eintreffen einer Boundary wird diese bearbeitet und erneut die Warteschleife angesprungen, bis alle für den Zeitschritt notwendigen Boundaries empfangen

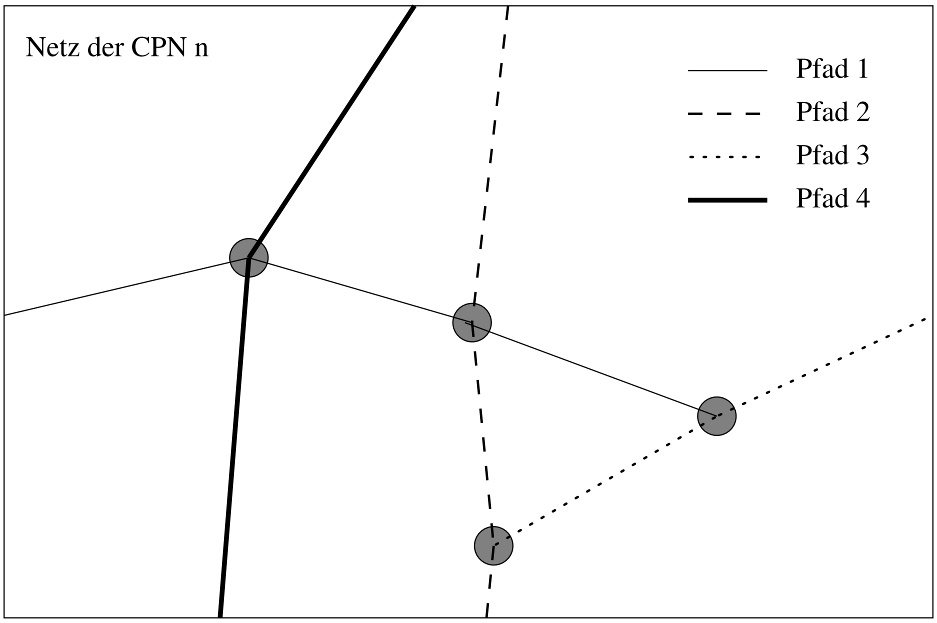

Abbildung 4.13: *Zusammenfassen der Superkanten zu Pfaden* — Es ist ein reduzierter Teilgraph des Netzes zu sehen. Die Superkanten sind zu Pfaden zusammengefaßt, die möglichst dem geradlinigen Straßenverlauf folgen.

wurden. In dieser Implementation wird demnach die *erste* Wartezeit verkürzt oder bleibt höchstens gleich der vorherigen Implementation. Sie wurde deshalb für *PAMINA* gewählt. Allerdings müssen hier empfangene Boundaries zukünftiger Zeitschritte gepuffert werden.

In Tab. 4.5 ist der Aufwand, der durch die Aufteilung auf CPNs entsteht, für das Netz von NRW zusammengefaßt. Es zeigt sich, daß die einfache geometrische Aufteilung ausreichend für die Problematik ist. Die Anzahl der Boundaries pro CPN bleibt in annehmbaren Grenzen zwischen 6.1 und 9.5, wobei der kleinste Wert von 6.1 sogar bei der größten Anzahl von CPNs auftritt.

# CPN	# Boundaries				# BoundarySets		
	min	max	Summe	# pro CPN	min	max	Summe
2	8	8	16	8.0	1	1	2
4	7	11	38	9.5	2	3	10
8	4	14	61	7.6	2	5	26
16	2	12	98	6.1	2	6	58

Tabelle 4.5: *Aufwand an Boundaries (NRW)*

4.5.4 Zusammenfassen der Superkanten zu Pfaden

Durch die Reduktion der Kanten auf Superkanten wurde das Netz bereits vereinfacht. Nach der Verteilung der Superkanten auf CPNs erfolgt eine weitere Zusammenfassung. Wie in den Abschnitten über Netzelemente erläutert, gibt es bei Autobahndreiecken und Autobahnkreuzen Straßenabschnitte mit durchgehender Verkehrsführung: bei Dreiecken ein Paar übergeordneter Straßenabschnitte und bei Kreuzen zwei Paare gleichberechtigter Straßenabschnitte. Diese Eigenschaft wird in *PAMINA* so ausgenutzt, daß alle Superkanten, die auf dem gleichen durchgehenden Straßenverlauf liegen, zu einem *Pfad* zusammengefaßt werden. Ein Pfad beginnt und endet demnach entweder in einer Boundary oder in dem untergeordneten Straßenabschnitt eines Autobahndreiecks (siehe Abb. 4.13). Dadurch wird die Anzahl zu betrachtender Netzabschnitte noch einmal beträchtlich reduziert. Tab. 4.6 enthält eine Aufstellung der Anzahl der Superkanten und Pfade für verschiedene Anzahl von CPNs. Grundlage war das Netz von NRW mit 549 Knoten und 1160 Kanten.

# CPN	# Superkanten			# Pfade		
	min	max	Summe	min	max	Summe
2	108	128	236	36	38	74
4	40	70	258	22	26	96
8	22	56	266	12	20	124
16	6	32	322	4	16	160

Tabelle 4.6: *Reduktion des Datenbestandes (NRW)*

4.6 Ablauf der Simulation

4.6.1 Individualmodus (Run Mode 0)

Im *Individualmodus* sind alle Straßenabschnitte und Transferabschnitte anfänglich unbelegt. Eine Füllung des Systems mit Fahrzeugen geschieht ausschließlich über die individuell definierten Randbedingungen an Endstücken und Ausfahrten. Dieser Modus soll später der Simulation des rellen Verkehrsflusses innerhalb eines Tagesablaufs dienen. Leider lagen bei Fertigstellung der Diplomarbeit keine ausreichenden Informationen über das Verkehrsaufkommen und das Abbiegeverhalten für Nordrhein–Westfalen vor, so daß kein Testlauf durchgeführt werden konnte.

4.6.2 Modus mit vorgegebenen Dichten (Run Mode 1)

Der *Modus mit vorgegebenen Dichten* dient der theoretischen Untersuchung eines Netzes. Dazu kann das System gezielt mit einer Dichte an Fahrzeugen vorbelegt werden. Anschließend werden in regelmäßigen Intervallen insgesamt *Relaxationsabschnitte* viele Statistiken eingeholt. Dabei dient die Größe *DecaySteps* zur Angabe der Zeitschritte, die verworfen werden sollen, bevor zum ersten mal Meßdaten ermittelt werden. Dies ist notwendig, damit das System von den Anfangsbedigungen, bei denen die Geschwindigkeiten aller Fahrzeuge auf null gesetzt sind, zu einer normalen Geschwindigkeitsverteilung gelangen kann. Ein gesamter Ablauf sieht folgendermaßen aus:

FOR Alle gewünschten Dichten ϱ_i **DO**
  BEGIN
  Fülle das System homogen mit Dichte ϱ_i
  Warte *DecaySteps* Zeitschritte
  FOR Anzahl von Relaxationsabschnitten **DO**

```
        BEGIN
        Führe TimeSteps Zeitschritte mit regelmäßiger Statistik aus
        Schreibe die Statistik eines Abschnitts auf Datei
        END
    END
```

Algorithmus 2: Ablauf des Run Mode 1

Bei diesem Modus wird die Anzahl der Fahrzeuge und damit die Dichte *konstant* gehalten. Zu diesem Zweck sind alle Anschlußstellen deaktiviert. Zusätzlich werden Fahrzeuge, die das System an den Senken der Endstücke verlassen, sofort mit der gleichen Geschwindigkeit *reflektiert*. Für die Transferwahrscheinlichkeiten werden für alle Netzpunkte die gleichen Werte angenommen, so daß das Einlesen individueller Werte entfällt.

4.6.3 Modus für Geschwindigkeitsmessungen (Run Mode 2)

Beim *Modus für Geschwindigkeitsmessungen* wird das System wie im Run Mode 1 mit verschiedenen Dichten belegt. Für jede Dichte werden *TimeSteps* Zeitschritte berechnet *ohne* die Statistik durchzuführen. Anschließend werden wiederum *TimeSteps* Schritte durchgeführt, allerdings mit Statistikaufnahme in regelmäßigen Intervallen. Dadurch wird der Anteil, den die Statistik am gesamten Rechenaufwand hat, abschätzbar.

4.6.4 Modus für Teilnetz–Messungen (Run Mode 3)

Bei den ersten drei Modi bezogen sich ermittelte Statistiken jeweils auf das gesamte System. Es ist jedoch besonders im Netz aufgrund seiner inhomogen Struktur interessant, Teilbereiche gesondert zu betrachten. Zu diesem Zweck können *PAMINA* bis zu 8 Listen von Kantennummern übergeben werden, für die jeweils eine eigene Statistikdatei während der Simulation abgelegt wird. Dadurch ist die Untersuchung von Relaxation von Dichte, Fluß und Durchschnittsgeschwindigkeit auf ausgesuchten Abschnitten möglich. Ansonsten ist der Ablauf äquivalent zum dem des Run Mode 1. In 5.2 ist dieser Modus anhand von zwei Beispielen untersucht worden.

4.6.5 Aufgabenverteilung der CPN

Die CPN 0 hat zusätzliche Aufgaben, die sie von den restlichen CPNs unterscheidet. Sie führt die geometrische Aufteilung des Straßennetzes und den Transfer der aufgeteilten Daten von den Speichermedien zu den CPNs durch. Sobald die Simulation startet, wird zwischen zwei Modi unterschieden:

- Im *Steuerungsmodus* warten alle CPNs auf die Anweisungen der CPN 0. Dazu gehören zum Beispiel:
  - Starten einer Simulationssequenz[11]
  - Einholen einer Statistik über die aktuelle Belegung der Straßen.
  - Einholen einer Statistik über bisher an Endstücken absorbierte und emittierte Fahrzeuge.
  - Abbruch der Simulation

[11] Hintereinanderausführung einer gegebenen Anzahl von Zeitschritten

- Im *Simulationsmodus* verhält sich die CPN 0 wie die anderen CPNs und wartet jeweils auf die Boundaries der Nachbarn. Zusätzlich kann die CPN 0 noch folgende Nachrichten emfangen:
  - Graphikinformationen über einzelnen Kanten zur Darstellung in einer Gesamtgraphik.
  - Beliebige Textnachrichten (z.B. über Fehlerzustände)
  - Anforderung, die Simulation wegen nichtbehebbaren Fehlern abzubrechen (z.B. *Segmentation Violation*[12]).

4.6.6 Ablauf eines Zeitschrittes

Innerhalb eines Zeitschrittes im Simulationsmodus laufen auf jeder CPN die folgenden Schritte ab:

1. *Verschicken der Boundary–Sets an alle Nachbarn.*
2. *Warten auf die Boundary–Sets aller Nachbarn.*
3. *Laden der internen Boundaries an den Gabelungen der Autobahndreiecke.*
   An den Übergangsstellen der Gabelungen zwischen zweispurigen Straßenabschnitten und einspurigen Transferabschnitten existieren Bereiche, die als *interne Boundaries* bezeichnet werden. Da die Aktualisierung der Fahrzeugpositionen in den einspurigen und zweispurigen Abschnitten zu versetzten Zeiten innerhalb eines Zeitschrittes auftreten, müssen wie bei den Boundaries zwischen CPNs die Zustände der Bereiche ausgetauscht werden (siehe Abb. 4.6).
4. *Aktualisierung der zweispurigen Straßenabschnitte.*
5. *Durchführung der Absorptionen und Emissionen an Endstücken.*
6. *Auflösen der Deadlocks an Autobahndreiecken.*
7. *Durchführung der Absorptionen an Verzögerungsstreifen und Emissionen an Beschleunigungsstreifen sowie Aktualisierung der einspurigen Transferabschnitte.*
8. *Alle außer CPN 0: Verschicken von graphischen Informationen*
   In regelmäßigen Intervallen kann optional eine Graphik ausgegeben werden, die die aktuelle Dichte, Durchschnittsgeschwindigkeit oder den Durchschnittsfluß farblich darstellt. Dazu verschicken alle CPNs (außer CPN 0) die Belegung ihrer Kanten an die CPN 0.
   Nur CPN 0: Darstellung der Graphik
   Hier werden, wenn nötig, Anfragen (Events) vom Graphiksystem bearbeitet.

4.6.7 Ermittlung von Statistiken

Bisher können von der CPN 0 zwei Statistiken angefordert werden. Die Informationen der **Fahrzeugstatistik** sind in Tab. 4.7 aufgeführt. Jeder der Werte ist auch aufgespalten nach linker zweispuriger Fahrbahn, rechter zweispuriger Fahrbahn und Transferstrecken verfügbar. Falls die Dichte oder einer der Durchschnittswerte nicht berechenbar ist, wird

[12] Zugriff auf unerlaubte Speicherbereiche

Symbol	Beschreibung
L	Anzahl der Sites
L_{min}, L_{max}	Minimale/maximale Anzahl von Sites auf den CPNs
N	Anzahl der Fahrzeuge
N_{min}, N_{max}	Mininale/maximale Anzahl von Fahrzeugen auf den CPNs
$\sum_{i=1}^{N} v_i$	Summe der Fahrzeuggeschwindigkeiten
$\sum_{i=1}^{N} v_i^2$	Summe der Quadrate der Fahrzeuggeschwindigkeiten
ϱ	Dichte der Fahrzeuge
v_{mean}	Durchschnittsgeschwindigkeit
ϱv_{mean}	Durchschnittsfluß

Tabelle 4.7: *Fahrzeugstatistik*

der Wert -1 zurückgeliefert. Die Informationen der **Endstückstatistik** sind in Tab. 4.8 aufgeführt. Jeder der Werte ist auch aufgespalten nach den 5 Wunschgeschwindigkeiten verfügbar.

Symbol	Beschreibung
n_e	Anzahl emittierter Fahrzeuge
n_a	Anzahl absorbierter Fahrzeuge

Tabelle 4.8: *Endstückstatistik*

Durch Kombination der beiden Statistiken ist auf eine einfache Weise ein Plausibilitätstest möglich, in dem die aktuelle Gesamtanzahl der Fahrzeuge mit der Differenz von absorbierten und emittierten Fahrzeugen verglichen wird.

Die Ermittlung von Statistiken geschieht in drei Schritten. Im ersten schickt CPN 0 die Aufforderung an alle CPNs, *lokal* ihre Statistik zu ermitteln. Anschließend wird eine zweite Nachricht verschickt, in der alle CPNs ihre lokalen Daten zur *globalen* Statistik akkumulieren. Im dritten Schritt werden werden die Durchschnittswerte berechnet.

Vor Einholung der Statistik werden die einzelnen CPNs *nicht* synchronisiert, d.h. daß die Statistiken der CPNs unter Umständen zu anderen Zeitschritten gehören (siehe auch 4.5.3). Dies ist aber nicht weiter schlimm, da sich die Geschwindigkeitsverteilung innerhalb des auftretenden Zeitschrittgefälles nur geringfügig ändert und der Fehler, der durch falsch gezählte Fahrzeuge an den Boundaries entsteht, als vernachlässigbar klein anzusehen ist. Beim Netz von Nordrhein–Westfalen wurde der Effekt bei 110.000 Fahrzeugen (Dichte $\varrho = 0.1$) überprüft und ergab einen Fehler in der Messung von maximal 20 Fahrzeugen.

4.6.8 Graphik

Für die graphische Ausgabe des Netzes kann durch bedingte Compilierung ein Modul eingebunden werden, das eine Vektorgraphik des Netzes auf einem X–Windows–Display erzeugt. Dieses Modul ist in C++ implementiert und kann zum Beispiel mit dem GNU–C++–Compiler compiliert werden. Bei aktiver Graphik erscheint ein Fenster, in dem der Verlauf $q(t)$ für das gesamte System dargestellt wird, sowie eine Übersichtsgraphik des Netzes. Die Kanten können auf Wunsch in Abhängigkeit

- des Typs (Autobahn, Bundestraße, usw.),

Name	Wert [Sites]	Wert [Meter]	siehe
`MergeWidth`	10	75	Abb. 4.5 - Abb. 4.7
`SetMarkGap`	20	150	Abb. 4.5 - Abb. 4.7
`TurnLeftGap`	0	0	Abb. 4.7
`TurnRightGap`	15	112.5	Abb. 4.7
`TurnGap`	5	37.5	Abb. 4.6
`TransferLength`	40	300	Abb. 4.6, Abb. 4.7
`TerminalTransferLength`	20	150	Abb. 4.5
`RampGap`	10	75	Abb. 4.5

Tabelle 4.9: *Parameter der Netzsimulation*

- der lokalen Dichte,
- der lokalen Durchschnittsgeschwindigkeit oder
- des lokalen Flusses

eingefärbt werden.

4.6.9 Verwendete Parameter

Für die Simulationen wurden durchgehend abgesehen von der Dichte und dem Anteil an langsamen Fahrzeugen die gleichen Parameter verwendet. Sie sind in Tab. 4.9 aufgeführt. Die Transferwahrscheinlichkeiten an Autobahnkreuzen beträgt 5% für das Abbiegen nach links und ebenso viel für das Abbiegen nach rechts. Bei Autobahndreiecken beträgt die Wahrscheinlichkeit an der Gabelung zu den übergeordneten Abschnitten 50%, an den Abbiegespuren zu den untergeordneten Abschnitten auch jeweils 50%. Da sämtliche Simulationen mit reflektierenden Randbedigungen und konstanter Gesamtdichte durchgeführt wurden, sind die Emissionsraten und Absorptionsraten an allen Autobahnanschlüssen null. In der Realität fallen die Längen und Abstände sehr unterschiedlich aus. Es gibt kaum zwei identische Knotenpunkte. In [34] findet man eine grobe Zusammenstellung der charakteristischen Merkmale von Autobahndreiecken und –kreuzen. Für Autobahnkreuze liegen zum Beispiel die Radien für die Linksabbiegerrampen (genannt *Schleifenrampen*) zwischen 40 und 70 Metern (entspricht Transferlängen zwischen 188 und 330 Metern), für die Rechtsabbiegerrampen (genannt *Direkte Rampen*) zwischen 60 und 300 Metern (entspricht Transferlängen zwischen 94 und 471 Metern). Die angenommene Länge von 300 Metern ist also durchaus realistisch, wobei aber zu überlegen ist, ob in einer späteren Version bei den Transferlängen zwischen Links– und Rechtsabbiegern unterschieden werden sollte.

