

# **Ein Simulationsmodell für Öffentlichen Personennahverkehr mit regelbasiertem Verkehrsmanagement**

Inaugural-Dissertation  
zur  
Erlangung des Doktorgrades  
der Mathematisch-Naturwissenschaftlichen Fakultät  
der Universität zu Köln

vorgelegt von  
Daniel Lückerath  
aus Köln

Köln 2017

Berichterstatter:  
(Gutachter)

Prof. Dr. Ewald Speckenmeyer  
Prof. Dr. Rainer Schrader

Tag der mündlichen Prüfung:

01.12.2016

## Danksagungen

Mein besonderer Dank geht an meinen Doktorvater Prof. Dr. Ewald Speckenmeyer, der mir die Möglichkeit gegeben hat im Rahmen des CATS-Projekts promovieren zu können. Durch seine engagierte Betreuung, richtungsweisenden Impulse, permanente Diskussionsbereitschaft (nicht nur auf das Thema der Dissertation bezogen) und sein unerschütterliches Vertrauen habe ich fachlich aber insbesondere auch menschlich extrem viel gelernt.

Mein Dank gilt weiterhin Prof. Dr. Rainer Schrader, der sich dazu bereit erklärt hat, die Arbeit als Zweitgutachter zu prüfen.

Meiner Kollegin Gabriele Eslamipour und meinen Kollegen Patrick Kuckertz, Manuel Molina Madrid, Prof. Dr. Bert Randerath, Dr. Oliver Ullrich, Alexander van der Grinten, Felix Werth und Dr. Andreas Wotzlaw danke ich für die stets gute, über das bloß kollegiale hinweggehende, Zusammenarbeit und die – hoffentlich gegenseitigen – Impulse die unsere gemeinsamen Diskussionen gefördert haben.

Sebastian Franz, Luise Lückerath und Oliver Ullrich haben die Arbeit kritisch durchleuchtet, herzlichen Dank dafür! Alle noch enthaltenen Fehler nehme ich vollständig auf meine Kappe.

Ein spezieller Dank geht noch an meinen Kollegen Oliver Ullrich für die stets anregende gemeinsame Arbeit an Artikeln, Vorträgen und Vorlesungen, seinen Willen (und sein Durchhaltevermögen) sich auch die absurdesten meiner Ideen anzuhören und für die unzähligen gemeinsamen Nachtsitzungen im Büro, die ohne ihn unerträglich gewesen wären.

Zuletzt geht mein tiefempfundener Dank an meine Familie, die meine Entscheidung zu promovieren wie selbstverständlich mitgetragen hat und auf deren Rückhalt ich mich, wie bereits während meines gesamten Studiums, blind verlassen konnte.

## Abstract

Until recently the project *Computer Aided Traffic Scheduling (CATS)* was only concerned with the design and implementation of methods and software tools that enable transport providers to generate, simulate and evaluate tram schedules, assisting them during the tactical planning phase.

During the work on the project the desire for a simulation software arose, which would be able to represent multi modal public transit systems and could assist transport planners not only during the tactical planning phase but also during daily operations.

This thesis deals with three main goals:

At first, an optimization software for the generation of timetables with high service regularity for multi modal public transit systems is developed, which is also able to consider interdependencies within and between different modes of transportation, e.g. transfer connections. In addition, the software can incorporate other economic, operative, and political requirements during the optimization to assure feasibility of the resulting timetables during daily operations. The correctness and efficiency of the optimization model is demonstrated by applying it to models of artificial transit networks as well as to a model of the bus network of Cologne from 2001.

Furthermore, a new event-based, mesoscopic simulation software for multi modal public transit systems is developed, which can be used to evaluate timetable quality. The individual components of the software are designed in such a way that future extensions of the model, e.g. new modes of transportation, can be incorporated easily. The correctness and plausibility of the simulation model, even when using sparse data sets, is proven through theory and function based validation, using models of an artificial bus network as well as the tram network of Cologne from 2001. The efficiency of the simulation software is demonstrated via exemplary run-time analyses.

Thereupon, the simulation software is extended with capabilities to autonomously initiate rule-based traffic management measures, enabling transport planners to evaluate different traffic management strategies. The correctness of the implemented measures is demonstrated exemplarily by applying them to models of artificial and real-world transit networks.

Lastly, the applicability of the complete software package is demonstrated via its application to a typical use case of the complete public transit network of Cologne from 2001.