4.7 Geschwindigkeit

Die Geschwindigkeit der Simulation wurde mit Hilfe des Netzes von NRW auf drei verschiedenen Rechnersystemen bestimmt. Dazu wurden jeweils 200 Zeitschritte für 10 verschiedene Dichten und bei unterschiedlicher Anzahl von CPNs einmal mit Statistik (jeden 20. Zeitschritt) und einmal ohne Statistik berechnet. Die Ergebnisse für Dichte $\varrho = 0.066$ sind in Tab. 4.10 zu finden. Eine graphische Übersicht über alle Dichten und damit einen Überblick über die Dichteabhängigkeit der Rechengeschwindigkeit ist in Abb. 4.14 zu sehen.
Zusätzlich wurden die Geschwindigkeitsmessungen für das gesamte Autobahnnetz der Bun-

Rechner	CPUs	CPNs	MUPS	MUPS/CPU	Echtzeitverhältnis
486-DX 2 66MHz	1	2	0.61	0.61	0.55
Sparc 10	2	2	2.36	1.18	3.05
Paragon	2	2	1.77	0.88	1.60
Paragon	4	4	3.32	0.83	3.01
Paragon	8	8	6.37	0.80	5.77
Paragon	16	16	11.01	0.69	10.00

Tabelle 4.10: *Rechengeschwindigkeit bei Netz NRW*

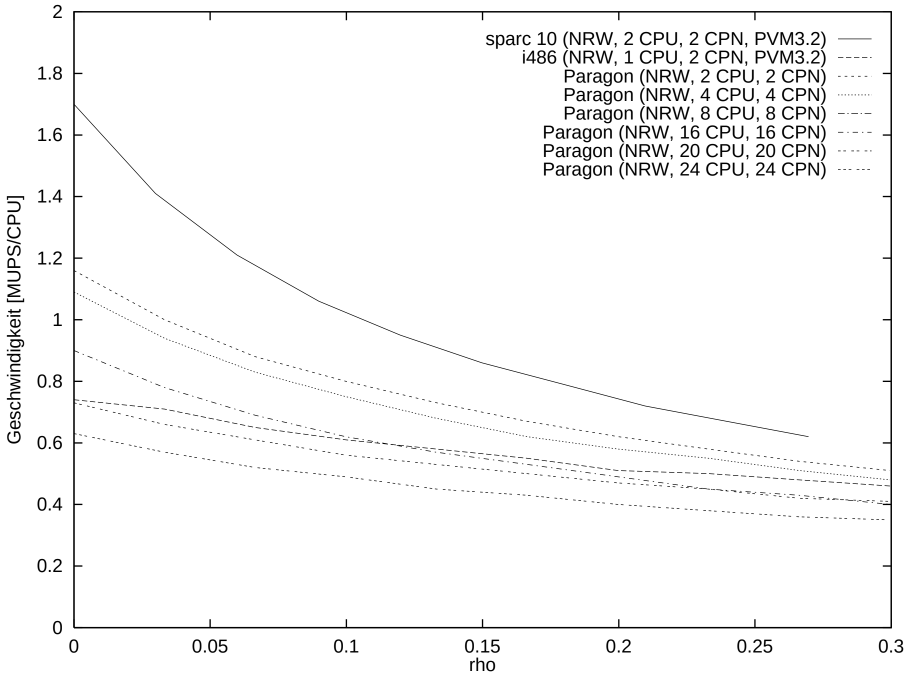

Abbildung 4.14: *Dichteabhängigkeit der Rechengeschwindigkeit (NRW)*

desrepublik durchgeführt. Sie beziehen sich wieder auf eine Dichte von $\varrho = 0.066$. Die Ergebnisse befinden sich in Tab. 4.11. Mit Hilfe von Run Mode 2 wurde der Einfluß der

Rechner	CPUs	CPNs	MUPS	MUPS/CPU	Echtzeitverhältnis
Paragon	16	16	14.02	0.88	2.1
Paragon	32	32	26.34	0.82	4.0
Paragon	64	64	38.45	0.60	5.8

Tabelle 4.11: *Rechengeschwindigkeit bei Netz BRD*

Statistik auf die Rechengeschwindigekeit untersucht. In Abb. 4.15 sind die Verluste durch das Einholen der Statistik jeden 20. Zeitschritt in Abhängigkeit der CPN–Anzahl dargestellt. Man erkennt, daß selbst für größere CPN–Anzahl der Aufwand nie über 4% steigt, so daß die Statistik als unbedeutend für den Zeitablauf eingestuft werden kann.

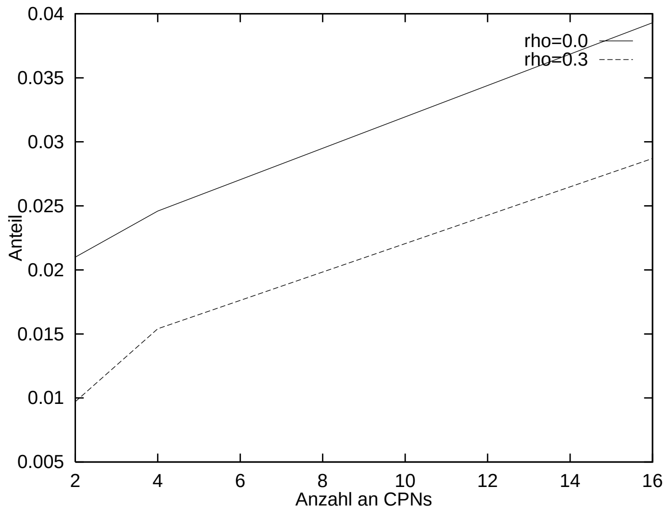

Abbildung 4.15: *Verluste durch Einholen der Statistik*

4.8 Probleme

4.8.1 Loadbalancing

Beim Verteilen des Netzes auf die CPNs tritt ein Problem auf, sobald die Anzahl der CPN vergleichbar wird mit der Anzahl der Superknoten. Durch die starke Fragmentierung fallen aufgrund der einfachen Regel zur mittigen Trennung von Superkanten die Anzahl der Sites auf den CPNs nicht gleich aus. Unter Umständen beträgt das Verhältnis vom kleinsten zum größten Wert ca. 2. Dadurch wird die potentielle Rechenkapazität halbiert (siehe zum Beispiel die Werte für MUPS/CPN in Tab. 4.10 für verschiedene CPN Anzahl). Durch einen intelligenteren Schnitt der Superkanten könnte demnach etwas Effizienz gewonnen werden.

4.8.2 Kurze Kanten

Alle Netzknoten haben durch die Anknüpfung von Beschleunigungs– und Verzögerungsstreifen gewisse Ausmaße, sind also nicht mehr wie in der abstrakten Repräsentation durch den Graphen punktförmig. Dadurch können an zwei Stellen Probleme entstehen:

- Durch zu geringen Abstand der Netzpunkte auf Superkanten *innerhalb einer CPN* kommt es zu einer Überlagerung der Funktionselemente. Dadurch sind zwar wahrscheinlich die Ergebnisse der Simulation geringfügig betroffen, die Konsistenz der Simulation ist jedoch gewährleistet.
- Durch zu geringen Abstand von Netzpunkten zu Endstücken bzw. durch Zerschneiden von CPN–übergreifenden Superkanten an Stellen eng benachbarter Netzpunkte ist eine fehlerhafte Absorption und Emission von Fahrzeugen möglich. Zur Zeit wird durch Warnungen während der Netzaufbereitung auf diesen Zustand aufmerksam gemacht. Im günstigsten Fall entsteht nur ein Überschuß bzw. Verlust an Fahrzeugen. Es ist aber auch möglich, daß die Simulation einen *Segmentation Violation* Runtime–Fehler hervorruft.

4.8.3 Blockierende Fahrzeuge

Durch abbiegende Fahrzeuge können in der Höhe der Verzögerungsstreifen Spuren ganz für den nachfolgenden Verkehr blockiert werden. Das gleiche Phänomen tritt bei den Deadlocks an Dreiecken auf (siehe 4.3.2). Es ist die Frage, inweit dieses Verhalten als so unrealistisch anzusehen ist, daß es ganz unterbunden werden sollte. Fahrzeuge, die nicht die korrekte Spur innerhalb eines gewissen Abstandes hinter einem Markierungsbereich erreichen, würden dann einfach ihre alte Richtung beibehalten. Der Anteil der dadurch falsch fahrenden Fahrzeuge ist als sehr gering einzustufen.

4.8.4 Mehrspurige Fahrbahnen

Im Rahmen einer realistischen Simulation muß später das mikroskopische Modell so erweitert werden, daß eine größere Anzahl (als zwei) an Spuren unterstützt wird. Nur auf diese Weise kann eine gute Modellierung der Straßenkapazitäten erreicht werden. Dabei besteht der Hauptunterschied beim Übergang von zwei auf drei Spuren darin, daß zwei Fahrzeuge aufgrund der Regeln zum Spurwechsel auf das gleiche Site der gleichen Spur markiert werden. In diesem Fall muß eines der beiden Fahrzeuge ausgesucht werden, das tatsächlich wechselt. Das Hinzufügen von weiteren Spuren beinhaltet keine qualitative Veränderung mehr.

4.9 Bestehende Verfahren

Die Problematik der Verkehrssimulation auf Rechnersystemen ist von vielen Gruppen weltweit in den letzten Jahren angegangen worden. Die meisten beschränken sich jedoch in ihrem Konzept auf Einzelrechner, wodurch die bereits erwähnten Geschwindigkeitseinbußen entstehen. Es sollen deshalb in diesem Rahmen zwei Projekte skizziert werden, die zwar zur Zeit ihrer Enstehung für Einzelrechner gedacht waren, aber augenblickblich auf Parallel–Rechner umgestellt werden. Zum einem handelt es sich dabei um das von der EG unterstützte Projekt PARAMICS und zum anderen um das am Los Alamos National Lab (LANL) ins Leben gerufene Projekt TRANSIMS.

4.9.1 PARAMICS

Das Projekt PARAMICS ist aus drei Teilprojekten entstanden: MICSIM dient zur mikroskopischen Simulation von Fahrzeugen (lokal), PACSIM zur Simulation von Fahrzeuggruppen gleichen Ursprungs und Ziels (regional) und TOPSORT zur Ermittlung globaler Verkehrseigenschaften wie Fluß, Kapazitäten und Kosten. Die Simulation kann in der nicht–parallelisierten Version ein Fahrzeugaufkommen von 200 Fahrzeugen in einem Netz von 50 Netzpunkten auf einer DECstation 3100 in Echtzeit simulieren (im Vergleich dazu berechnet *PAMINA* auf einem System mit gleicher nominaler CPU–Leistung ca. 50000 Fahrzeuge, siehe auch Tab. 4.10). Die grundlegenden Eigenschaften sind wie folgt (siehe [32]):

- vor der Simulation Erstellung von Routenplänen über minimale Kosten, während der Simulation online–Korrekturen
- eigene Hochsprache zur Formulierung des Fahrerverhaltens, anschließende Crosscompilation nach C bzw. C*

- Fahrzeuge werden auf Kanten in Queues verwaltet (geradeaus, links und rechts), Überholvorgänge sind möglich
- an Kreuzen werden Fahrzeuge umverteilt nach Beachtung von Mindestsicherheitsabständen, Ampelschaltungen sind möglich
- Parallelisierung auf der Connection Machine CM–200 unter C*
- Loadbalancing durch Verteilung der Queues und Knoten auf Rechner, Auslastung aller vorhandenen Prozessoren durch Einführung von virtuellen Knoten

4.9.2 TRANSIMS

Das Projekt TRANSIMS soll Verkehrsnetze von Städten und Umgebung simulieren. Der Demo–Version liegen die Daten der Stadt Albuquerque, New Mexico, zugrunde. Dabei werden die Interstates[13] mit ca. 350 Knoten und 450 Kanten mikroskopisch simuliert, der restliche Teil des Netzes mit ca. 10000 Knoten und Kanten nur bei der Routenbedarfsplanung mitberücksichtigt. Die wichtigsten Eigenschaften sind die folgenden:

- Detailierte Formulierung der Fahrer– und Fahrzeugeigenschaften, mit Routenplänen für jedes Fahrzeug
- Ermittlung des Verkehrsaufkommens durch Demographie, daraus Berechnung von Quell–Ziel–Matrizen, die schließlich über Optimierung die Routenpläne ergeben
- Ermittlung von Luftbelastung, Stau–Dynamik, Stau–Häufigkeiten
- soll später als Plattform für intelligente Leitsysteme dienen
- Parallisierung auf Connection Machine CM-5 unter C++
- Es ist geplant, zellulare Automaten als alternative Simulationsmethode zu integrieren.

[13] enstprechen gut ausgebauten mehrspurigen Bundesstraßen

Kapitel 5

Ergebnisse

Während das komplette Netz der Bundesrepublik nur zur Messung der Geschwindigkeit von *PAMINA* diente, wurde das Netz von Nordrhein–Westfalen für gezielte Beobachtungen eingesetzt. Zum einen wurde wie im zweispurigen Fall das dynamische Fundamentaldiagramm berechnet, zum anderen die zeitliche Entwicklung von Meßgrößen im gesamten System und auf ausgesuchten Teilabschnitten betrachtet[1]. Für die Durchführung der Simulation wurde Run Mode 1 (dynamisches Fundamentaldiagramm, Relaxation des gesamten Systems) und Run Mode 3 (Relaxation ausgewählter Bereiche) mit den in 4.6.9 angegebenen Parametern und Randbedingungen verwendet.

5.1 Relaxation des Gesamtflusses

Für die Messung der Relaxation des Gesamtflusses wurde der Run Mode 1 verwendet. Nach Füllen des Systems mit einer homogenen Dichte wurde ein Simulationsdurchlauf durchgeführt, bei dem in Intervallen von jeweils 1024 Zeitschritten Messungen gesammelt wurden, die in Abb. 5.1 und Abb. 5.2 zu sehen sind. Man erkennt einen deutlichen Unterschied von den Dichten unterhalb und oberhalb von $\varrho = 0.04$. Bei Dichten unterhalb des Schwellenwertes ist keinerlei Relaxation festzustellen. Das Fundamentaldiagramm ist zeitunabhängig. Bei Dichten oberhalb erkennt man eine starke Abnahme des Flusses mit der Zeit. Selbst nach über 15000 [s] ist dieser Effekt nicht gänzlich abgeklungen.

[1] Hierbei ist zu beachten, daß nur der Bereich, über den sich die Statistik erstreckt, eingeschränkt wurde, nicht jedoch der Simulationsbereich. Dieser belief sich immer auf das komplette Netz.

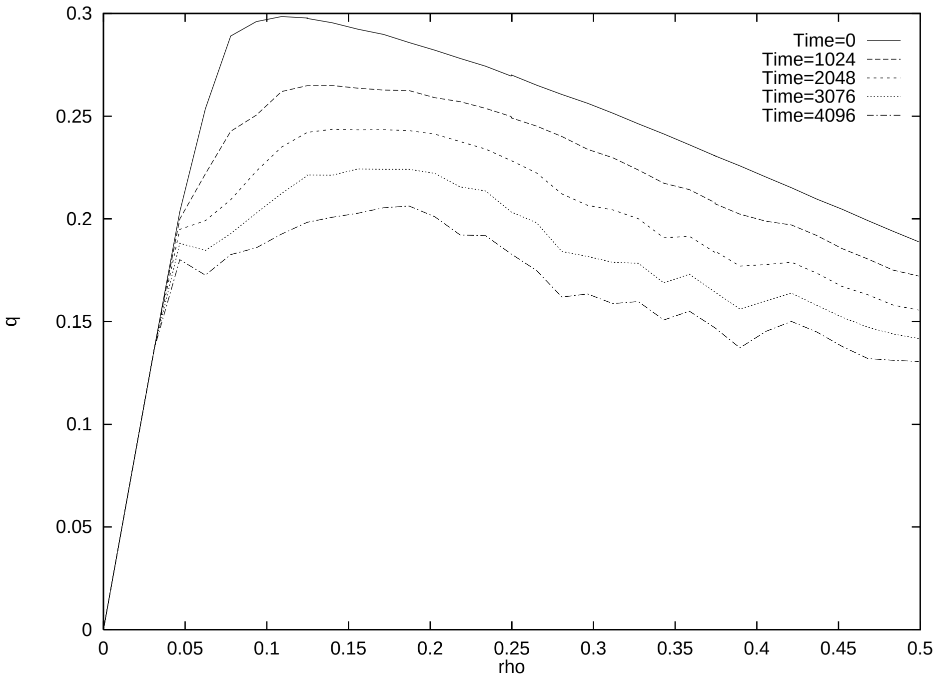

Abbildung 5.1: *Relaxation des Gesamtflusses NRW*

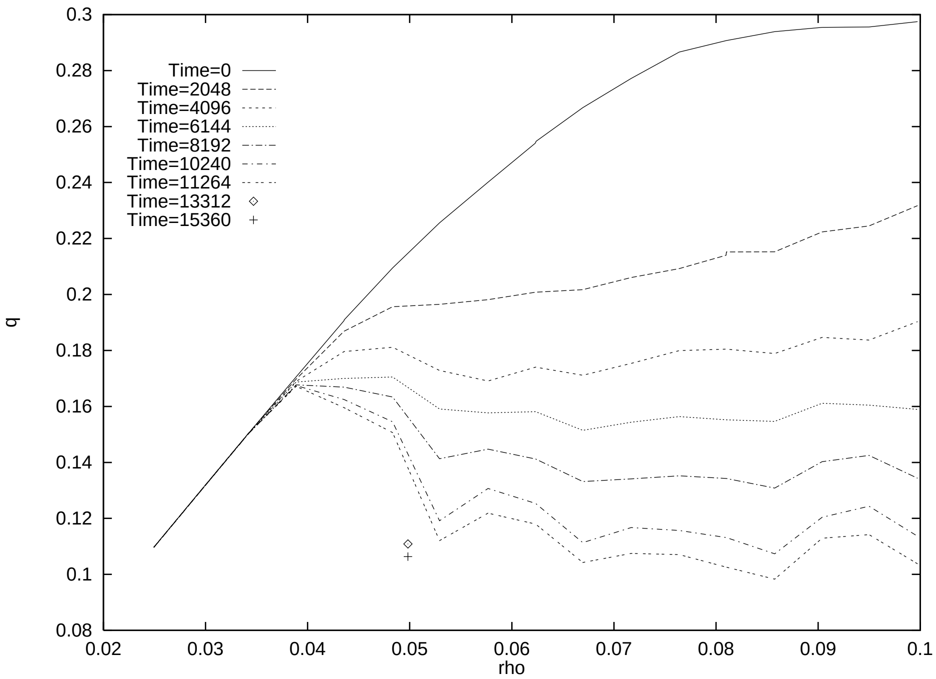

Abbildung 5.2: *Relaxation des Gesamtflusses NRW (Detail)*

5.2 Relaxation ausgewählter Bereiche

Im Netz NRW wurden mit Hilfe des Programms *PROFIATLAS* der Firma PROFI.S, Remscheid, zwei Netzabschnitte zur individuellen Flußbeobachtung extrahiert. Dabei handelt es sich um den Autobahnring um Köln mit einer Gesamtlänge von 51.7 [km] und um die Strecke vom Kamener Kreuz bis zur Anschlußstelle Bielefeld Sennestadt (genannt *lineare Strecke*) mit einer Länge von 76.5 [km] (siehe Abb. 5.3). Es wurde jeden 20. Zeitschritt Statistik aufgenommen. Der Verlauf der Dichte als Funktion der Zeit ist in Abb. 5.4 zu sehen. Gemessen wurde die zeitliche Entwicklung von $\varrho(t)$, $v_{mean}(t)$ und $q(t)$ für vier verschiedene Gesamtdichten ϱ_0. Die Kurvenverläufe sind auf den Seiten 68 bis 74 abgebildet.

Bei Dichte $\varrho_0 = 0.1$ (Seite 68) kommt es durch die stark zunehmende Fahrzeuganzahl nach ca. 5000 Sekunden zu einem Verkehrskollaps auf dem Ring. Im Gegensatz dazu leeren sich die linearen Strecken immer mehr, so daß dort ein freier Verkehrfluß möglich wird. Besonders auffällig ist der asymmetrische Verlauf für die beiden Flußrichtungen am Kölner Ring. Während sich der Fluß im Uhrzeigersinn auf Dichte 0.2 stabilisiert, steigt die Dichte entgegen des Uhrzeigersinns stetig an und hat auch nach 10000 [s] noch kein Maximum erreicht.

Bei $\varrho = 0.05$ (Seite 69) ist ein langsameres Ansteigen der Dichte und des Flusses am Ring zu erkennen. Jedoch ist nach ca. 10000 Sekunden auch bei dieser geringeren Dichte kein nennenswerter Fluß mehr zu verzeichnen. Die linearen Strecken sind wieder nach kurzer Anlaufzeit frei befahrbar. Es fällt auf, daß die Kurven nicht mehr monoton verlaufen, wie bei $\varrho = 0.1$, sondern schon leichte Schwingungen zeigen.

Bei $\varrho = 0.04$ (Seite 70) ist ein *grundsätzlicher* Unterschied in allen Kurven zu erkennen. Der Fluß auf dem Kölner Ring kollabiert nicht mehr, sondern nähert sich schnell einem Wert, der geringfügig oberhalb der vorgegebenen Gesamtdichte liegt. Die Asymmetrie der beiden Richtungen des Rings ist sehr viel geringer als bei größeren Dichten. Dies ist konform mit den Ergebnissen aus Abb. 5.2, wo die Dichte 0.04 die Grenze zwischen den Bereichen mit und ohne Relaxation bildet. Bei Dichte und Fluß der linearen Strecken wird ein schwingendes Verhalten (mit Schwebung) sichtbar, wobei die Perioden der Schwingungn auf beiden

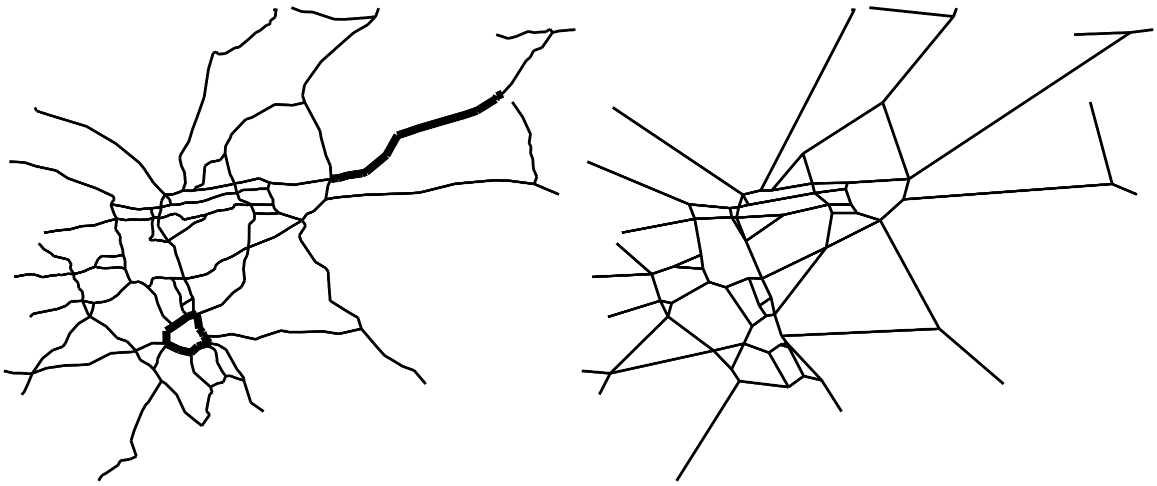

Abbildung 5.3: *Das Autobahnnetz von Nordrhein–Westfalen* — Auf der linken Seite ist das Originalnetz, auf der rechten Seite das reduzierte Netz zu sehen. Die Strecken, die in 5.2 gesondert untersucht wurden, sind mit einer fetteren Strichstärke gezeichnet.

Richtungsfahrbahnen zueinander um einen konstanten Wert versetzt sind. Es wurde deshalb eine Fourieranalyse (FFT) und eine Cross–Korrelations–Analyse der Kurven durchgeführt. Die Fourieranalyse basiert auf der Routine `spctrm.c` aus [27], die mit rechteckigem Fenster aufgerufen wurde. Die Ergebnisse sind in Abb. 5.13 und Abb. 5.15 auf Seite 71 zu sehen.

Im Gegensatz zur FFT der Durchschnittsgeschwindigkeit, die (abgesehen von den starken Schwankungen) fast waagerecht verläuft (weißes Rauschen), zeigt die FFT des Flusses einen klaren Anstieg in Abhängigkeit der Periodenlänge. Dieser Zusammenhang wird als $1/f$–Rauschen bezeichnet. Ähnliche Abhängigkeiten ergeben sich zum Beispiel in der Dichteverteilung beim Fluß von granularen Medien durch Röhren [14]. Dort liegt der Koeffizient α bei 1.33 ± 0.02.

Für die lineare Strecke wurde deshalb eine erneute Simulation durchgeführt mit der doppelten Anzahl an Zeitschritten und mit einer Aufnahme der Statistik jeden 5. statt jeden 20. Zeitschritt. Für die FFT wurden die Daten in $k = 8$ Segmente unterteilt[2]. Anschließend wurde bei doppelt–logarithmischer Abtragung eine Regressionsanalyse für die Werte mit $f < 0.2$ durchgeführt. Es ergab sich der folgende Zusammenhang zwischen Intensität P und Frequenz f:

$$P(f) \quad = \quad e^{\beta + \alpha \log f}$$

mit den Parametern aus Tab. 5.1. Der lineare Abfall über drei Größenornungen bei den

Strecke	α	β
BS nach KK	-2.05 ± 0.02	-25.86 ± 0.08
KK nach BS	-2.03 ± 0.02	-25.80 ± 0.08
Ring im Uhrzeigersinn	-1.39 ± 0.01	-22.68 ± 0.04
Ring entgegen Uhrzeigersinn	-1.81 ± 0.03	-24.0 ± 0.1

Tabelle 5.1: *Koeffizienten der Regression*

linearen Strecken BS nach KK (Abb. 5.25) und KK nach BS (Abb. 5.26) ist bemerkenswert. Erst bei Frequenzen über 0.2 ist ein Abknicken in die Horizontale sichtbar. Beim Ring im Urzeigersinn (Abb. 5.27) ist der lineare Abfall nicht so stark ausgeprägt. Er erstreckt sich nur über knapp 2 Größenordnungen und ist sowohl bei kleinen als auch bei großen Frequenzen schwer zu definieren. Die FFT für den Ring entgegen dem Uhrzeigersinn (Abb. 5.28) liegt qualitativ zwischen denen der linearen Strecken und der des Rings im Uhrzeigersinn.

Bei Dichte $\varrho = 0.03$ (Seite 72, 73) treten ähnliche Effekte wie bei $\varrho = 0.04$ auf, allerdings weniger stark ausgeprägt. Sie wurden deswegen nicht gesondert untersucht.

[2] Es wurde hierbei von der Dichte zum Fluß übergegangen. Die beiden Größen sind jedoch bei der allgemein schwachen Belegung der Strecke äquivalent und unterscheiden sich nur durch den Faktor v_{free}. Desweiteren wurde dieses Mal die Frequenz und nicht die Periodendauer abgetragen. Dadurch fällt jetzt die Kurve.

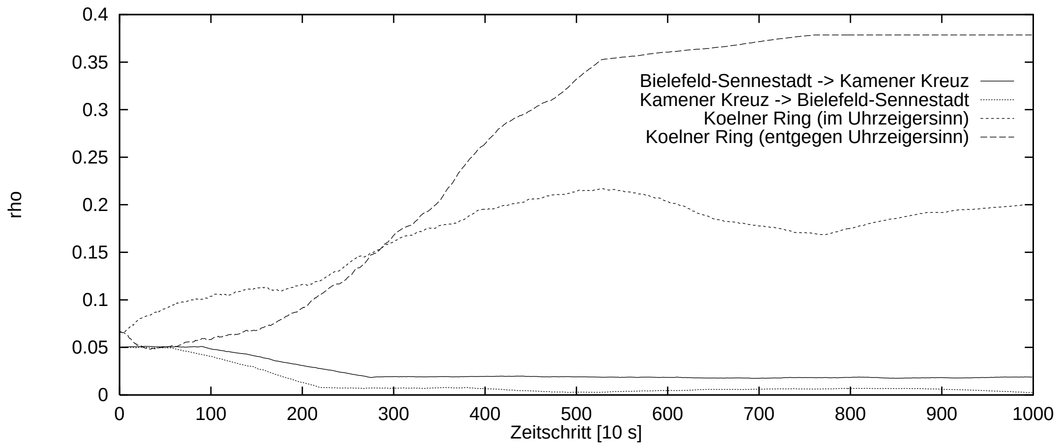

Abbildung 5.4: *Relaxation $\varrho(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.1$*

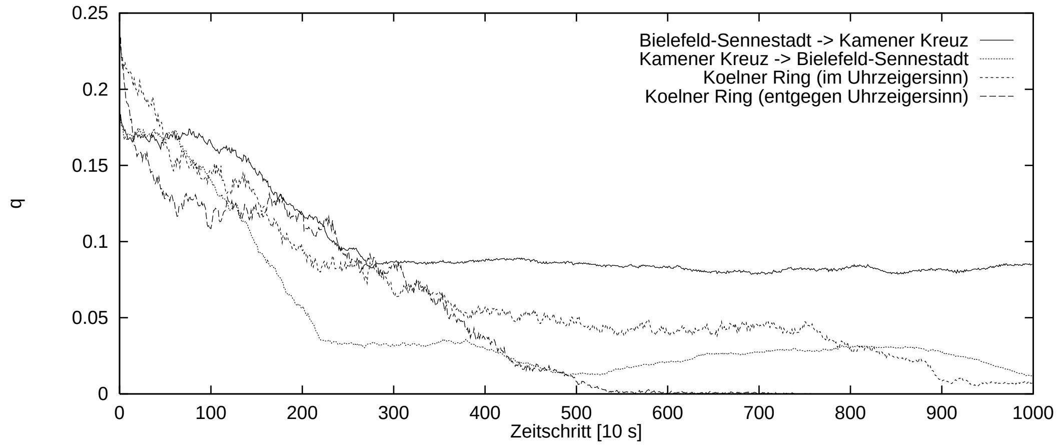

Abbildung 5.5: *Relaxation $q(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.1$*

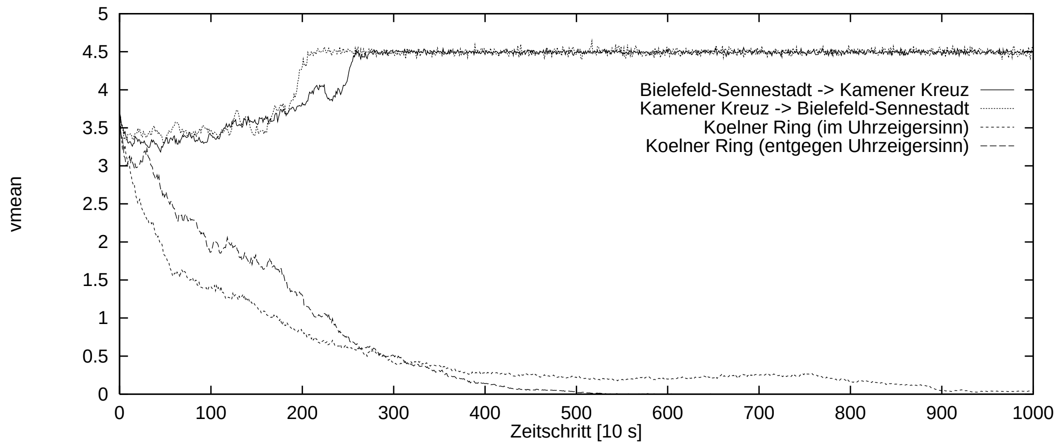

Abbildung 5.6: *Relaxation $v_{mean}(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.1$*

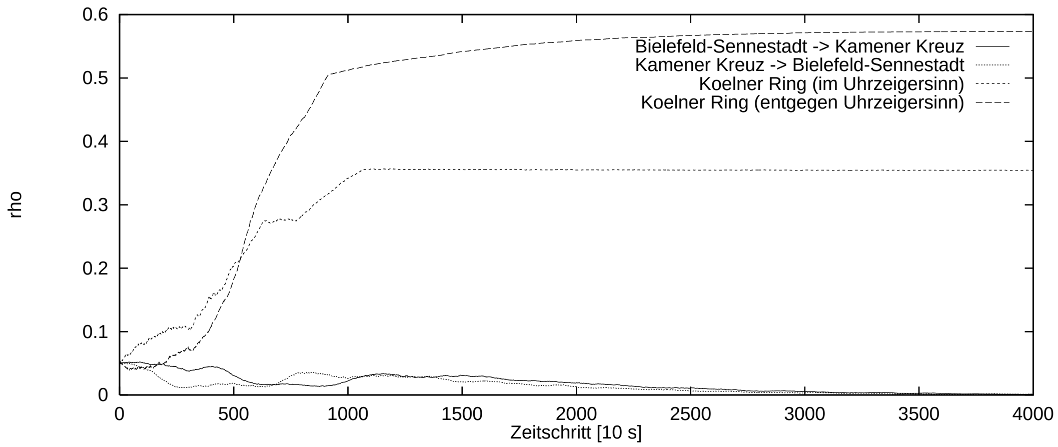

Abbildung 5.7: *Relaxation $\varrho(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.05$*

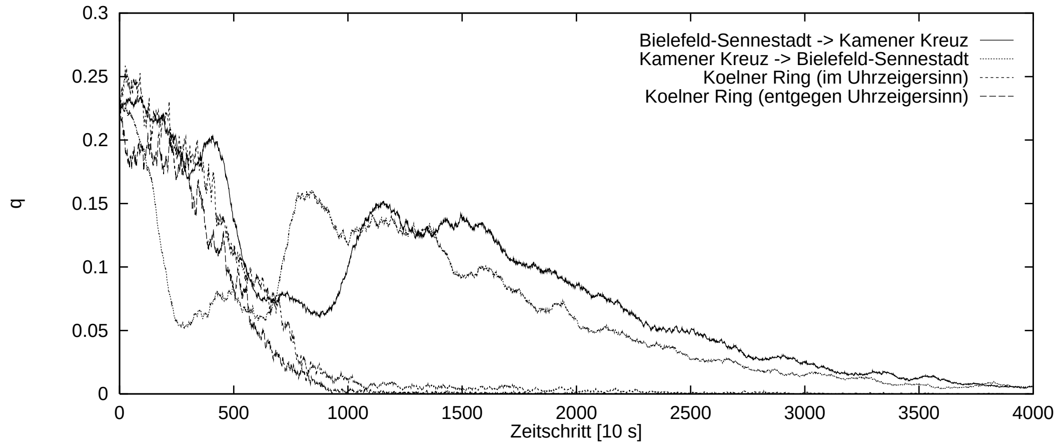

Abbildung 5.8: *Relaxation $q(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.05$*

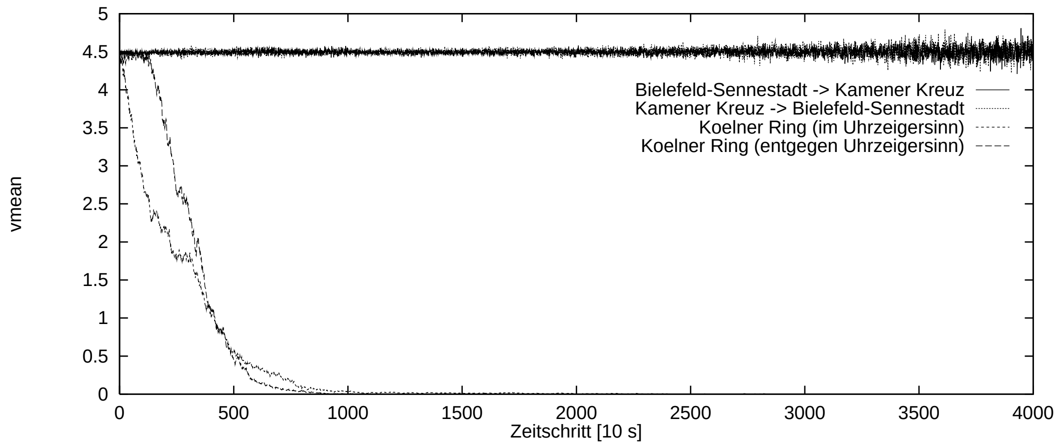

Abbildung 5.9: *Relaxation $v_{mean}(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.05$*

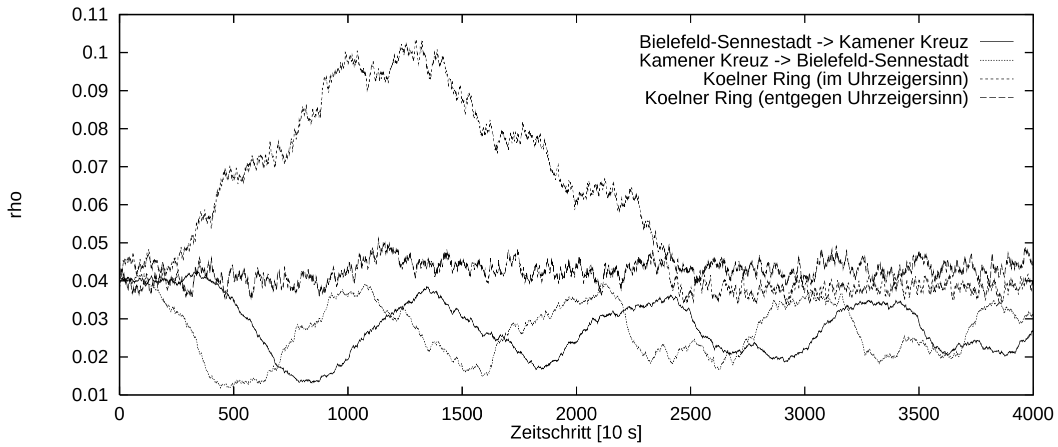

Abbildung 5.10: *Relaxation $\varrho(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.04$*

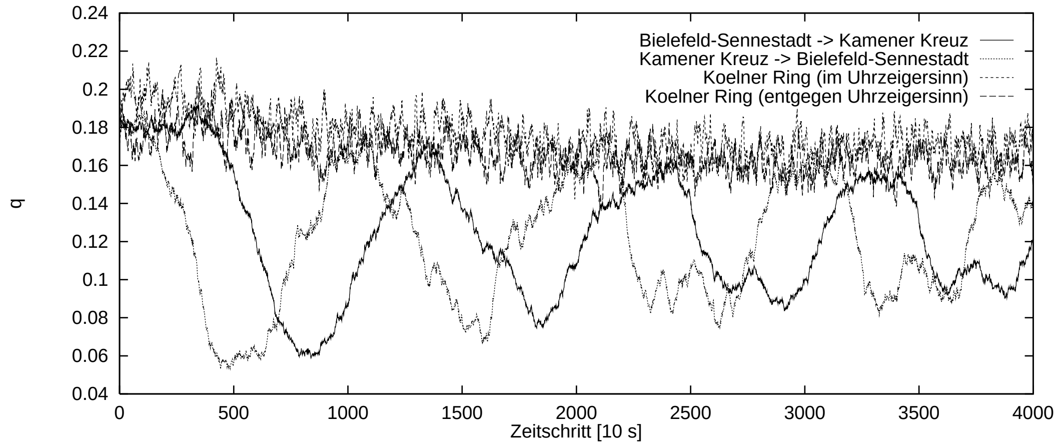

Abbildung 5.11: *Relaxation $q(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.04$*

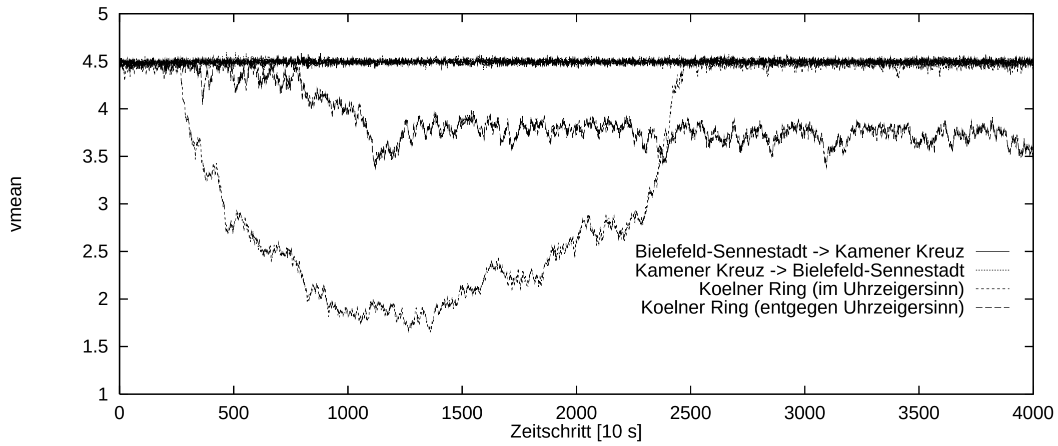

Abbildung 5.12: *Relaxation $v_{mean}(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.04$*