The thesis begins with an introduction to context, motivation and aims (chapter 1), followed by an introduction to the domain of public transit timetable generation as well as a review of selected existing literature (chapter 2). Afterwards, a new optimization model for the generation of timetables for multi modal public transit systems is presented, which is based on two models reviewed beforehand (chapter 3). The correctness and efficiency of the optimization software designed to solve the optimization model is proven via sample applications. After this, some general simulation techniques as well as existing simulation models specific to public transit systems are discussed (chapter 4), before a new simulation model for multi modal public transit systems is presented (chapter 5). Its correctness and efficiency is demonstrated by applying it to models of artificial and real-world transit networks. Then the simulation model is extended with capabilities to autonomously initiate rule-based traffic management measures. To this end a short introduction to the domain of traffic management for short- and long-distance transit systems as well as a review of selected existing literature is given (chapter 6), before the actual extensions of the model are presented (chapter 7). The correctness of selected extensions is demonstrated via sample applications. After completion of the analyses of single components of the software package, its applicability is demonstrated by applying it to a typical use case of the complete public transit network of Cologne from 2001 (chapter 8). The thesis concludes with a short summary and some thoughts on further research (chapter 9).

## Kurzzusammenfassung

Das Projekt *Computer Aided Traffic Scheduling (CATS)* zielte bisher auf den Entwurf und die Entwicklung von Methoden und Softwarewerkzeugen zur Generierung, Simulation und Bewertung von Stadtbahnfahrplänen, um Verkehrsplaner während der taktischen Planungsphase zu unterstützen

Im Zuge der Arbeit am Projekt entstand der Wunsch nach einem Simulationsmodul, das zur Abbildung von multi-modalem *Öffentlichen Personennahverkehr (ÖPNV)* in der Lage ist und Verkehrsplaner nicht nur während der taktischen Planungsphase, sondern auch während des operativen Betriebs unterstützen kann.

Im Zuge dieser Arbeit werden Ziele aus drei Bereichen bearbeitet:

Zunächst wird ein Optimierungsmodul zur Generierung von Fahrplänen mit hoher Serviceregularität für multi-modale Verkehrssysteme entworfen und entwickelt, das zusätzlich die Möglichkeit bietet Interdependenzen innerhalb und zwischen Verkehrsmodi, wie bspw. Umsteigeverbindungen, zu berücksichtigen. Um eine Anwendbarkeit der resultierenden Fahrpläne im täglichen Betrieb zu gewährleisten, besteht außerdem die Möglichkeit bereits während der Optimierung von Verkehrsplanern formulierte ökonomische, betriebliche und politische Bedingungen zu berücksichtigen. Anhand beispielhafter Anwendungen des Optimierungsmoduls auf Modelle künstlicher Nahverkehrsnetze sowie auf ein Modell des Kölner Busnetzes aus dem Jahr 2001 wird seine Korrektheit und Leistungsfähigkeit nachgewiesen.

Weiterhin wird ein neues ereignisbasiertes, mesoskopisches Simulationsmodul für multi-modalen ÖPNV entworfen und entwickelt, das zur Evaluation der Güte eingesetzter Fahrpläne verwendet werden kann. Die einzelnen Bestandteile des Moduls sind so gestaltet, dass zukünftige Erweiterungen, wie z.B. neue Verkehrsmodi, auf einfache Art und Weise möglich sind. Durch theorie- und funktionsbezogene Validierungen anhand von Modellen eines künstlichen Busnetzes sowie des Kölner Stadtbahnnetzes aus dem Jahr 2001 wird die Korrektheit des Simulationsmodells nachgewiesen und gezeigt, dass es auch bei Verwendung sehr spärlicher Datengrundlagen noch plausibles Verhalten aufweist. Die Leistungsfähigkeit des Simulationsmoduls wird anhand von beispielhaften Laufzeituntersuchungen nachgewiesen.

Das Simulationsmodul wird daraufhin um die Möglichkeit erweitert selbstständig Maßnahmen des Verkehrsmanagements anstoßen zu können, um Anwenderinnen in die Lage zu versetzen unterschiedliche Verkehrsmanagementstrategien evaluieren zu können. Die Korrektheit der implementierten Verfahren wird beispielhaft anhand einiger Anwendungen auf Modelle künstlicher und realer Verkehrsnetze nachgewiesen.

Zuletzt wird die Anwendbarkeit des resultierenden Gesamtprogrammpakets anhand eines typischen Anwendungsfalls für das Kölner Gesamt-ÖPNV-Netz aus dem Jahr 2001 nachgewiesen.