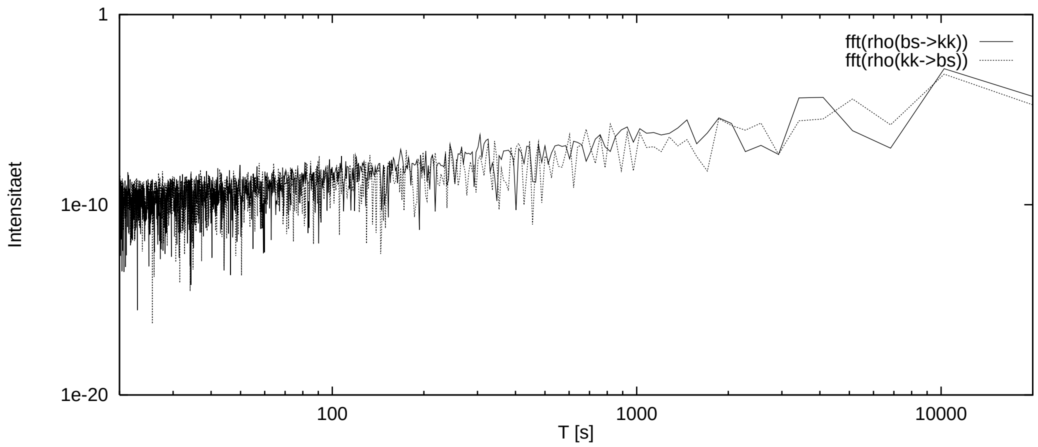

Abbildung 5.13: *Fourier–Transformation von $\varrho(t)$ für $\varrho_0 = 0.04$*

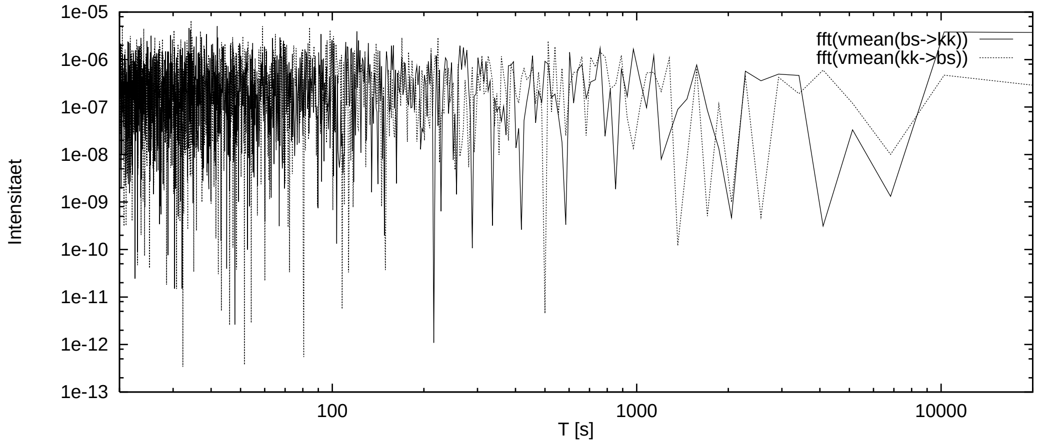

Abbildung 5.14: *Fourier–Transformation von $v_{mean}(t)$ für $\varrho_0 = 0.04$*

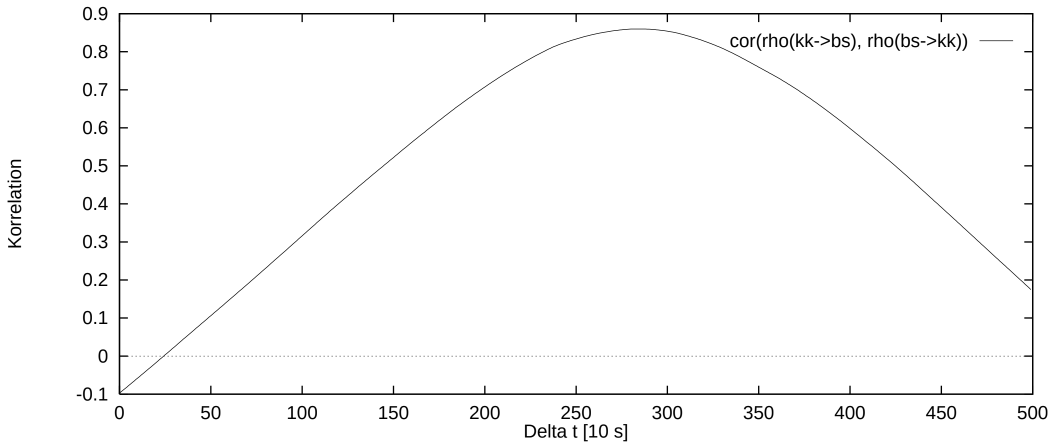

Abbildung 5.15: *Cross–Korrelation für entgegengesetzte Richtungen für $\varrho_0 = 0.04$*

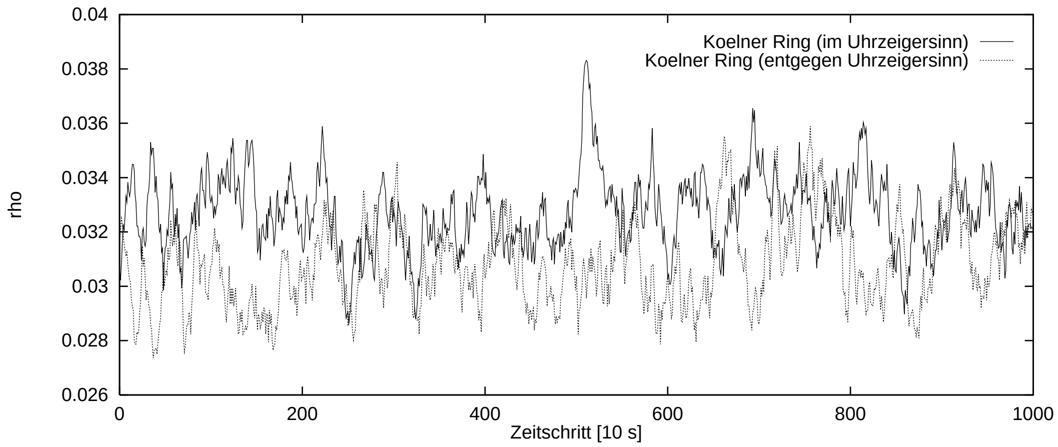

Abbildung 5.16: *Relaxation $\varrho(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.03$ (Ring)*

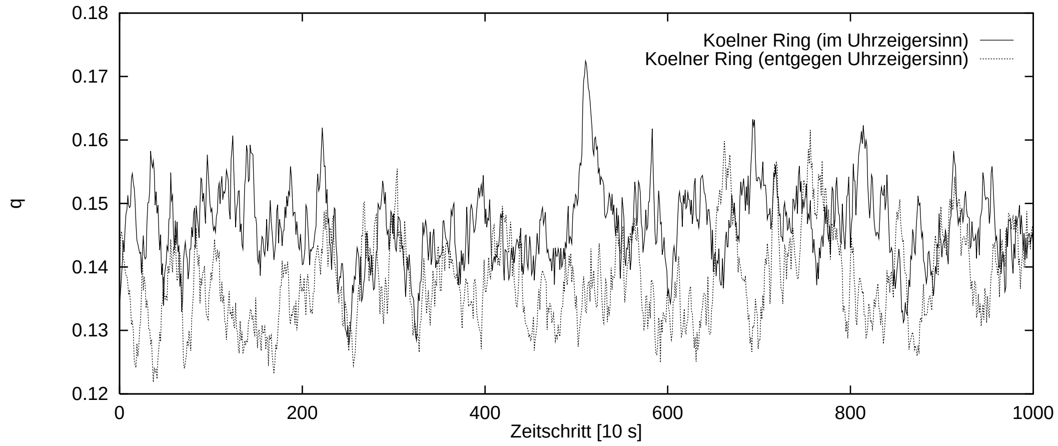

Abbildung 5.17: *Relaxation $q(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.03$ (Ring)*

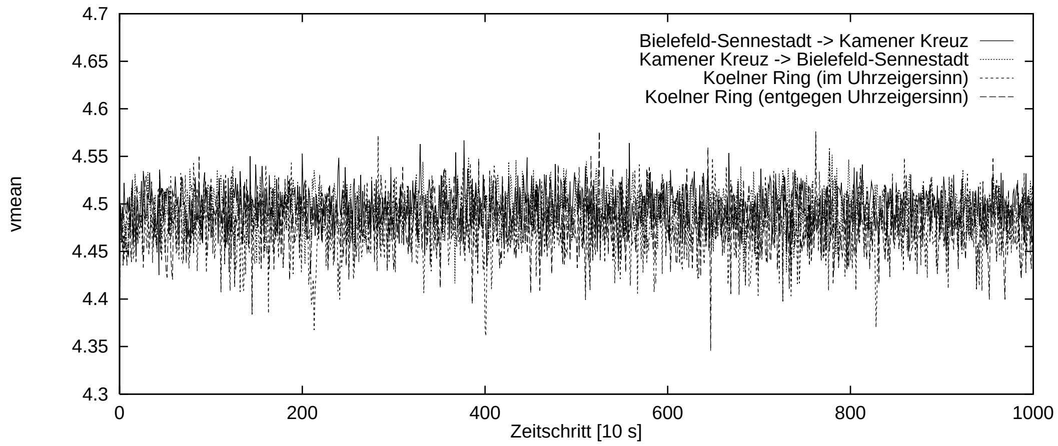

Abbildung 5.18: *Relaxation $v_{mean}(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.03$*

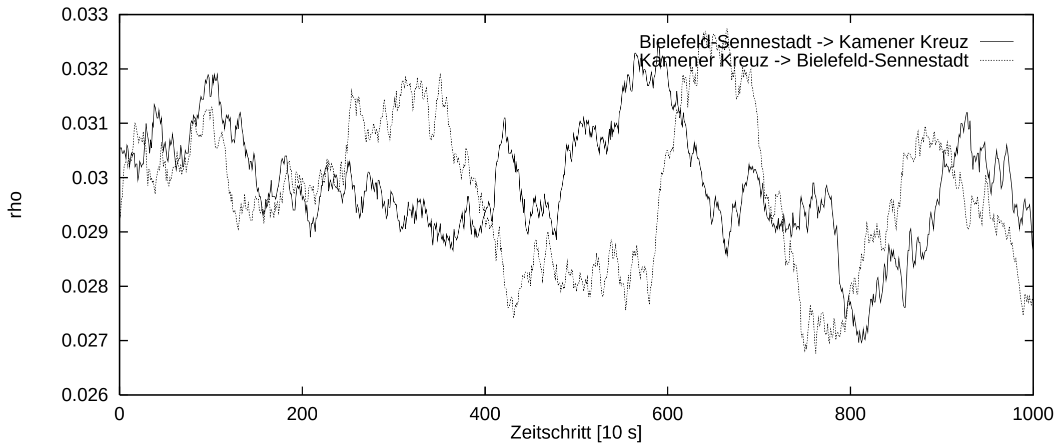

Abbildung 5.19: *Relaxation $\varrho(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.03$ (linear)*

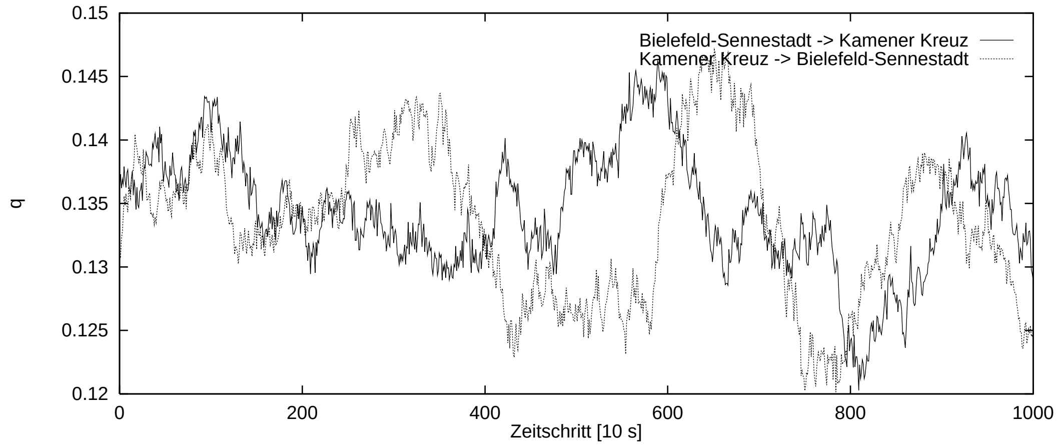

Abbildung 5.20: *Relaxation $q(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.03$ (linear)*

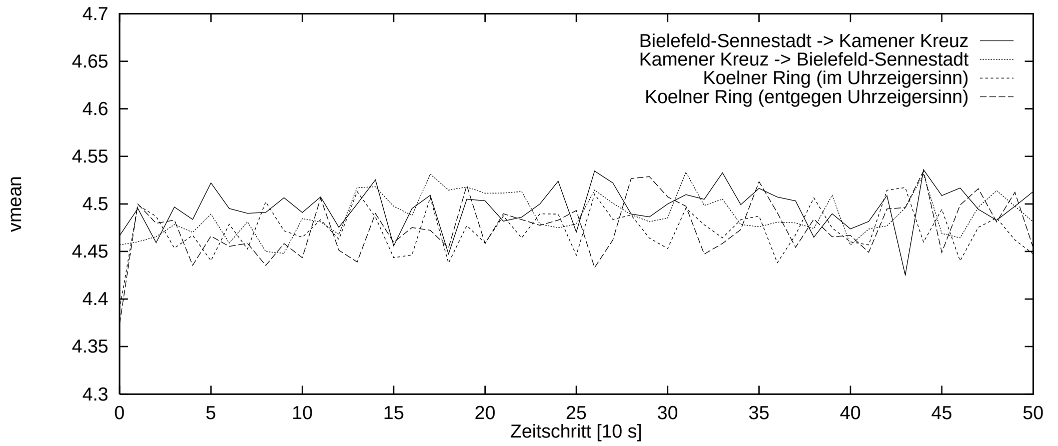

Abbildung 5.21: *Relaxation $v_{mean}(t)$ ausgewählter Bereiche in NRW bei $\varrho_0 = 0.03$ (Detail)*

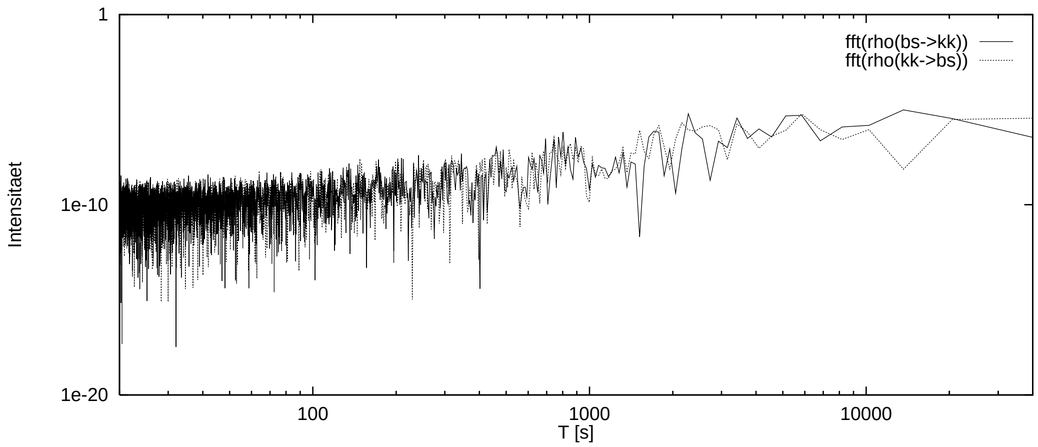

Abbildung 5.22: *Fourier–Transformation von $\varrho(t)$ für $\varrho_0 = 0.03$*

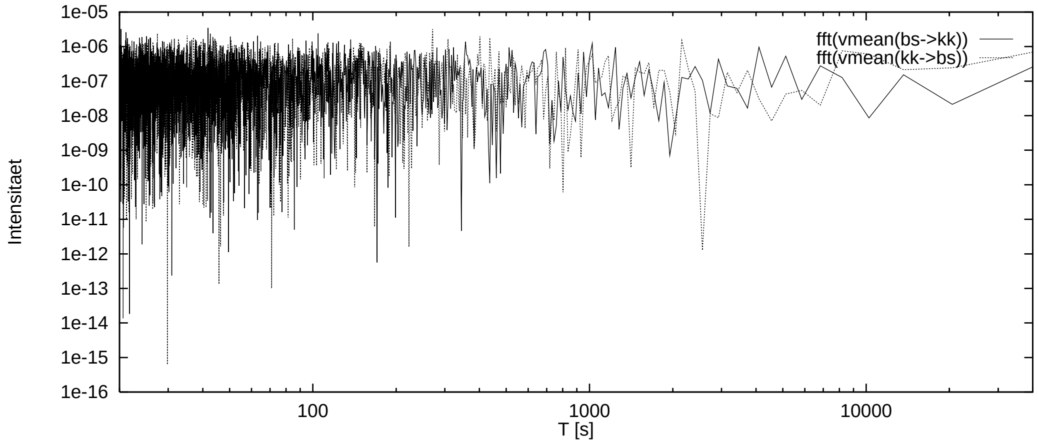

Abbildung 5.23: *Fourier–Transformation von $v_{mean}(t)$ für $\varrho_0 = 0.03$*

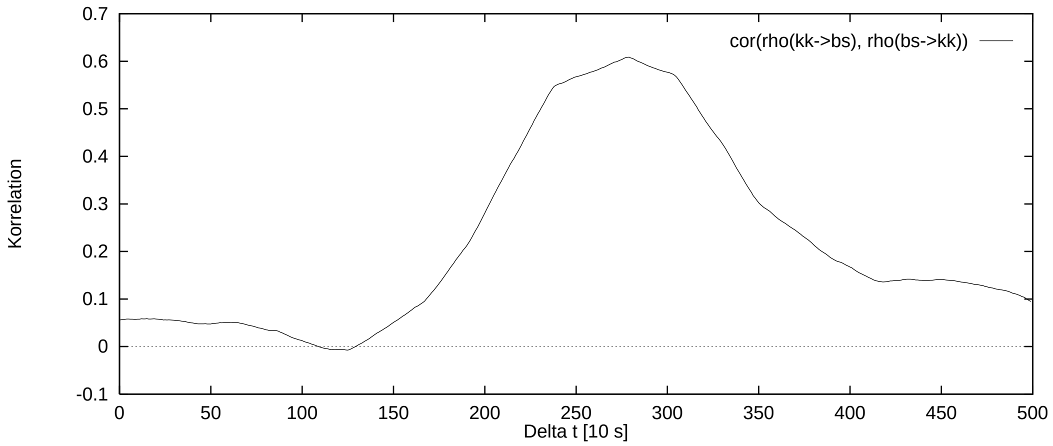

Abbildung 5.24: *Cross–Korrelation für entgegengesetzte Richtungen für $\varrho_0 = 0.03$*

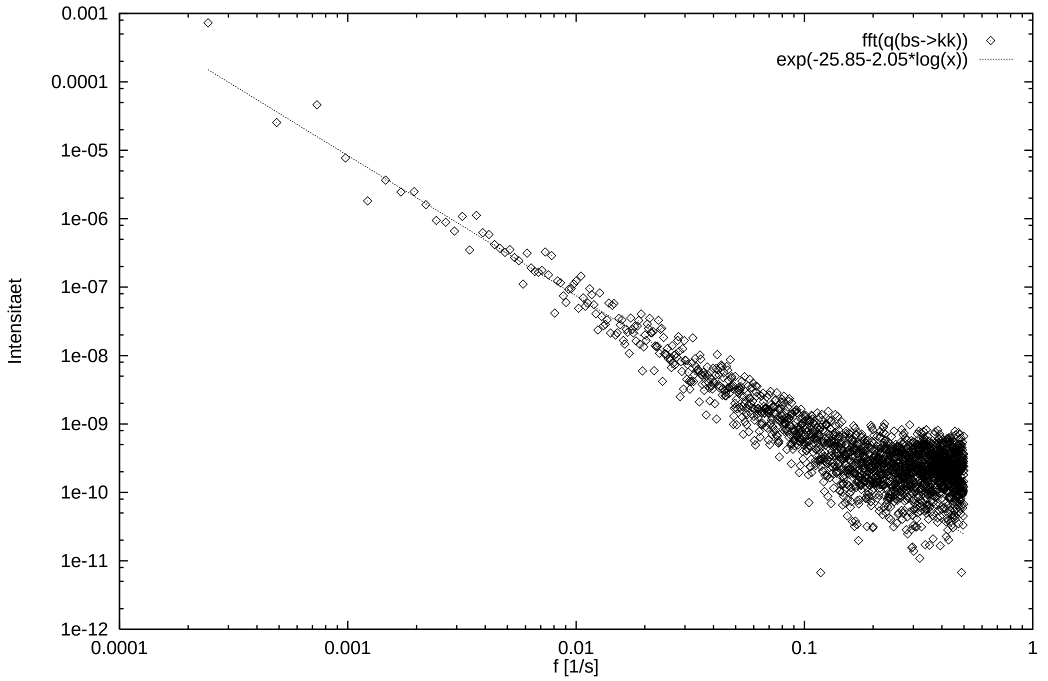

Abbildung 5.25: *Fourier–Transformation von $q(t)$ für BS nach KK bei* $\varrho_0 = 0.04$

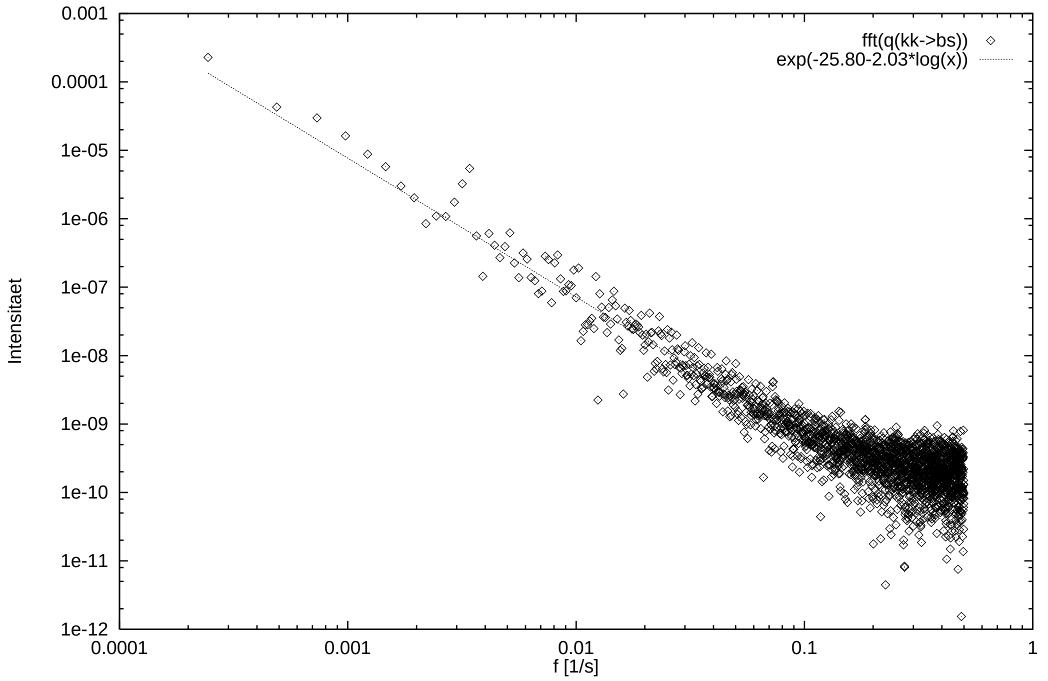

Abbildung 5.26: *Fourier–Transformation von $q(t)$ für KK nach BS bei* $\varrho_0 = 0.04$

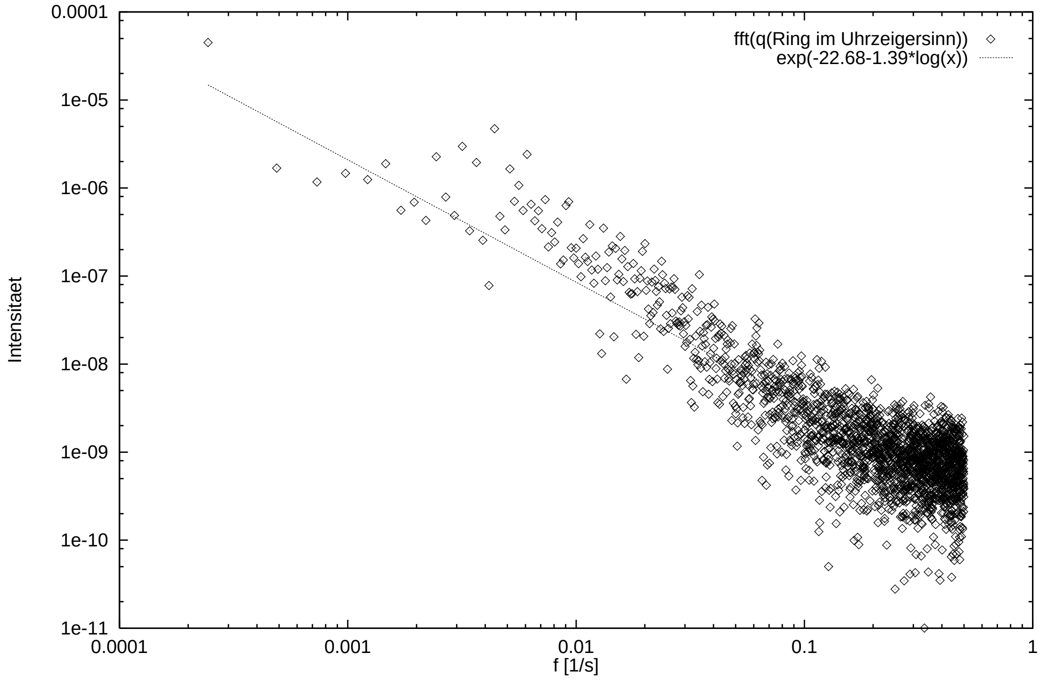

Abbildung 5.27: *Fourier–Transformation von $q(t)$ für Ring i.U. bei $\varrho_0 = 0.04$*

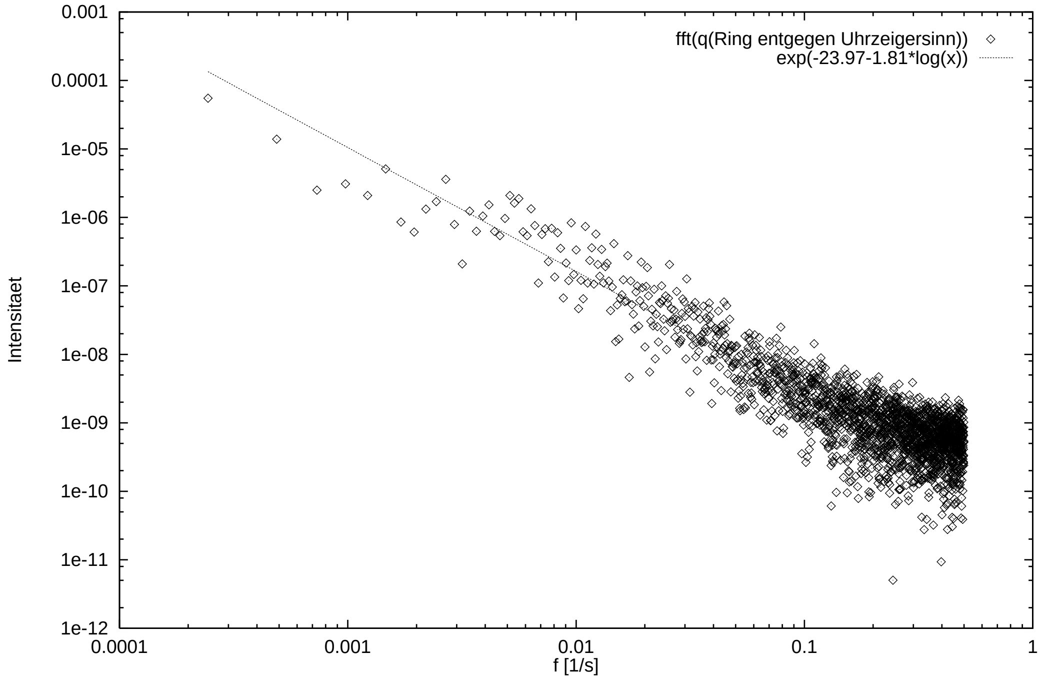

Abbildung 5.28: *Fourier–Transformation von $q(t)$ für Ring e.U. bei $\varrho_0 = 0.04$*

5.3 Dynamisches Fundamentaldiagramm

Beim dynamischen Fundamentaldiagramm für Netze wurde die Einteilung des Netzes in Pfade ausgenutzt. Nach Vorgabe einer globalen Dichte im Bereich von 0.0 bis 0.3 (in 0.015 Schritten) wurden 500 Zeitschritte verworfen. Anschließend wurde für jeden Pfad mit mindestens 2500 Sites (Länge größer gleich 18.7 [km]) die aktuelle Belegung mit Fahrzeugen ermittelt. Die Diagramme zeigen einen deutlichen Unterschied zum einfachen zweispurigen Fall: bei Dichten oberhalb von 0.04 existieren Punkte bei kleinen Flüssen, die weit unterhalb des normalen Verlaufs des Fundamentaldiagramms liegen. Andererseits findet man auch Punkte weit oberhalb der maximal vorgegebenen Gesamtdichte, die aber dem Verlauf des Fundamentaldiagramms entsprechen. Dies ist durch Pfade zu erklären, die durch dichte Teilnetze laufen (siehe auch 5.2): In dichten Teilnetzen ist die Anzahl der Autobahndreiecke und Autobahnkreuze höher als normal. An ihnen entstehen durch Abbieger Rückstaus, die die Dichte lokal heraufsetzen, aber keinen Beitrag zum Fluß liefern (lokaler Stillstand). Je größer die Anzahl der Knoten pro Pfad ist, desto höher wird die Dichte, und desto weiter liegt der Punkt rechts im Fundamentaldiagramm.

Was die Unterbrechungen im linearen Anstieg bei Dichten unterhalb von 0.04 angeht, so ist das eine Folge der Vorgabe der Gesamtdichte in Schritten von 0.015. Wie in 5.2 gesehen, verändern sich die lokalen Meßgrößen bei Dichten unterhalb 0.04 kaum, so daß in diesen Bereichen keine Ausschmierung der Punkte erwartet werden kann. Es wurde keine feinere Unterteilung durchgeführt, weil dadurch die Punktanzahl des Diagramms insgesamt zu hoch gewesen wäre, was seinerseits die visuelle Abschätzung der Punktdichte in den einzelnen Bereichen erschwert hätte.

Bei $RatioS = 0.1$ findet man keine so starke Auffächerung bei kleinen Dichten — es fehlen dort zu viele Punkte — wie im einfachen zweispurigen Fall. Sie beginnt erst bei Dichten oberhalb 0.05. Allgemein verläuft die Kurve im Bereich des Maximums etwas flacher als ohne langsame Fahrzeuge. Sie ist am ehesten mit der Kurve für $RatioS = 0.2$ im zweispurigen Modell (3.27) zu vergleichen.

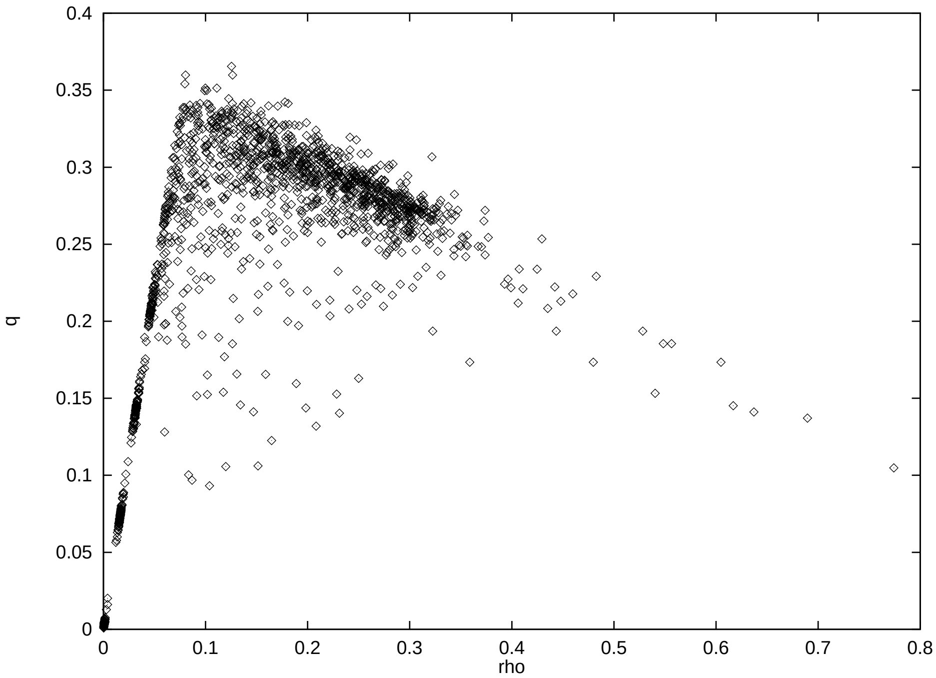

Abbildung 5.29: *Dynamisches Fundamentaldiagramm von NRW für RatioS* = 0.0

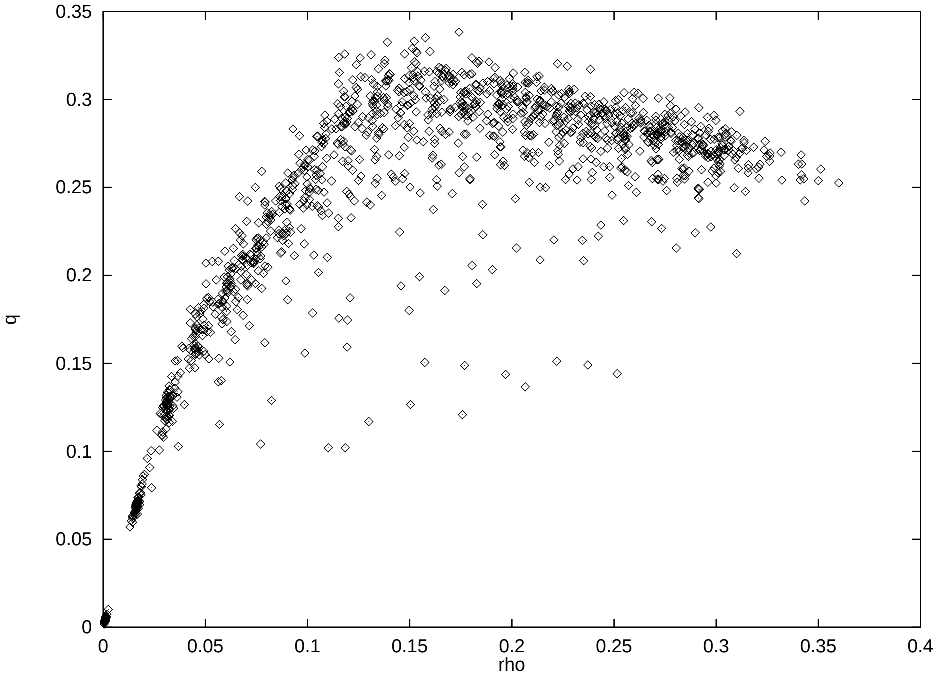

Abbildung 5.30: *Dynamisches Fundamentaldiagramm von NRW für RatioS* = 0.1

Kapitel 6

Implementation

Dieses Kapitel wendet sich an den technisch interessierten Leser.

6.1 Allgemeine Aspekte

Hier sollen kurz die grundlegenden Aspekte der Implementation von zellularen Automaten im allgemeinen und des Verkehrsmodells im besonderen angesprochen werden.

6.1.1 Gitterorientierte oder listenorientierte Programmierung?

Das einspurige Modell kann auf zwei grundlegend verschiedenen Arten implementiert werden:

Gitterorientiert Der gitterorientierte Ansatz entspricht genau dem Konzept des zellularen Automaten, indem jedes Site (sowohl belegte als auch unbelegte) der Strecke im Speicher durch einen Speicherplatz repräsentiert wird. Durch die lineare Anordnung und der identischen Länge kann über die Numerierung der Sites die Position auf der Strecke bestimmt werden. Ein explizites Abspeichern der Position eines Fahrzeugs ist daher nicht mehr notwendig. Beim Update wird auf *alle* Sites des Systems einmal zugegriffen, unabhängig davon, ob ein Site belegt oder frei ist. Demnach beträgt die untere Schranke für die Worst–Case–Laufzeit eines gitterorientierten Algorithmus $O(L)$, wobei L die Länge der Strecke in Sites ist.

Listenorientiert Beim listenorientierten Ansatz werden nicht die Sites durch Speicherelemente repräsentiert, sondern die Fahrzeuge selbst. Jedes Fahrzeug hat dabei Informationen über seine augenblickliche Position und einen Verweis auf seinen Vordermann[1]. Beim Update wird nur einmal auf jedes Fahrzeug zugegriffen, so daß sich die untere Schranke für die Worst–Case–Laufzeit eines listenorientierten Ansatzes zu $O(N)$ ergibt, wobei N die Anzahl der Fahrzeuge im System ist.

In der Praxis (und auch in den Algorithmen dieser Arbeit) wird bei der Implementation des gitterorientierten Algorithmus zuerst für jedes Site geprüft, ob dieses belegt ist oder nicht. Nur wenn die Bedingung zutrifft, werden alle weiteren Abfragen (zum Beispiel für den Spurwechsel) durchgeführt. Dies hat besonders bei schwach besetzen Systemen eine Erhöhung

[1] wie bei einer einfach verketteten Liste

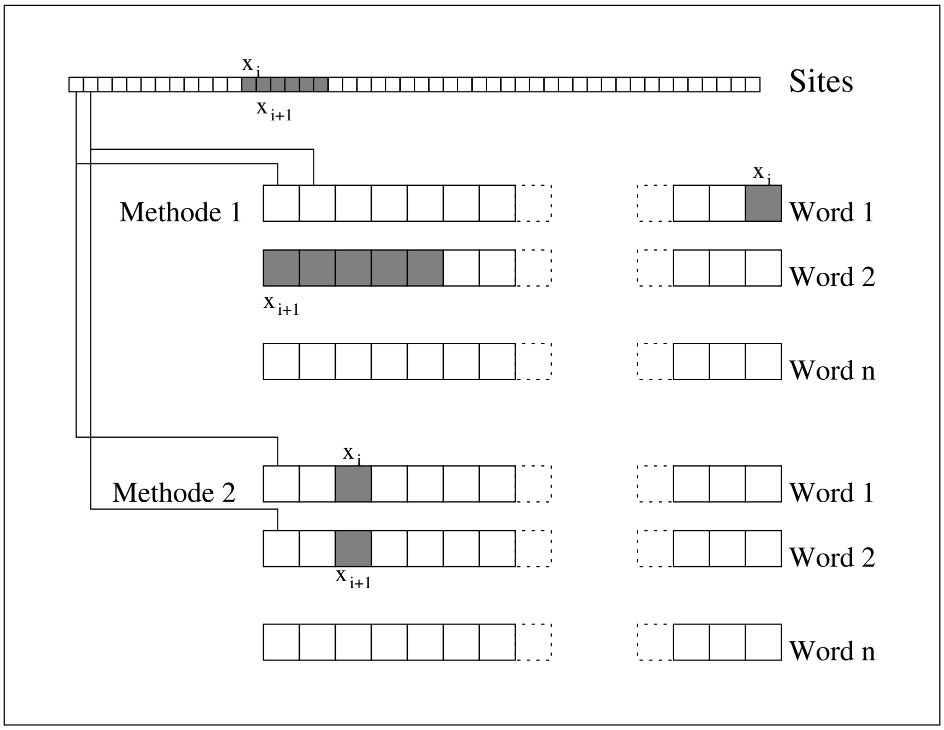

Abbildung 6.1: *Methoden der Bitspeicherung*

der Geschwindigkeit zur Folge. Man nennt diese Abwandlung des streng gitterorientierten Algorithmus auch *intermediate version*.