Die Arbeit beginnt mit einer Einführung in Kontext, Motivation und Ziele (Kapitel 1), woraufhin eine Einführung in das Gebiet der Fahrplanerstellung für den ÖPNV, sowie eine kurze Besprechung ausgewählter aus der Literatur bekannter Optimierungsmodelle folgt (Kapitel 2). Auf Basis zweier dort besprochener Modelle wird anschließend ein neues Optimierungsmodell zur Fahrplangenerierung für multi-modale ÖPNV-Systeme vorgestellt (Kapitel 3). Die zur Lösung des Optimierungsmodells entworfene Softwareanwendung wird dabei anhand einiger Beispielanwendungen auf ihre Korrektheit und Leistungsfähigkeit untersucht. Darauf folgend werden einige allgemeine Simulationsverfahren sowie diverse in der Literatur zu findende speziell für Verkehrssysteme entwickelte Simulationsmodelle besprochen (Kapitel 4), bevor ein neues Simulationsmodell zur Abbildung von multi-modalen ÖPNV-Systemen vorgestellt wird (Kapitel 5). Die Korrektheit und Leistungsfähigkeit der resultierenden Simulationsanwendung wird durch beispielhafte Anwendungen auf Modelle künstlicher und realer Nahverkehrssysteme untersucht. Im Anschluss wird das Simulationsmodell um die Möglichkeit erweitert selbstständig regelbasierte Verkehrsmanagementmaßnahmen einzuleiten. Dazu erfolgt zunächst eine kurze Einführung in die Grundlagen des Verkehrsmanagements im Nah- und Fernverkehr sowie eine Besprechung ausgewählter in der Literatur zu findender Ansätze (Kapitel 6), bevor die eigentlichen Erweiterungen des Modells besprochen werden (Kapitel 7). Ausgewählte Erweiterungen des Simulationsmodells werden dabei anhand weiterer Beispielanwendungen auf ihre Korrektheit und Plausibilität

untersucht. Nach Abschluss der Einzeluntersuchungen, wird die Anwendbarkeit des Gesamtprogramm-pakets abschließend demonstriert, indem ein typischer Anwendungsfall für ein Modell des gesamten ÖPNV-Netzes der Stadt Köln aus dem Jahr 2001 besprochen wird (Kapitel 8). Die Arbeit schließt mit einer Zusammenfassung des Erreichten und einem Ausblick auf weitere Forschungsmöglichkeiten (Kapitel 9).

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>10</b>
<b>Tabellenverzeichnis</b>	<b>12</b>
<b>Verzeichnis der Algorithmen</b>	<b>15</b>
<b>Abkürzungen und Symbole</b>	<b>16</b>
<b>1. Einführung</b>	<b>20</b>
1.1. Kontext und Motivation .....	20
1.2. Ziele der Arbeit .....	21
1.3. Vorgehen .....	22
<b>2. Erstellung von Fahrplänen für den Öffentlichen Personennahverkehr</b>	<b>23</b>
2.1. Einführung und Abgrenzung .....	23
2.2. Optimierungsmodelle zur Erstellung von Fahrplänen für den Öffentlichen Personennahverkehr .....	24
2.2.1. Ein Ansatz zur Maximierung zeitlicher Sicherheitsabstände .....	27
2.2.2. Ein Ansatz zur Maximierung zeitlicher Sicherheitsabstände unter Berücksichtigung verkehrsplanerischer Vorgaben .....	33
<b>3. Ein neuer Ansatz zur Generierung regulärer Fahrpläne unter Berücksichtigung verkehrsplanerischer Vorgaben</b>	<b>36</b>
3.1. Modell .....	36
3.2. Implementierung .....	40
3.3. Beschleunigung des Optimierungsverfahrens .....	40
3.3.1. Zerlegung des (Multi-)Linienkonflikt-Graphen in Zusammenhangskomponenten .....	41
3.3.2. Zerlegung der Menge der verkehrsplanerischen Vorgaben .....	41
3.4. Validierung .....	42
3.4.1. Experimente anhand künstlicher Testnetze .....	42
3.4.1.1. DCL 2014 .....	42
3.4.1.2. UCL 2015 .....	44
3.4.2. Vergleich mit einer realen Testinstanz .....	46
<b>4. Simulationsmodelle für Nahverkehrssysteme</b>	<b>52</b>
4.1. Einführung und Abgrenzung .....	52
4.1.1. Mechanismen zur Abbildung von Zeit .....	52
4.1.2. Allgemeine Simulationsmethoden .....	53
4.1.2.1. Zelluläre Automaten .....	53
4.1.2.2. Ereignisbasierte Simulation .....	55
4.1.2.3. Agentenbasierte Simulation .....	58
4.1.3. Klassifizierung von Modellen zur Verkehrssimulation .....	59
4.2. Simulationsmodelle für Individual- und Busverkehr .....	59
4.3. Simulationsmodelle für schienengebundenen Verkehr .....	68
4.4. Simulationsmodelle für multi-modalen Verkehr .....	73
<b>5. Ein neues Simulationsmodell für den Öffentlichen Personennahverkehr</b>	<b>75</b>