Der Vorteil des listenorientierten Algorithmus für geringe Fahrzeugdichten ist offensichtlich. Jedoch wird durch Kombination mit Vektorisierung der Vorteil zunichte gemacht. Für das zweispurige Modell ist der listenorientierte Ansatz ebenfalls ungünstig, was in 6.2.1 besprochen wird. Nagel und Schleicher [24] haben Algorithmen der beiden Ansätze miteinander verglichen. Dabei ergab sich, daß bei gleichzeitiger Verwendung von binärer Codierung (siehe 6.1.2) der gitterorientierte Algorithmus entweder gleich schnell oder schneller ist als der listenorientierte. Die einzige Ausnahme bilden kleine Systeme mit $L < 4096$ pro CPN auf der Connection Machine CM-5. Der größte Geschwindigkeitsvorteil von ca. 200 ergab sich bei Simulation von Systemen mit über 2^{18} Sites auf dem Vektorrechner NEC.

6.1.2 Binäre Codierung

Da sich zellulare Automaten durch eine geringe Anzahl von Zuständen auszeichnen, kann versucht werden, diese Zustände nicht wie üblich durch `integer`–Variablen zu repräsentieren, sondern die einzelnen Möglichkeiten in Bits zu codieren. Es werden dann je nach Wortlänge 32 oder 64 bits — und damit Sites — zu einem `integer` zusammengefaßt. Vergleichsoperationen bzw. Zuweisungen werden über Boolesche Operationen durchgeführt, womit eine Abarbeitung von 32 bzw. 64 Sites in einem Systemtakt des Rechners möglich wird. Im vorliegenden Fall hat man sieben Zustände und benötigt demnach sieben Bit–Felder. Im Prinzip kann das Feld `occupied` aus den anderen über eine Boolesche ODER–

Name	Inhalt
`occupied`	gesetzt, wenn Site besetzt
`v0`	gesetzt, wenn Fahrzeug Geschwindigkeit 0 hat
...	
`v5`	gesetzt, wenn Fahrzeug Geschwindigkeit 5 hat

Tabelle 6.1: *Bit–Felder*

Verknüpfung ermittelt werden. Es hat sich jedoch herausgestellt, daß es günstiger ist, diesen Wert gesondert abzuspeichern.

Bei der Zuweisung der einzelnen Bit–Werte der Sites auf Computer–Worte darf nicht der naheliegende Ansatz gewählt werden, *nebeneinanderliegende Sites* auf *nebeneinanderliegende Bits* in den Computer–Worten zu übertragen (Methode 1, [29]). Viel besser ist es nebeneinander liegende *Sites* auf nebeneinander liegende *Computer–Worte* zu verteilen (Methode 2, [30]). Dies hat folgende Ursache (siehe dazu Abb. 6.1). In der einspurigen Simulation müssen für ein Site x_i die fünf folgenden Sites $x_{i+1} \dots x_{i+5}$ überprüft werden. Diese fünf Sites befinden sich im ersten Fall entweder in den folgenden Bits des gleichen Wortes oder im folgenden Wort, oder in beiden, was eine komplizierte Abfrage von Spezialfällen fordert. Liegen die Sites jedoch in aufeinanderfolgenden Worten, kann für Site x_i, dessen Bit in Wort w_i liegt, der Nachfolger x_{i+j} im Wort w_{i+j} gefunden werden und zwar an der gleichen Bitposition, wodurch ein einfaches Aufstellen von Masken möglich wird.

6.1.3 Vektorisierung

Bei den gängigen Vektor–Computern versucht der Compiler innerste `DO`–Schleifen nicht sequentiell, sondern gleichzeitig mit Hilfe von Vektorregistern abzuarbeiten. Die Länge dieser Vektorregister ist variabel und liegt ungefähr zwischen 64 und 1024 Vektorelementen. Diese Werte können unter günstigen Voraussetzungen gleichzeitig als Leistungsfaktor gegenüber sequentiell arbeitenden Maschinen angenommen werden. Leider ist der effiziente Einsatz von Vektor–Computern stark von der Struktur des Algorithmus abhängig. Besonders zwei Eigenschaften erschweren den Einsatz:

- Innere Abhängigkeiten von Schleifen, bei denen im Durchlauf n ein Ergebnis eines früheren Durchlaufs $1, \dots, n-1$ verwendet wird. In diesem Fall ist auch der Vektorcomputer zur sequentiellen Abarbeitung gezwungen. Glücklicherweise ist beim parallelen Update des Verkehrsmodells dieses Problems nicht gegeben. Ein Random–Sequential–Update hingegen würde den Vektorrechner degradieren!
- Konditionale Verzweigungen durch `IF`–Statements innerhalb von Schleifen führen dazu, daß bei denjenigen Vektorelementen, für die die Bedingung der Abfrage nicht zutrifft und deren `ELSE`–Block keine Anweisungen enthält, der Vektor–Computer keine Anweisungen durchführen kann. Da beim Verkehrsmodell haupsächlich mit Fahrzeugdichten von 0.0 bis 0.2 gearbeitet wird, sind im Mittel nur etwa ein Zehntel der vorhandenen Sites mit Fahrzeugen gefüllt. Für den Vektor–Computer bedeutet dies, daß er nur mit etwa einem Zehntel seiner tatsächlichen Leistung arbeiten kann.

Das einspurige Modell und auch das zweispurige symmetrische Modell wurde auf dem Vektorrechner NEC–SX3 im Kölner Rechenzentrum implementiert. Die Rechenleistungen waren in Kombination mit Bit–Codierung (siehe 3.10) exzellent. Die

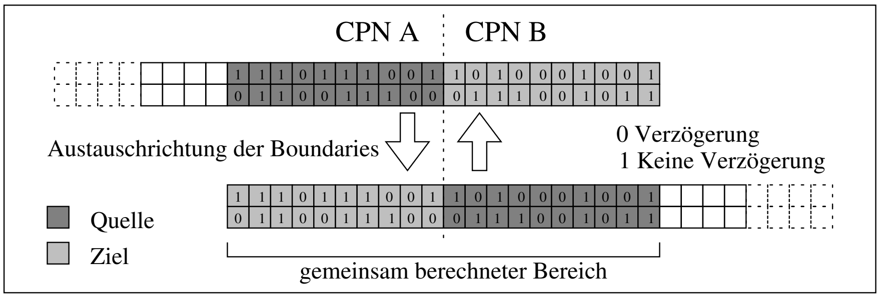

Abbildung 6.2: *Boundaries* — Die Nullen und Einsen stellen die Zufallszahlen dar, die vor dem Austausch der Boundaries ermittelt werden. Sie legen beim anschließenden Update sowohl in der Quell–CPN als auch in der Ziel–CPN das Verhalten der mit ihnen markierten Fahrzeuge beim Verzögerungsschritt fest.

6.1.4 Parallellisierung und Boundaries

Die Implementation des Systems mit periodischen Randbedingungen besteht auf dem SIMD[2]–Prinzip: das Programm ist auf allen CPNs gleich, nur die von der CPN behandelten Daten sind unterschiedlich. Bei der Aufteilung das Gesamtsystems befinden sich an den Übergangsstellen zwischen den CPNs Boundaries, die zwischen den CPNs über Nachrichten ausgetauscht werden. Dabei ist das Ziel, die Anzahl der Nachrichten möglichst auf eine pro Boundaries, pro Richtung und pro Zeitschritt zu begrenzen. Folgende Punkte sind zu beachten:

- Die *Breite* der Boundaries ist von zwei Parametern abhängig. Zum einen
  - vom Maximum an Sites, die im Algorithmus "vorausgeschaut" (n_{succ}) bzw. "zurückgeschaut" (n_{pred}) werden, und die Anzahl von Sites, die ein Fahrzeug in einem Zeitschritt durchlaufen kann (n_v), als auch
  - von der Anzahl von Unterschritten n_t, in die ein Zeitschritt aufgeteilt ist.

  Die Breite der Boundaries ergibt sich zu $n_b = n_t \max(n_{pred}, n_{succ}, n_v)$. Im einspurigen Modell ist $n_{pred} = 0$, $n_{succ} = n_v = 5$ und $n_t = 1$, was zu der Breite 5 führt.

- Die Sites, die sich innerhalb der Boundaries befinden, werden von beiden CPNs berechnet. Wenn Fahrzeuge von CPN A zur CPN B fahren, so macht CPN A innerhalb der Boundaries den Update für die Fahrzeuge, die die CPN verlassen. Den gleichen Update für die gleichen Fahrzeuge macht CPN B für die neu eintreffenden Fahrzeuge. Im Fall eines deterministischen Modells bereitet das keine Probleme, da die Regeln auf beiden CPNs identisch sind. Im vorliegenden nichtdeterministischen Modell, muß jedoch gewährleistet werden, daß die zufällige Verzögerung der Fahrzeuge in Regel $S3$ auf beiden CPNs *identisch* ist (siehe dazu Abb. 6.2). In der vorliegenden Implementation wurde dies erreicht durch Integration der Zufallswerte in die Boundaries. Die Entscheidung, ob ein Fahrzeug verzögert wird oder nicht, fällt also *vor* dem eigentlichen Zeitschritt bei Austausch der Boundaries und nicht *während* der Zeitschritts.

[2] Single Instruction Multiple Data

6.1.5 Kombinierte Verfahren

Die schnellsten Laufzeiten für einspurige und symmetrische, zweispurige Simulationen wurden durch die gitterorientierte Implementation auf einem Vektor–Computer erreicht. Der Nachteil der geringen Auslastung durch kleine Dichten, die sich bei beiden Einzelverfahren als nachteilig erweist, fällt dabei nicht *doppelt* ins Gewicht, sondern nur *einfach*. Das bedeutet, daß durch die Vektorisierung die theoretisch mögliche Beschleuningung (64 bei der NEC SX–3) auch praktisch fast voll erreicht wird.

Beim asymmetrischen Modell ist jedoch die Codierung der Regeln so aufwendig und fehleranfällig, daß in dieser Arbeit alle asymmetrischen Simulationen mit einer nicht–bitorientierten Version des Algorithmus durchgeführt wurden.

6.1.6 Versuch einer kanonischen Vektorisierung

Im Rahmen dieser Arbeit wurde ermittelt, welche Leistungsverluste auftreten, wenn die Verkehrssimulation in der sequentiellen Version für Workstations auf kanonische Weise auf einen Vektorcomputer übertragen wird. Der Source–Code sollte dabei so aufgebaut sein, das er später auf beiden Rechnersystemen lauffähig ist, und auf der Workstation nur geringe Leistungsverluste gegenüber der dedizierten Version auftreten. Als einfaches Bespiel diente das einspurige Modell mit Systemlänge L und die NEC als Vektorrechner mit Vektorlänge $v = 64$. Das System wurde in v Stücke der Länge L/v unterteilt (siehe Abb. 6.3). Dann wurden die Stücke den einzelnen Vektor-Einträgen zugeteilt, wobei an den Enden

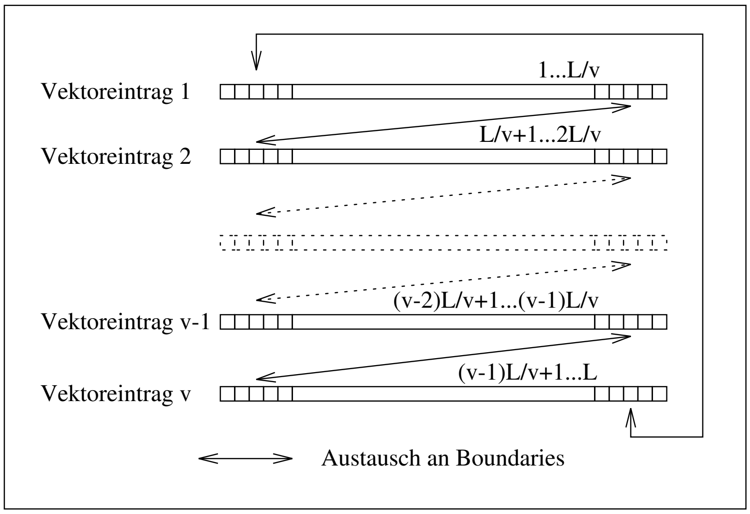

Abbildung 6.3: *Aufteilung des Systems auf Vektorelemente*

jeweils Boundaries eingerichtet wurden. Für jeden Abschnitt wurde zusätzlich ein eigener Zufallsgenerator eingerichtet. Für das Regelwerk der Simulation besteht kein Unterschied zwischen den künstlichen Boundaries der Vektorisierung und den Boundaries, die an CPN–Grenzen entstehen, so daß der sequentielle Source–Code übernommen werden konnte[3]. Bei einer Dichte von $\varrho = 0.1$ ergab sich eine Geschwindigkeitssteigerung von nur ca. Faktor 2.5 gegenüber der dedizierten Workstation–Implementation. Deswegen wurde dieses Verfahren nicht weiter verfolgt.

[3] nach Einführung einer weiteren Dimension bei den Feldern für den Bezug auf den Vektor–Eintrag

6.2 Zweispuriges Modell

Das zweispurige Modell wurde für die NEC SX–3 und die Parallel–Rechner in FORTRAN 77 implementiert. Dabei wurden im ersten Fall einige FORTRAN 90 Erweiterungen bezüglich Boolescher Operationen und im zweiten Fall die systemspezifischen Message-Passing-Bibliotheken verwendet.

6.2.1 Probleme bei listenorientierter Programmierung

Bei der listenorientierten Implementation von zweispurigen[4] Modellen können zwei Wege gegangen werden:

- Die Fahrzeuge beider Spuren befinden sich entsprechend ihrer Position auf der Strecke sortiert in der gleichen Liste, wobei bei benachbarten Fahrzeugen eines der Fahrzeuge nach einer willkürlich zu bestimmenden Konvention zuerst steht. Der notwendige Zugriff auf Vorgänger macht ein *doppeltes Verketten* notwendig. Beim Update der Geschwindigkeiten bewegen sich Fahrzeuge aneinander vorbei, so daß Elemente in der Liste umplaziert werden. Dies ist mit einem großen Zeitaufwand verbunden. Zusätzlich sind nach einer gewissen Zeit die Listenelemente so zufällig im Speicher verteilt, daß das Cache des Rechners die Zugriffe auf die Listen nur schlecht puffern kann.
- Die Fahrzeuge beider Spuren werden auf getrennten Listen verwaltet. Damit auf Nachfolger und Vorgänger der anderen Spuren zugegriffen werden kann, müssen jeweils Verweise gepflegt werden, die sich bei jedem Update ändern können. Die Lösung hat sonst alle obigen Nachteile und als einzigen Vorteil, daß hier das Einzelspur–Update so schnell wie im einspurigen Modell durchgeführt werden kann.

Für die in dieser Arbeit verwendeten Algorithmen wird deswegen nur die gitterorientierte Implementation verwendet.

6.2.2 Boundaries

Für den Fall der zweispurigen Simulationen sind die Parameter (siehe 6.1.4) zur Ermittlung der Boundary–Länge in Tab. 6.2 zusammengefaßt.

Modell	n_{pred}	n_{succ}	n_v	n_t	n_b
symmetrisch	5	5	5	2	10
asymmetrisch	5	10	5	2	20

Tabelle 6.2: *Boundary–Längen*

6.2.3 Ermittlung der Nachfolger und Vorgänger

Im symmetrischen Modell muß für ein Fahrzeug bis zu fünf Sites, im asymmetrischen Modell sogar bis zu zehn Sites vorausgeschaut werden, um die Bedingungen für den Spurwechsel zu überprüfen. Da das Feld vom letzten Site bis zum ersten Site bearbeitet wird,

[4] oder mehrspurigen

können die Positionen der Vorgänger auf beiden Spuren während des Durchlaufs ständig als Index-Wert mitgeführt werden. Damit verringert sich der Aufwand beim Vorausschauen auf den Vergleich von augenblicklicher Position zur gespeichertern Vorgänger–Position. Beim Zurückschauen ist jedoch kein Index–Wert verfügbar[5], so daß explizit alle betroffenen Sites abgefragt werden müssen. Dies bedeutet jedoch keine ernstzunehmende Verlangsamung, da durch das Vorausschauen der größte Teil der potentiellen Spurwechselkandidaten schon eliminiert wurde und damit das Zurückschauen nur selten aufgerufen wird.

6.3 Netzmodell

Das Netzmodell ist in ANSI–C implementiert und kann zum Beispiel mit Hilfe des GNU–C++–Compilers ab Version 2.4.5 kompiliert werden. Zur Zeit existieren Portierungen für PVM 3.x sowie für die Message–Passing–Library des Paragon XP/S 10 bzw. IPSC von Intel.

6.3.1 Ausnutzung des Caches

Bei Verwendung von Workstations oder von Parallel–Rechnern, deren CPNs auf Workstations basieren, muß zum Erreichen der optimalen Geschwindigkeit das Cache des Rechners gut ausgenutzt werden. Dies ist im Normalfall bereits durch Vermeidung von verketteten Listen gewährleistet. Als Speicherbild ist das ein– oder mehrdimensionale *Feld* vorzuziehen, das am besten durch die Algorithmen *linear* durchlaufen wird. In *PAMINA* werden außer bei der Datenaufbereitung nur lineare Felder verwendet. Bei der Anwendung der Regelwerke werden nicht einzelne Straßenabschnitte betrachtet, sondern Mengen von Abschnitten, die zu Pfaden (siehe 4.5.4) zusammengefaßt wurden, welche ihrerseits linear im Speicher abgelegt sind. Alle Transferstrecken einer CPN werden zu einer Kette aneinandergereiht (siehe 6.3.3).

6.3.2 Codierung der Fahrzeuge und Sites

Bei der Implementation des gitterorientierten Algorithmus wurde versucht, *alle* Informationen über ein Site in einem Computer–Word von 32 Bit Breite unterzubringen. Die einzelnen Bits zerfallen dabei in drei Gruppen. Zum einen gibt es *fahrzeugbezogene* Daten, die das Fahrzeug auf seinem Weg begleiten, zum anderen *gitterplatzbezogene* Daten, die am jeweiligen Site verbleiben. Die dritte Gruppe der *temporären* Daten wird nur zwischen erstem und zweitem Teilschritt des Updates benutzt. Durch Bitmasken werden die einzelnen Informationen extrahiert. In Tab. 6.3 ist eine Aufstellung zu sehen.

6.3.3 Konzept der Zeiger

Nach der Verteilung des Netzes auf die CPNs werden auf jeder CPN vier Felder aufgebaut, die während der Simulation hauptsächlich verwendet werden:

- Das Feld `Paths` enthält Informationen, die zur Verwaltung der zu Pfaden zusammengefügten Straßenabschnitte notwendig sind. Jeder Pfad enthält einen Zeiger auf seine dynamisch belegten Sites.