5.1. Modell.....	75
5.1.1. Das physische Netz.....	76
5.1.2. Fahrzeuge.....	78
5.1.3. Linien und Fahrten.....	83
5.1.4. Verwaltungsmodule.....	84
5.1.4.1. Fahrzeugmanager.....	84
5.1.4.2. Linienmanager.....	85
5.1.4.3. Dispatcher.....	86
5.1.5. Randomisierung.....	87
5.1.5.1. Fahrzeiten.....	87
5.1.5.2. Passagierwechsel.....	89
5.2. Implementierung.....	91
5.2.1. Das generische Simulationsframework.....	91
5.2.2. Die Anwendung zur Verkehrssimulation.....	93
5.3. Validierung.....	94
5.3.1. UCL 2015.....	95
5.3.2. Das Kölner Stadtbahnnetz aus dem Jahr 2001.....	111
5.3.3. Leistungsfähigkeit der Simulationsanwendung.....	121
<b>6. Verkehrsmanagement im Nah- und Fernverkehr</b>	<b>124</b>
6.1. Einführung und Abgrenzung.....	124
6.2. Modelle für das Verkehrsmanagement im Nah- und Fernverkehr.....	124
<b>7. Ein Simulationsmodell für regelbasiertes Verkehrsmanagement im Öffentlichen Personennahverkehr</b>	<b>132</b>
7.1. Modellerweiterungen.....	132
7.1.1. Störungen.....	132
7.1.2. Fahrten.....	133
7.1.3. Dispatcher.....	133
7.2. Strategien.....	134
7.2.1. Haltepunkt-Kontrolle.....	134
7.2.1.1. Haltestrategien.....	134
7.2.1.2. Dynamische Modifikation des Linienverlaufs.....	137
7.2.2. Störfallmanagement.....	141
7.2.2.1. Umleiten.....	142
7.2.2.2. Verkürzen.....	143
7.2.2.3. Überspringen.....	144
7.2.2.4. Auftrennen.....	145
7.3. Validierung.....	147
7.3.1. Haltepunkt-Kontrolle.....	147
7.3.1.1. Anti-Bunching.....	148
7.3.1.2. Dynamische Modifikation von Linienverläufen.....	150
7.3.2. Störfallmanagement.....	154
7.3.2.1. Auftrennen.....	154
<b>8. Anwendung auf das Gesamt-ÖPNV-Netz der Stadt Köln</b>	<b>158</b>
<b>9. Zusammenfassung und Ausblick</b>	<b>167</b>
9.1. Zusammenfassung.....	167
9.2. Ausblick.....	169

<b>A. Beschreibung des Datenbankmodells</b>	<b>170</b>
<b>B. Literaturverzeichnis</b>	<b>177</b>

## Abbildungsverzeichnis

Abbildung 2.1: Netzwerk DCL 2014 .....	28
Abbildung 2.2: Linienkonflikt-Graph zum Beispielnetzwerk DCL 2014 .....	30
Abbildung 2.3: Multi-Linienkonflikt-Graph zum Netzwerk DCL 2014 .....	31
Abbildung 2.4: Beispiel der Haltepunktreduktion für einen gemeinsamen Streckenabschnitt zweier Linien. Haltepunkte $s_4$ , $s_5$ und $s_6$ können durch Haltepunkt $s_3$ repräsentiert werden. Die zugehörige Äquivalenzklasse hat eine Kardinalität von vier .....	32
Abbildung 3.1: Abfahrten von Linie 1-B01, 2-B01 und 3-B01 an Haltepunkten 2 und 3 von DCL 2014 unter Verwendung unterschiedlicher Sicherheitsabstände. Dargestellt sind alle Abfahrten innerhalb einer Stunde .....	43
Abbildung 3.2: Beispielnetzwerk UCL 2015, dunkler unterlegte Haltestellen identifizieren Start-/Endhaltestellen der jeweiligen Linien .....	45
Abbildung 3.3: Nördliches, linksrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001 .....	48
Abbildung 3.4: Südliches, linksrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001 .....	49
Abbildung 3.5: Nördliches, rechtsrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001 .....	50
Abbildung 3.6: Südliches, rechtsrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001 .....	51
Abbildung 4.1: Von-Neumann- und Moore-Nachbarschaft (jeweils in Grau) in unterschiedlichen Gittergeometrien.....	54
Abbildung 4.2: Beziehung zwischen Ereignis und Aktivität .....	55
Abbildung 4.3: Schematische Darstellung des Ablaufs einer ereignisbasierten Simulation .....	56
Abbildung 4.4: Modellierung von Kanten und Abbiegebewegungen in Mezzo. Entnommen aus [16], S. 50.....	64
Abbildung 4.5: Klassendiagramm zur Erweiterung von Mezzo um Busse, entnommen aus [88], S. 898.....	65
Abbildung 4.6: Flussdiagramm des Simulationsablaufs für den ÖPNV in Mezzo, entnommen aus [88], S. 900.....	66
Abbildung 5.1: Modifizierte Ereignisverarbeitungsroutine.....	75
Abbildung 5.2: Verkürztes UML-Klassendiagramm zum Streckennetz.....	77
Abbildung 5.3: Verkürztes UML-Klassendiagramm für Fahrzeuge .....	79
Abbildung 5.4: Ereignisprozesskette für Bus-Entitäten .....	80
Abbildung 5.5: Ereignisprozesskette für Bahn-Entitäten. Rot umrandet: Unterschiedliche Abfolge im Vergleich zu Bus-Entitäten .....	81
Abbildung 5.6: Verkürztes UML-Klassendiagramm für Fahrten und Linien .....	83
Abbildung 5.7: Verkürztes UML-Klassendiagramm für Verwaltungsmodule .....	84
Abbildung 5.8: Ereignisprozesskette für Dispatcher-Entitäten .....	86
Abbildung 5.9: UML-Klassendiagramm des generischen Simulationsframeworks .....	91
Abbildung 5.10: Häufigkeitsverteilung der Abfahrtszeitabweichungen, UCL 2015 .....	97
Abbildung 5.11: Häufigkeitsverteilung der Abfahrten mit einer Verspätung von mehr als 60 Sekunden, UCL 2015 .....	97
Abbildung 5.12: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 1. Links: 1-B01, rechts: 1-B02.....	99
Abbildung 5.13: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 2. Links: 2-B01, rechts: 2-B02.....	99
Abbildung 5.14: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 3. Links: 3-B01, rechts: 3-B02.....	100
Abbildung 5.15: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 4. Links: 4-B01, rechts: 4-B02.....	100