[5] sonst müsste das Feld vom ersten bis zum letzten Site bearbeitet werden
[6] z.Z. nicht benutzt

Name	bits	f/g/t	Bedeutung
v_c	3	f	Augenblickliche Geschwindigkeit
v_d	3	f	Wunschgeschwindigkeit
Vehicle	1	f	Site enthält Fahrzeug
Slow[6]	1	f	Zusätzliches Flag für langsame Fahrzeuge
LeftRoute	1	f	gesetzt, falls linke Route markiert wurde
RightRoute	1	f	gesetzt, falls rechte Route markiert wurde
LeftRouteMarked	1	f	gesetzt nach Passieren des Markierungsbereiches
RightRoutemarked	1	f	gesetzt nach Passieren des Markierungsbereiches
ChangeLeft	1	f	Wechsel auf linke Spur erforderlich
ChangeRight	1	f	Wechsel auf rechte Spur erforderlich
v_{max}[6]	3	g	Höchstgeschwindigkeit des Sites
$prob_d$	3	g	Index auf Verzögerungswahrscheinlichkeiten
Blocked	1	g	Site ist blockiert
SuppressChange	1	g	Überholverbot
Random	1	g	gesetzt mit Verzögerungswahrscheinlichkeit
PanicChange	1	g	Panic–Lane–Change aktiv
NoAccelaration	1	g	unterdrückt Beschleunigung
Boundary	1	g	Markierung der Boundaries (für korrekte Statistik)
ChangeFromLeft	1	t	temporäre Markierung für den Wechsel von links
ChangeFromRight	1	t	temporäre Markierung für den Wechsel von rechts

Tabelle 6.3: *Codierung der Fahrzeuge und Sites*

- Das Feld `TransferSites` enthält alle zu einer Kette zusammengefügten Transferstrecken. Sie sind jeweils nur durch ein Site mit der Markierung `Blocked` getrennt.

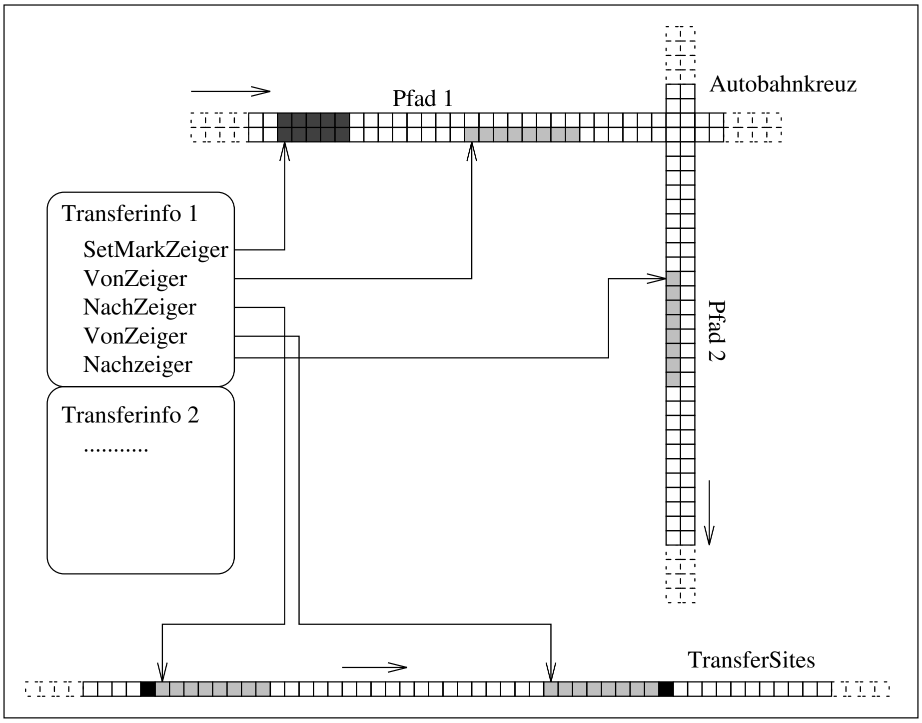

Abbildung 6.4: *Transferzeiger für eine Rechtsabbiegerspur am Kreuz*

Dadurch kann die Aktualisierung aller Transferstrecken durch einen einzigen Aufruf der einspurigen Simulation erfolgen.

- Das Feld `Transfers` enthält Informationen, die zur Absorption und Emission an Beschleunigungs– und Verzögerungsstreifen notwendig sind. Jeder Eintrag besitzt jeweils einen Zeiger auf das erste Site des Absorptionsbereiches im zweispurigen Abschnitt und auf die dazugehörige Position in den `TransferSites` sowie jeweils Zeiger auf das erste Site des Emmissionsbereiches im zweispurigen Abschnitt und auf die dazugehörige Position in den *TransferSites*. Siehe dazu das Beispiel einer Rechtsabbiegerspur an einem Autobahnkreuz in Abb. 6.4.
- Das Feld `Terminals` enthält Informationen, die zur Absorption und Emission an Endstücken notwendig sind. Jeder Eintrag enthält Zeiger auf das erste Site des Absorptions– und des Emmisionsbereiches.

Der ganze Austausch von Fahrzeugen an Netzpunkten wird also mit Hilfe eines einheitlichen Zugriffsmechanismus, nämlich Auswertung von Zeigern auf Sites, durchgeführt. Dies ist ein sehr effizientes Verfahren, da während der Simulation keinerlei Offset–Berechnungen anfallen.

6.3.4 PVM

Das in dieser Arbeit verwendete PVM ist ein Software–Paket, mit dessen Hilfe eine Gruppe von Einzelcomputern zu einer Recheneinheit mit verteiltem Speicher kombiniert werden kann. Die Kommunikation geschieht über normale Netzwerk–Hardware, wie sie in Workstation–Clustern heute üblich ist. PVM wurde 1989 am Oak Ridge National Lab ins Leben gerufen und wird heute von Mitarbeitern an mehreren amerikanischen Hochschulen gepflegt und weiterentwickelt. Das komplette Paket ist Public Domain. Für die Simulation mit *PAMINA* wurde PVM 3.2 verwendet. Allerdings ist sie auch unter PVM 3.0 lauffähig.

Kapitel 7

Diskussion

7.1 Zusammenfassung

In dieser Arbeit wurde versucht, die Idee des einfachen zellularen Automaten konsequent auf ein Netzwerk unter der besonderen Berücksichtigung des Straßenverkehrs zu erweitern. Es ist deutlich geworden, daß grundlegende Probleme, wie zum Beispiel der Flußzuwachs durch Geschwindigkeitsbeschränkungen, modelliert werden können und gleichzeitig durch Ausnutzung von Parallel–Rechnern noch reale Problemgrößen berechenbar sind. Die Simulation *PAMINA* versteht sich dabei nicht als Konkurrenz gegenüber den in 4.9 erwähnten oder ähnlichen Systemen. Dies ist auch auf Grund der Diskrepanz von zeitlichem Rahmen und Manpower überhaupt nicht möglich. Vielmehr sollte ein Beispiel für die vielseitige Verwendbarkeit der Konzeptes der zellularen Automaten gegeben werden.

Bei der Simulation des inhomogenen Netzes von Nordrhein–Westfalen wurde das interessante Phänomen festgestellt, daß sich im Gegensatz zur zweispurigen Version mit periodischen Randbedingungen keine schnelle Relaxation des Gesamtflusses einstellt. Vielmehr ergibt sich bei Dichten über ca. 0.04 ein komplexer zeitabhängiger Verlauf des Fundamentaldiagramms. Es taucht ein grundsätzlicher Unterschied im Verhalten dichter Netzbereiche (Kölner Ring) und einer freien Autobahnstrecke auf. Bei der letzteren wurden Schwingungen der Dichte und des Flusses mit regelmäßiger Periode gefunden, die im Übergangsbereich der Dichte 0.04 besonders stark sind. Obwohl nur mit trivialen Randbedingungen gearbeitet wurde, nämlich Vorgabe einer konstanten, globalen Dichte und Reflexion an den Endstücken, sind die Ergebnisse interressant. Besonders das an $1/f$–Rauschen erinnernde Verhalten des Flusses und der Dichte sollte in weiterführenden Arbeiten untersucht werden.

7.2 Regelwerk

Beim Entwurf eines Netzmodells auf der Basis zellularer Automaten taucht das gleiche Dilemma auf, das auch schon beim zweispurigen Modell aufgetreten ist: die verwendeten Regeln basieren zwar oft auf Erfahrungen, die im Umgang mit dem Straßenverkehr im Alltag gemacht wurden, und sollten deswegen im weitesten Sinne “vernünftig” sein. Es ist jedoch wegen der erheblichen Vereinfachung und Abstraktion der Regeln ohne Vergleich zur Realität schwer über deren Sinn oder Unsinn zu entscheiden. Daher werden oft Regeln so der Realität angepaßt, daß sie etwaige Meßdaten reproduzieren können. Dies steht im krassen Gegensatz zu dem Ansatz, dem Verkehrswissenschaftler folgen: Dort sind die Regeln an sich als bekannt vorauszusetzen. Sie sollen alle Fahrer– und Fahrzeugeigenschaften bis ins Detail modellieren. Stellt sich nachher heraus, daß Simulationsergebnisse nicht mit der Realität übereinstimmen, so kann lediglich der Fehler in der implementationstechnischen Formulierung der Regeln gesucht werden. Durch die ständige Erweiterung und Verfeinerung der Formulierungen wird das Regelwerk als Ganzes jedoch unüberschaubar. Der *Rechner* kann zwar diese Arbeit bewältigen, der *Mensch* als Lieferant der Regeln ist jedoch ihrem komplexen Wechselspiel hilflos ausgeliefert. Etwaige Fehler in der Programmierung sind nicht mehr greifbar. Problematisch wird es dann, wenn die Simulation nicht nur zur Bestätigung vorhandener Daten dienen soll, sondern auch zur Projektion von zukünftigem Verkehrsaufkommen. Denn sobald Meßdaten fehlen, gibt es keine Grundlage mehr für Plausibilitätstests, die Fehler hätten aufzeigen können. Somit ist man in der gleichen Lage wie vorher mit den zellularen Automaten: Man hat einen Regelsatz, der die Realität meistens reproduziert, es gibt jedoch keine Garantie dafür. Also brachte unter Umständen der gesamte Aufwand keine direkten Vorteile, sondern nur den Nachteil einer erheblich langsameren Rechengeschwindigkeit.

Wahrscheinlich liegt die Lösung des Problems in der Kombination vom Konzept der zellularen Automaten mit dem detaillierter Regelwerke. Ziel bei der Implementation sollte sein, Regelwerke nur dann komplex zu gestalten, wenn sich herausgestellt hat, daß einfachere zellulare Automaten nicht zum gewünschten Ziel führen. Man könnte somit einen ausgewogenen Kompromiß aus Geschwindigkeit und Sicherheit (durch gute Wartbarkeit) auf der einen Seite und Realitätsnähe auf der anderen Seite finden.

7.3 Statistik und mesoskopische Simulation

Die in 4.8 beschriebenen Probleme betreffen hauptsächlich die Implementation des Programms. Sie sind beide mit entsprechendem Aufwand zu lösen, stellen also kein unüberwindliches Hindernis dar.

In Bezug auf die Vorhersage von Verkehrsaufkommen mit Hilfe von mikroskopischen Modellen ist jedoch folgendes zu vermerken: Es hat sich gezeigt, daß realistisches Verhalten in mikroskopischen zellularen Verkehrsmodellen nur dann erhalten wird, wenn der Zufall (zum Beispiel über die Verzögerungswahrscheinlichkeit) das System mitbeeinflußt. Diese Eigenschaft bringt aber auch Nachteile mit sich. Ausgehend von identischen Anfangskonfigurationen kann sich das mikroskopische Modell aufgrund kleiner Dichteschwankungen vollkommen unterschiedlich entwickeln. Soll nun eine Vorhersage für einen bestimmten Streckenabschnitt gewonnen werden, so müssen zumindest mehrere Simulationen durchgeführt und über ihre Ergebnisse gemittelt werden, um eine aussagekräftige Prognose zu erstellen. Dadurch vermindert sich natürlich die Berechnungsgeschwindigkeit des Systems, was wiederum zu einer Verkürzung des möglichen Planungshorizontes führt.

Als Alternative zur mikroskopischen Simulation steht die *mesoskopische*[1] Simulation zur Verfügung. In ihr werden nicht einzelne Fahrzeuge modelliert, sondern nur noch Dichte– und Geschwindigkeitsprofile. Die Dynamik ist hauptsächlich durch Zusammenhänge von Dichte, Geschwindigkeit und Kapazität bestimmt (siehe [11]), die in Fundamentaldiagrammen formuliert werden. Letztere sind wiederum statistische Mittelungen aus mikroskopischen Simulationen oder aus im Straßenverkehr ermittelten Daten. Der Ablauf der Simulation ist

[1] Der Begriff *mesoskopisch* soll hier den Übergangsbereich zwischen mikroskopischen und makroskopischen Größenordnungen bezeichnen

dann deterministisch, so daß im Prinzip *ein* Lauf ausreicht, um eine Vorhersage zu bestimmen. Sie stellt also als deterministische Simulation von gemittelten Daten sozusagen das Pendant zur gemittelten Vorhersage individueller nichtdeterministischer Daten dar. Durch eine Vergröberung der Ortsauflösung auf ca. 100 [m] und der Zeitauflösung auf ca. 3 [s] ist eine erhebliche Geschwindigkeitssteigerung gegenüber dem vorgestellten mikroskopischen Modell möglich. Vor kurzem wurde von Daganzo [11] ein Modell vorgestellt, das eine Diskretisierung des hydrodynamischen Modells repräsentiert und sich als vielversprechender Ansatz für die Behandlung des Verkehrsmodells in mesoskopischer Auflösung herausgestellt hat. Es hat dabei die interessante Eigenschaft, daß die Dynamik der Fahrzeuge auf einem Streckenabschnitt nur von der Belegung des folgenden Streckenabschnitts zum vorgehenden Zeitschritt abhängt. Dadurch ist das Modell auf triviale Weise parallelisierbar, weil keine Abarbeitungsreihenfolge vorgeschrieben ist. Eine auf dem Modell von Daganzo beruhende Simulation des Straßennetzes von Nordrhein–Westfalen, die gleichzeitig die Technik des Virtual Shared Memory nutzt (siehe unten), ist in Vorbereitung [26].

Es bleibt zu zeigen, ob die Vertauschung der Reihenfolge von Mittelung und Simulation erlaubt ist, oder ob dadurch grundsätzliche systematische Fehler entstehen, die nur durch eine mikroskopische Simulation zu beheben sind. Zumindest in den Anwendungen der Simulation, bei denen es auf individuelle Fahrzeugeigenschaften ankommt, wird sich die mesoskopische Simulation nicht durchsetzen. Dazu gehören zum Beispiel:

- *Ermittlung von Fahrtzeiten und Wartezeiten:* Jedes Fahrzeug besitzt eine individuelle Uhr, die die Fahrzeit mißt, unter Umständen unterteilt in “freie Fahrt” und “Stau”.
- *Ermittlung der Umweltbelastung durch Abgase:* Hier wird eine genaue Kenntnis der Motorcharakteristika der verstrichenen Zeit seit Fahrtbeginn[2] benötigt.
- *Routenpläne:* Jedes Fahrzeug besitzt einen individuellen Routenplan.

7.4 Virtual Shared Memory

Bei der Programmierung von *PAMINA* wurde offensichtlich, daß die herkömmliche Implementationsmethode durch Message–Passing zwar im Prinzip ausreicht, in punkto Komfortabilität und Erweiterbarkeit aber erhebliche Schwachstellen besitzt. Jeder Datenaustausch zwischen CPNs muß einzeln programmiert werden. Das ist für lineare Datenstrukuren (einzelne Variablen oder Felder) einfach. Handelt es sich jedoch um dynamische Strukturen (Zeiger, Listen), so wird viel Detailarbeit notwendig, da einzelne Informationen auf der Sender–Seite in einem Puffer gesammelt und auf der Empfänger–Seite ebenso wieder verteilt werden müssen. Zeiger auf Speicherplätze verlieren ihre Gültigkeit bei Transfer auf eine andere CPN gänzlich. Als Ausweg ist das Konzept des Virtual Shared Memory (siehe zum Beispiel [18]) zu sehen, bei dem alle CPN auf den gleichen linearen Adressraum zugreifen. Die Verwendung von komplexen Datenstrukuren ist durch simple Weitergabe eines Zeigers möglich, da dieser, unabhängig vom Kontext, immer das gleiche Objekt referenziert. Intern wird der gesamte Speicherbereich in Seiten verwaltet, wobei jede CPN auf eine gewisse Anzahl von Seiten direkt zugreifen kann. Nur ein Zugriff auf eine nicht lokale Seite resultiert in einer Kommunikation zwischen den betroffenen CPNs. Das Virtual Shared Memory ist als eine Shell anzusehen, die die systemnahen Aufgaben der Kommunikation übernimmt und somit den Programmierer entlastet. Dabei ist jedoch zu beachten, daß der

[2] Kalte Motoren verbrauchen erheblich mehr Kraftstoff als warme!

Programmierer nicht von der Verantwortung befreit ist, auf eine hohe Lokalität der verwendeten Daten zu achten, da sonst unbedachte Zugriffe auf nichtlokale Speicherbereiche zu versteckter Kommunikation führen.

7.5 Ausblick

In *PAMINA* wird davon ausgegangen, daß für Run Mode 0 die Transferwahrscheinlichkeiten an Netzpunkten als Inputdaten vorliegen. Dies ist leider bis heute noch nicht flächendeckend verwirklicht. Die Messung von Fahrzeugströmen setzt die Installation von kostspieligen Zählschleifen mit verbundenen Zählstationen voraus. Zusätzlich müssen die Daten von den verstreut liegenden Meßstellen zu Zentralrechnern weitergeleitet und dort aufbereitet werden. Dies setzt eine EDV-technische Infrastruktur voraus, die nicht vorhanden ist. Deshalb wird in den nächsten Jahren die Entwicklung auf diesem Gebiet eher langsam verlaufen. Erschwerend kommt hinzu, daß manche deutsche Großkonzerne eher den Ausbau eines individuellen Fahrzeugleitsystems fördern. Sie zielen darauf ab, daß in Zukunft jedes Fahrzeug mit einer bidirektionalen Funkeinheit versehen ist, die die aktuelle Position des Fahrzeugs direkt (z.B. über Satellit) an Zentralrechner weitergibt, wo *individuell* die Fahrtroute zum vorher übermittelten Ziel bestimmt wird. Dabei werden staugefährdete Gebiete mit berücksichtigt, so daß die Fahrzeit für den Einzelnen optimiert werden kann. Insofern wäre es sinnvoll, das vorliegende Modell zu einer Simulation mit *Routenplänen* auszubauen, in der jedem Fahrzeug eine individuelle Fahrtroute mit auf den Weg gegeben wird. Bei Veränderung der Verkehrslage während der Fahrt wird die Route entsprechend modifiziert.

Anhang A

Begriffe

A.1 Komplexitätstheorie

Zur Klassifizierung des Laufzeitverhaltens eines Algorithmus betrachtet man seine Ausführungsdauer[1] $t(l)$ in Abhängigkeit einer *Inputlänge* l. In den Verkehrssimulationen bieten sich entweder die Anzahl an Sites bzw. die Anzahl der Fahrzeuge als Inputlänge an. Man sagt nun, daß ein Algorithmus A bezüglich des Input I mit der Länge $l = l(I)$ die *Worstcase–Laufzeit* $O(f(l))$ besitzt, wenn es eine Konstante c und ein natürliche Zahl l_0 gibt, so daß

$$\bigwedge_{l>l_0} t(l) < cf(l).$$

Die Funktion $f(l)$ ist demnach asymptotisch die obere Schranke für die Ausführungszeit für genügend große Inputlängen. Man schreibt auch

$$A \in O(f(l)).$$

A.2 Graphentheorie

Für die Beschreibung der Netzstruktur werden Begriffe verwendet, die sich in der Graphentheorie für die exakte Beschreibung eines Graphen eingebürgert haben.