Abbildung 5.16: Abweichungsverlauf, Linie 1-B01. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	101
Abbildung 5.17: Abweichungsverlauf, Linie 1-B02. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	101
Abbildung 5.18: Abweichungsverlauf, Linie 2-B01. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	103
Abbildung 5.19: Abweichungsverlauf, Linie 2-B02. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	103
Abbildung 5.20: Abweichungsverlauf, Linie 3-B01. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	104
Abbildung 5.21: Abweichungsverlauf, Linie 3-B02. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	104
Abbildung 5.22: Abweichungsverlauf, Linie 4-B01. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	106
Abbildung 5.23: Abweichungsverlauf, Linie 4-B02. Links: UCL2015 <sup>+</sup> , rechts: UCL2015 <sup>-</sup> .....	106
Abbildung 5.24: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1111, UCL 2015 ..	108
Abbildung 5.25: Durchschnittlicher, kumulierter Anteil reduzierter Sicherheitsabstände, UCL 2015	108
Abbildung 5.26: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1091, UCL 2015 ..	108
Abbildung 5.27: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1121, UCL 2015 ..	109
Abbildung 5.28: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1131, UCL 2015 ..	109
Abbildung 5.29: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1142, UCL 2015 ..	109
Abbildung 5.30: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1102, UCL 2015 ..	109
Abbildung 5.31: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1062, UCL 2015 ..	109
Abbildung 5.32: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1052, UCL 2015 ..	109
Abbildung 5.33: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1043, UCL 2015 ..	110
Abbildung 5.34: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1031, UCL 2015 ..	110
Abbildung 5.35: Häufigkeitsverteilung der größeren Abfahrtszeitabweichungen, KVB-T 2001 .....	112
Abbildung 5.36: Häufigkeitsverteilung der Abfahrtszeitabweichungen, KVB-T 2001 .....	112
Abbildung 5.37: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 6-T01, KVB-T 2001 ..	114
Abbildung 5.38: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 6-T02, KVB-T 2001 ..	114
Abbildung 5.39: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 15-T02, KVB-T 2001	115
Abbildung 5.40: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 17-T01, KVB-T 2001	115
Abbildung 5.41: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 19-T02, KVB-T 2001	116
Abbildung 5.42: Abweichungsverlauf, Linie 6-T01, Fahrplan KVBT2001 <sup>+</sup> .....	117
Abbildung 5.43: Abweichungsverlauf, Linie 6-T01, Fahrplan KVBT2001 <sup>-</sup> .....	117
Abbildung 5.44: Abweichungsverlauf, Linie 6-T02, Fahrplan KVBT2001 <sup>+</sup> .....	118
Abbildung 5.45: Abweichungsverlauf, Linie 6-T02, Fahrplan KVBT2001 <sup>-</sup> .....	118
Abbildung 5.46: Abweichungsverlauf, Linie 15-T02, Fahrplan KVBT2001 <sup>+</sup> .....	119
Abbildung 5.47: Abweichungsverlauf, Linie 15-T02, Fahrplan KVBT2001 <sup>-</sup> .....	119
Abbildung 5.48: Abweichungsverlauf, Linie 17-T01, Fahrplan KVBT2001 <sup>+</sup> .....	120
Abbildung 5.49: Abweichungsverlauf, Linie 17-T01, Fahrplan KVBT2001 <sup>-</sup> .....	120
Abbildung 5.50: Abweichungsverlauf, Linie 19-T02, Fahrplan KVBT2001 <sup>+</sup> .....	121
Abbildung 5.51: Abweichungsverlauf, Linie 19-T02, Fahrplan KVBT2001 <sup>-</sup> .....	121
Abbildung 5.52: Durchschnittliche Laufzeit pro Phase in Abhängig der Zahl der durchzuführenden Simulationsläufe, UCL 2015 .....	123
Abbildung 7.1: UML-Klassendiagramm für die Erweiterung des Simulationsmodells um Störungen .....	132
Abbildung 7.2: Beispiel zur Fahrtverkürzung, bei dem die notwendigen, aber nicht die hinreichenden Bedingungen erfüllt sind .....	141
Abbildung 7.3: Abfahrtszeiten aller im Messzeitraum simulierter Fahrten der Linie 1-T01 (Richtung Bensberg) an den Haltepunkten der Linie, für einen typischen Simulationslauf ohne Störfallmanagement, KVB-T 2001. Gestrichelt: Start- bzw. Endzeitpunkt der Störung .....	155
Abbildung 7.4: Abfahrtszeiten aller im Messzeitraum simulierter Fahrten der Linie 1-T01 (Richtung Bensberg) an den Haltepunkten der Linie, für einen typischen Simulationslauf mit Störfallmanagement, KVB-T 2001. Gestrichelt: Start- bzw. Endzeitpunkt der Störung .....	156

## Tabellenverzeichnis

Tabelle 3.1: Kenngrößen der untersuchten Streckennetze .....	42
Tabelle 3.2: Abfahrtszeiten an Starthaltepunkten mit und ohne Aufteilung, DCL 2014 .....	44
Tabelle 3.3: Sicherheitsabstände an Haltepunkten mit und ohne Aufteilung, DCL 2014.....	44
Tabelle 3.4: Beispielhafte Umsteigeverbindungen zwischen Linien an Haltepunkten, UCL 2015 .....	45
Tabelle 3.5: Generierte Abstände für Umsteigeverbindungen, UCL 2015 .....	46
Tabelle 3.6: Durchschnittliche Laufzeit in Sekunden, mit Zerlegung in Zusammenhangskomponenten .....	47
Tabelle 3.7: Durchschnittliche Laufzeit in Sekunden, ohne Zerlegung in Zusammenhangskomponenten .....	47
Tabelle 3.8: Anzahl Haltepunkte, die einen spezifischen Sicherheitsabstand erreichen, Köln 2001 .....	47
Tabelle 3.9: Kennzahlen für die Optimierung der Streckennetze .....	48
Tabelle 4.1: Von Lückemeyer in [58] identifizierte Ereignistypen.....	70
Tabelle 4.2: Von Lückerath in [60] identifizierte Ereignistypen.....	72
Tabelle 5.1: Simulationsereignistypen zur Simulation von Fahrzeugen des ÖPNV .....	79
Tabelle 5.2: Simulationsereignistypen zur Abbildung des Verkehrsmanagement im Regelbetrieb .....	86
Tabelle 5.3: Kennzahlen der simulierten Streckennetze .....	95
Tabelle 5.4: Ankunftsraten mit relevanten Berechnungsparametern, UCL 2015. Als Kontrollpunkte ausgewählte Haltepunkte sind unterstrichen .....	96
Tabelle 5.5: Simulierte Fahrpläne für UCL 2015.....	96
Tabelle 5.6: Durchschnittliche Frühzeitigkeit und Verspätung pro Linie, UCL 2015 .....	98
Tabelle 5.7: Durchschnittliche Frühzeitigkeit und Verspätung an ausgewählten Haltepunkten, UCL 2015.....	107
Tabelle 5.8: Simulierte Fahrpläne für KVB-T 2001.....	112
Tabelle 5.9: Durchschnittliche Verspätung in Sekunden, pro Linie, KVB-T 2001 .....	113
Tabelle 5.10: Durchschnittliche Simulationslaufzeit in Sekunden, aufgeteilt in einzelne Phasen .....	122
Tabelle 5.11: Durchschnittliche Zahl verarbeiteter Ereignisse .....	123
Tabelle 7.1: Ereignistypen zur Abbildung von Störungen .....	133
Tabelle 7.2: Durchschnittlicher, kumulierter Anteil reduzierter Sicherheitsabstände unter unterschiedlichen Haltestrategien, UCL 2015. Erste Spalte: Start-HP – Nur die Starthaltepunkte fungieren als Kontrollpunkte; Alle – Alle Haltepunkte fungieren als Kontrollpunkte. Zweite Spalte: Lokal – Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. – Die erwartete Pünktlichkeit am nächsten Haltepunkt wird mit zur Entscheidungsfindung herangezogen. Dritte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an .....	149
Tabelle 7.3: Verspätung, Frühzeitigkeit und Wartezeit unter unterschiedlichen Haltestrategien, UCL 2015. Erste Spalte: Start-HP – Nur die Starthaltepunkte fungieren als Kontrollpunkte; Alle - Alle Haltepunkte fungieren als Kontrollpunkte. Zweite Spalte: Lokal – Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. – Die erwartete Pünktlichkeit am nächsten Haltepunkt wird zur Entscheidungsfindung herangezogen. Dritte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an .....	150
Tabelle 7.4: Durchschnittliche Linienverspätung bei Auftritt einer Störung unter unterschiedlichen Strategien, UCL 2015. A: Nur Haltestrategie, ausgewählte Kontrollpunkte, lokale Information, Grenzwert von null Sekunden. B: Modifikation des Linienverlaufs bei Detektion einer erwarteten Verspätung von mindestens 60 Sekunden. C: Kombinierte Anwendung von A und B.....	151
Tabelle 7.5: Durchschnittliche Frühzeitigkeit der Linien bei Auftritt einer Störung unter unterschiedlichen Strategien, UCL 2015. A: Nur Haltestrategie, ausgewählte Kontrollpunkte, lokale Information, Grenzwert von null Sekunden. B: Modifikation	