Knoten Ein *Knoten* ist ein beliebiges Element, dem man Eigenschaften zuordnen kann. Ein *Netzpunkt* ist ein *Knoten*, der als zusätzliche Eigenschaft eine Koordinate und unter Umständen einen Namen trägt (Auf– und Abfahrten, Dreiecke, Kreuze).

Kante Eine *Kante* ist jeweils (z.B. über eine Numerierung) zwei Knoten zugeordnet oder auch *adjazent*. Sie kann ebenfalls zusätzliche Eigenschaften besitzen. In *PAMINA* entspricht eine *Kante* einem Straßenabschnitt mit den zusätzlichen Informationen über Länge und Typ des Straßenabschnittes. Über sie kann der Verkehr zwischen beiden Knoten fließen, normalerweise in beide Richtungen.

gerichtet Eine *Kante* heißt gerichtet, wenn der Fluß auf der Kante nur in eine Richtung verlaufen darf. In diesem Fall sind die beiden Knoten nicht äquivalent, sondern es existiert ein *Von–Knoten* und ein *Nach–Knoten*.

[1] gemessen in einer willkürlichen Einheit, z.B. Sekunden

Graph Ein *Graph* besteht aus einer Menge von *Knoten* und einer Menge von *Kanten* auf diesen Knoten.

Teilgraph Ein *Graph* A heißt Teilgraph eines Graphen B, wenn die Knotenmenge von A Teilmenge der Knotenmenge von B und die Kantenmenge von A Teilmenge der Kantenmenge von B ist.

Valenz Man nennt die Anzahl der Kanten, die zu einem gegebenen Knoten adjazent sind, die *Valenz* des Knotens.

Verbindung Zwei Knoten heißen *verbunden*, wenn man von einem über eine Folge von Kanten zum anderen gelangen kann.

Zusammenhang Ein Graph heißt *zusammenhängend*, wenn je zwei Knoten in ihm verbunden sind. Man sagt ein Graph A zerfällt in n Zusammenhangskomponenten, wenn es eine Zerlegung von A in n Teilgraphen gibt, so daß alle Teilgraphen disjunkt[2] und zusammenhängend sind.

[2] Knoten– und Kantenmengen enthalten jeweils keine gemeinsamen Elemente

Anhang B

Daten, Datenstrukturen und Algorithmen

B.1 Beschaffenheit der Inputdaten

Das Programm *PAMINA* kennt 4 Inputdateien. Alle beginnen mit dem gleichen Basisnamen, an den eine spezifische Endung angehängt wird. Die Spalte *Run Mode* gibt an, in welchen Run Modes die jeweilige Input Datei benötigt wird. Folgende Aspekte sind zu beachten:

Endung	Run Mode	Bedeutung
`.nod`	0,1,2,3	Netzknoten (Nodes)
`.edg`	0,1,2,3	Netzkanten (Edges)
`.rat`	0	Absorptions– und Emissionsraten für Endstücke
`.trn`	0	Transferraten

Tabelle B.1: *Inputdateien*

- Die ASCII–Dateien dürfen keine Kommentare und keine Leerzeilen enthalten.
- Ein Eintrag in der Knoten– bzw. Kantendatei muß genau eine Zeile einnehmen. Felder sind durch Leerzeichen getrennt. Es gibt Standardfelder und optionale Felder. Alle Standardfelder müssen in der vorgegebenen Reihenfolge (siehe Spalte *Position* in den folgenden Tabellen) auftreten, bei den optionalen Feldern ist die Reihenfolge beliebig. Bei Angabe eines optionalen Feldes muß der *Token-Name* vorangestellt werden.
- Zeichenketten müssen mit doppelten Anführungszeichen umfaßt werden.
- Die logischen Numerierungen der Knoten und Kanten (siehe unten) brauchen nicht durchgehend zu sein. Die logischen Nummern müssen positiv oder gleich null sein.

B.1.1 Knoten

In Tab. B.2 sind die Felder der Knotendatei aufgelistet. Für die Koordinaten wird ein rechtwinkliges Koordinatensystem vorausgesetzt.

Token	Position	Typ	Bemerkung
	1	`unsigned long`	logische Nummer (für Verweise in Kanten)
	2	`signed long`	X–Koordinate in Metern
	3	`signed long`	Y–Koordinate in Metern
`N`	> 3	`string`	Name des Knotens

Tabelle B.2: *Felder der Knotendatei* `.nod`

B.1.2 Kanten

In Tab. B.3 sind die Felder der Kantendatei aufgelistet. Die Kanten sind gerichtet. Da alle

Token	Pos.	Typ	Bemerkung
	1	`unsigned`	logische Nummer
	2	`unsigned`	Verweis auf *VonKnoten*
	3	`unsigned`	Verweis auf *NachKnoten*
`N`	> 3	`string`	Name der Kante
`Source`	> 3	`int`	Ratentabellennummer, falls *Von–Knoten* Quelle
`Sink`	> 3	`int`	Ratentabellennummer, falls *Nach–Knoten* Senke
`Left`	> 3	`int`	Transfertabellennummer, für Wechsel nach links
`Right`	> 3	`int`	Transfertabellennummer, für Wechsel nach rechts
`L`	> 3	`int`	Länge der Kante in Metern
`T`	> 3	`int`	Typ der Kante[1]

Tabelle B.3: *Felder der Kantendatei* `.edg`

Strecken in der Simulation als bidirektional angenommen werden, müssen zwischen zwei Knoten entweder null oder zwei Kanten in entgegensetzter Richtung verlaufen. Selbst wenn die Eigenschaften beider Richtungen identisch sind, muß ein Repräsentant für jede Richtung in der Datei vorhanden sein.

B.1.3 Emissions– und Absorptionsraten

Die Datei `.rat` enthält Emissions– und Absorptionsraten in tabellarischer Form. In der ersten Zeile steht ein einzelner Wert, der die logische Nummer der darauffolgenden Tabelle angibt. Auf diese Nummer wird in den Feldern `Source` und `Sink` der Kanteneinträge verwiesen. In den 12 darauffolgenden Zeilen stehen jeweils 6 Werte, die die Anzahl der zu emittierenden bzw. absorbierenden Fahrzeugen für die sechs Wunschgeschwindigkeiten $v_d = 0 \ldots 5$ angeben. Wunschgeschwindigkeit $v_d = 0$ wird zur Zeit nicht verwendet. Eine Zeile repräsentiert 5 Minuten, so daß sich als maximale Gesamtzeit für die Simulation eine Stunde ergibt. Danach wird wieder bei der ersten Zeile begonnen. Es gibt zwei Raten, die zur Verfügung gestellt werden müssen. Sie gelten für Netzpunkte, die keine eigene Rate zugewiesen bekommen haben. Eine Erläuterung befindet sich in Tabelle Tab. B.4.

[1] siehe 4.3

LNr	Rate
0	Default–Quellen–Emmissionsrate
1	Default–Senken–Absorptionsrate

Tabelle B.4: *Default–Raten*

B.1.4 Transferraten

Die Datei `.trn` enthält Transferwahrscheinlichkeiten in tabellarischer Form. In der ersten Zeile steht ein einzelner Wert, der die logische Nummer der darauf folgenden Tabelle angibt. Auf diese Nummer wird in den Feldern `Left` und `Right` der Kanteneinträge verwiesen. In den 12 darauffolgenden Zeilen stehen jeweils 6 Werte, die die Wahrscheinlichkeit der zu transferierenden Fahrzeugen für die sechs Wunschgeschwindigkeiten $v_d = 0 \dots 5$ in *Promille* angeben. Wunschgeschwindigkeit $v_d = 0$ wird zur Zeit nicht verwendet. Eine Zeile repräsentiert 5 Minuten, so daß sich als maximale Gesamtzeit für die Simulation eine Stunde ergibt. Danach wird wieder bei der ersten Zeile begonnen. Es gibt fünf Transferwahrscheinlichkeiten, die zur Verfügung gestellt werden müssen. Sie werden für Netzpunkte benutzt, die keine eigene Wahrscheinlichkeit zugewiesen bekommen haben. Eine Erläuterung befindet sich in Tab. B.5.

LNr	Transferwahrscheinlichkeit
0	Default–Abfahrtswahrscheinlichkeit an Anschlüssen
1	Default–Links–Wahrscheinlichkeit an Kreuzen
2	Default–Rechts–Wahrscheinlichkeit an Kreuzen
3	Default–Rechts–Wahrscheinlichkeit bei Dreiecken
4	Default–Abfahrwahrscheinlichkeit bei Dreiecken

Tabelle B.5: *Default–Wahrscheinlichkeiten*

B.2 Zufallsgeneratoren

Für die Regel $S3$ wird ein Zufallsgenerator benötigt. Für die Simulation auf Workstations und Parallel–Rechnern wurden zwei Integer–orientierte Generatoren verwendet, die im Rahmen des statistischen Fehlers übereinstimmende Resultate liefern. Für die Simulation auf dem Vektorechner NEC SX–3 wurde ein bit–orientierter Generator verwendet.

B.2.1 Integer–orientierter Zufallsgenerator

FUNCTION Rand1
BEGIN
$Seed \leftarrow (Seed \times 1366 + 150889) \bmod 714025$
RETURN $Seed$
END

Algorithmus 3: Zufallsgenerator 1

Der Algorithmus 3 basiert auf der Kongruenzmethode. Die Variable *Seed* wird am Anfang mit einer beliebigen Zahl vorbelegt.

FUNCTION Rand2
  BEGIN
  IF $RndK = 0$ **THEN**
    $RndK \leftarrow 54$
  ELSE
    $RndK \leftarrow RndK - 1$
  IF $RndJ = 0$ **THEN**
    $RndJ \leftarrow 54$
  ELSE
    $RndJ \leftarrow RndJ - 1$
  $Magic[RndK] \leftarrow Magic[RndK] + Magic[RndJ]$
  RETURN $Magic[RndK]$ **AND** MAX_RAND
  END

Algorithmus 4: Zufallsgenerator 2

Das Feld $Magic[0\ldots53]$ in Algorithmus 4 wird einmal am Start mit beliebigen Werten gefüllt. In Abb. B.1 sind beide Zufallsgeneratoren in Vergleich zu sehen. Man erkennt, daß die Kurve des zweiten Generators etwas glatter verläuft als die des ersten. Sonst gibt keine entscheidenden qualitativen oder quantitativen Unterschiede. Daß die Kongruenzmethode in diesem Modell trotz ihrer kurzen Periode vergleichsweise gut funktioniert, liegt an der Tatsache, daß die Zufallszahlen nicht *ortsgebunden* verwendet werden, sondern *fahrzeuggebunden* im zweispurigen Fall und vollkommen wahlfrei im Netzmodell. Durch die unterschiedlichen Kombinationen an Fahrzeugen, die sich während des Simulationsablaufs ergeben, werden die Zufallszahlen in sich ständig veränderndem Kontext verwendet, so daß die effektive Periode viel länger ist.

B.2.2 Bit–orienter Zufallsgenerator

Beim bit–orientierten Zufallsgenerator wird vor dem Beginn der Simulation mit der Hilfe des Generators *Rand*1 ein `integer`–Feld erzeugt, in dem die Bits mit Wahrscheinlichkeit $prob_d$ gesetzt sind. Anschließend wird die Belegung des Zufalls–Bitfeldes überprüft, und wenn nötig korrigiert durch zufälliges Setzen (oder Löschen) einzelner Bits, so daß nachher das Verhältnis von gesetzten Bits zu vorhandenen Bits gleich $prob_d$ ist. Das Zufalls–Bitfeld *RndArr* ist fünf mal so groß ausgelegt wie das Datenfeld (Länge L), auf das es nachher angewandt wird. Vor jeder Anwendung wird ein Zufallswert mit *Rand*1 bestimmt, der den Anfangspunkt im Zufalls–Bitfeld angibt. Nach jeder 16. Anwendung wird das gesamte Bitfeld durcheinandergewürfelt über Algorithmus 5.

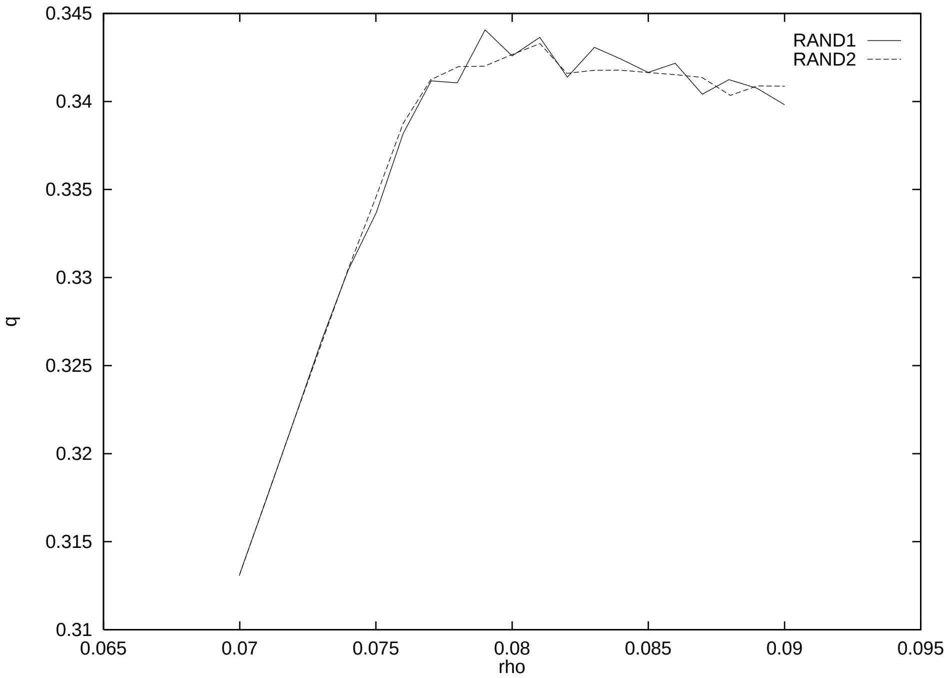

Abbildung B.1: *Vergleich der Zufallsgeneratoren*

PROCEDURE BitRandomize

BEGIN

$Last := 5 \times L - 147$

FOR $Unit := 1$ **TO** $Last$

$Temp[Unit] := RndArr[Unit]$ **XOR** $RndArr[Unit + 147]$

FOR $Unit := 1$ **TO** 147

$RndArr[Unit] := RndArr[Last + Unit]$

FOR $Unit := 1$ **TO** $Last$

$RndArr[Unit + 147] = Temp[Unit]$

END

Algorithmus 5: Schritte nach jeder 16. Anwendung

Zusätzlich werden nach L Anwendungen alle Worte des Bitfeldes um ein Bit zirkulär nach links rotiert. Dadurch wird die Peridode um Faktor 32 (Wortlänge) vergrößert.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] K. Agyemang-Duah and F.L. Hall. Some issues regarding the numerical value of freeway capacity. In Brannolte, editor, *Highway Capacity and Level of Service*. Balkema, Rotterdam, 1991.

[2] P. Bak, Ch. Tang, and K. Wiesenfeld. Self–organized criticality. *Physical review A*, 38, 1988.

[3] A. Baldus. Simulation von Zellular–Automaten–Modellen für Verkehrsfluß mit offenen Randbedingungen. *Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung*, 1993.

[4] W. Balz, R. Kühne, and H. Zackor. Untersuchungen des Verkehrsablaufs im Bereich der Leistungsfähigkeit und bei instabilem Fluß. Technical Report Heft 524, Forschung Straßenbau und Straßenverkehrstechnik, 1988.

[5] E. Ben-Naim, P.L. Krapivsky, and S. Redner. Kinetics of clustering in traffic flows. Center for Polymer Studies and Department of Physics, Boston University, Boston, MA 02215.

[6] O. Biham, A. Middleton, and D. Levine. Self-organization and a dynamical transition in traffic flow models. *Physical Review A*, 46, 1992.

[7] E. Cascetta, D. Inaudi, and G. Marquis. Dynamic estimators of origin–destination matrices using traffic counts. *Trans. Science*, pages 363–373, 1993.

[8] M. Cremer. *Der Verkehrsfluß auf Schnellstraßen.* Number 3 in Fachberichte Messen, Steuern, Regeln. Springer, 1979.

[9] M. Cremer and H. Keller. A new class of dynamic methods for the identification of origin–destination flows. *Trans. Res.*, 21B:117–132, 1987.

[10] J. A. Cuesta, F.C. Martínez, J.M. Molera, and A. Sánchez. Phase transitions in two–dimensional traffic flow models. submitted, 1993.

[11] C.F. Daganzo. The cell transmission model: A dynamic representation of traffic consistent with the hydrodynamic theory. University of California, Berkeley CA 94720, 1993.

[12] B. Harenberg, editor. *Aktuell 91.* Harenberg Lexikon Verlag, Dortmund, 1990.

[13] B. Harenberg, editor. *Aktuell 94.* Harenberg Lexikon Verlag, Dortmund, 1993.

[14] H.J. Herrmann, E. Flekkoy, K. Nagel, G. Peng, and G. Ristov. Density waves in granular flow. *Lecture Notes in Physics, Springer*, 1994.

[15] A. Latour. Simulation von Zellularautomaten–Modellen für Mehrspurverkehr. *Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung*, 1993.

[16] M.J. Lighthill and G.B. Whitham. On kinematic waves (part ii): A theory of traffic flow on long crowded roads. *Proc. R. Soc. London*, A229:317–345, 1955.

[17] A.D. May. *Traffic Flow Fundamentals.* Prentice Hall, 1990.

[18] R.G. Minnich. Mether–nfs: A modified nfs which supports virtual shared memory. Technical report, Supercomputing Research Center, 1993?

[19] T. Nagatani. Jamming transition in the traffic–flow with two–level crossings. *Phys. Rev. E*, 48:3290–3294, 1993.

[20] T. Nagatani. Self-organization and phase transition in traffic-flow model of a two-lane roadway. *J. Phys. Soc. Jpn.*, 62:1085, preprint 1993.

[21] K. Nagel. Fast low fidelity microsimulation of vehicle traffic on supercomputers. In *Transportation Research Board Meeting, Paper No. 94 09 01*, Jan. 1994.

[22] K. Nagel. Life-times of simulated traffic jams. *Int. J. Mod. Physics*, C, in press.

[23] K. Nagel and H.J. Herrmann. Deterministic models for traffic jams. *Physica A*, 199, 1993.

[24] K. Nagel and A. Schleicher. Microscopic traffic modeling on parallel high performance computers. *Parallel Computing*, 20:125–146, 1994.

[25] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *J. Physique I*, 2:2221, 1992.

[26] J.T. Pfenning. private communication.

[27] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[28] A. Schadschneider and M. Schreckenberg. Cellular automaton models and traffic flow. *Journal of Physics A*, 26, 1993.

[29] H. Schütt. Entwicklung und Erprobung eines sehr schnell, bitorientierten Verkehrssimulationssystems für Straßennetze. Technical report, Schriftenreihe der AG Automatisierungstechnik TU Hamburg–Harburg, 1991.

[30] D. Stauffer. Computer simulations of cellular automata. *J. Phys. A: Math. Gen.*, 24:909–927, 1991.

[31] S. Wolfram. Theory and applications of cellular automata. *Singapore: World Scientific*, 1988.

[32] B.J.N. Wylie, D. McArthur, and M.D. Brown. Paramics parallelisation schemes. Technical report, University of Edinburgh, 1992.

[33] H. Zackor. Beurteilung verkehrsabhängiger Geschwindigkeitsbegrenzungen auf Autobahnen. Technical Report Heft 128, Forschung Straßenbau und Straßenverkehrstechnik, 1972.

[34] H. Zackor. Untersuchung zur Verkehrsführung in Autobahnknotenpunkten bei Einsatz dynamischer Betriebssysteme. Technical Report Heft 255, Forschung Straßenbau und Straßenverkehrstechnik, 1978.