	des Linienverlaufs bei Detektion einer erwarteten Verspätung von mindestens 60 Sekunden. C: Kombinierte Anwendung von A und B .....	151
Tabelle 7.6:	Relevante Kennzahlen von Haltepunkt 1131 bei Auftritt einer Störung unter unterschiedlichen Strategien, UCL 2015. A: Nur Haltestrategie, ausgewählte Kontrollpunkte, lokale Information, Grenzwert von null Sekunden. B: Modifikation des Linienverlaufs bei Detektion einer erwarteten Verspätung von mindestens 60 Sekunden. C: Kombinierte Anwendung von A und B .....	152
Tabelle 7.7:	Durchschnittlicher, kumulierter Anteil reduzierter Sicherheitsabstände unter unterschiedlichen Kombinationen von dynamischen Linienmodifikationen und Haltestrategien, UCL 2015. Erste Spalte: S – Schnellste Route; P – Pünktlichste Route. Zweite Spalte: Nein – Linienmodifikationen kommen nur bei erwarteter Verspätung zum Einsatz; Ja – Linienmodifikationen kommen auch bei erwarteter Frühzeitigkeit zum Einsatz. Dritte Spalte: Lokal - Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. - Die erwartete Pünktlichkeit am nächsten Haltepunkt wird zur Entscheidungsfindung herangezogen. Vierte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an .....	152
Tabelle 7.8:	Verspätung, Frühzeitigkeit und Wartezeit unter unterschiedlichen Kombinationen von dynamischen Linienmodifikationen und Haltestrategien, UCL 2015. Erste Spalte: S – Schnellste Route; P – Pünktlichste Route. Zweite Spalte: Nein – Linienmodifikationen kommen nur bei erwarteter Verspätung zum Einsatz; Ja – Linienmodifikationen kommen auch bei erwarteter Frühzeitigkeit zum Einsatz. Dritte Spalte: Lokal - Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. - Die erwartete Pünktlichkeit am nächsten Haltepunkt wird zur Entscheidungsfindung herangezogen. Vierte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an .....	153
Tabelle 7.9:	Durchschnittliche Verspätung pro Linie, mit und ohne Störfallmanagement (SFM), KVB-T 2001 .....	154
Tabelle 8.1:	Verkehrsplannerische Vorgaben zur Modellierung von gerichteten Umsteigeverbindungen an ausgewählten Haltestellen, Gesamt-ÖPNV-Netz Köln, 2001. Jede Zeile steht für bis zu drei Vorgaben, jeweils eine pro Priorität .....	158
Tabelle 8.2:	Zielfunktionswerte und benötigte Laufzeiten mit und ohne Berücksichtigung von Umsteigevorgaben, KVB-BT 2001 .....	160
Tabelle 8.3:	Abfahrtszeiten der durch Umsteigeverbindungen betroffenen Linien unter den zu vergleichenden Fahrplänen, gruppiert nach Haltestellen, KVB-BT 2001 .....	160
Tabelle 8.4:	Durchschnittlicher prozentualer Anteil erfüllter Umsteigeverbindungen an den unterschiedlichen Haltestellen in Abhängigkeit des Fahrplans und der Warte-/Umsteigezeit, KVB-BT 2001 .....	162
Tabelle 8.5:	Durchschnittliche Verspätung an den unterschiedlichen Haltestellen in Abhängigkeit des Fahrplans und der Warte-/Umsteigezeit, KVB-BT 2001 .....	164
Tabelle 8.6:	Durchschnittliche Wartezeit an den unterschiedlichen Haltestellen in Abhängigkeit des Fahrplans und des Warte-/Umsteigezeit, KVB-BT 2001 .....	164
Tabelle 8.7:	Durchschnittliche Linienverspätung der durch Umsteigeverbindungen betroffenen Linien, KVB-BT 2001 .....	165
Tabelle 8.8:	Durchschnittliche Frühzeitigkeit der Rückrichtungen der durch Umsteigeverbindungen betroffenen Linien, KVB-BT 2001 .....	166
Tabelle 8.9:	Durchschnittliche Linienverspätung der Rückrichtungen der durch Umsteigeverbindungen betroffenen Linien, KVB-BT 2001 .....	166
Tabelle A.1:	Relation NET_INDEX .....	170
Tabelle A.2:	Relation TIME_TABLE_INDEX .....	170
Tabelle A.3:	Relation BASIC_INTERVAL .....	171

Tabelle A.4: Relation TRIPS.....	171
Tabelle A.5: Relation LOCATIONS.....	171
Tabelle A.6: Relation TRANSIT_TYPE.....	171
Tabelle A.7: Relation STATIONS .....	171
Tabelle A.8: Relation STOPS .....	172
Tabelle A.9: Relation CONNECTIONS .....	172
Tabelle A.10: Relation SEGMENTS .....	173
Tabelle A.11: Relation SEGMENT_TO_CONNECTION .....	173
Tabelle A.12: Relation SWITCHES.....	173
Tabelle A.13: Relation SIGNALS.....	173
Tabelle A.14: Relation LINE_INDEX.....	174
Tabelle A.15: Relation LINE_ROUTES.....	174
Tabelle A.16: Relation OPPOSITE_LINES.....	174
Tabelle A.17: Relation TRUNCATED_LINES .....	174
Tabelle A.18: Relation SCENARIO_INDEX .....	174
Tabelle A.19: Relation SCENARIO_ARRIVAL_RATES .....	175
Tabelle A.20: Relation NODE_TYPE .....	175
Tabelle A.21: Relation DISRUPTION_TYPE.....	175
Tabelle A.22: Relation SCENARIO_DISRUPTIONS.....	175
Tabelle A.23: Relation SCENARIO_RENDEZVOUS_CONNECTIONS .....	176
Tabelle A.24: Relation SCENARIO_TRANSFER_CONNECTIONS .....	176

## Verzeichnis der Algorithmen

Algorithmus 5.1: Simple Heuristik zur Erzeugung eines Umlaufplans .....	85
Algorithmus 5.2: Quellcode der Methode void run () .....	93
Algorithmus 7.1: Pseudocode zur Bestimmung zusätzlicher Wartezeit.....	135
Algorithmus 7.2: Pseudocode der Entscheidungslogik bei Rendezvousverbindungen.....	136
Algorithmus 7.3: Pseudocode zur Modifikation des Linienvverlaufs aufgrund größerer Störungen.....	138
Algorithmus 7.4: Pseudocode zur Verkürzung von Fahrten aufgrund von Verspätungen.....	139
Algorithmus 7.5: Pseudocode für eine allgemeine Variante zur Verkürzung von Fahrten, die sicherstellt dass Passagiere einen Mindestanteil der restlichen Haltepunkte erreichen.....	140
Algorithmus 7.6: Pseudocode Ausschnitt der Ereignisverarbeitung bei einer blockierten Verbindung.....	142
Algorithmus 7.7: Pseudocode der Entscheidungslogik bei Fahrturnleitung aufgrund einer blockierten Verbindung.....	142
Algorithmus 7.8: Pseudocode der Entscheidungslogik bei Fahrtverkürzung aufgrund einer blockierten Verbindung.....	144
Algorithmus 7.9: Pseudocode zur Bestimmung aller einfachen, überspringenden Wege.....	145
Algorithmus 7.10: Pseudocode zur Entscheidungslogik beim Auftrennen von Fahrten aufgrund einer Störung .....	146

## Abkürzungen und Symbole

$a$	Passagierankunftsrate	S. 90
$a_s$	Passagierankunftsrate für Haltepunkt $s$	S. 90
$a_{max}^{Diff}$	Maximal zulässige Abfahrtszeitabweichung bei Anwendung von Verkehrsmanagementstrategien	S. 134
$\alpha$	Gewichtungsfaktor	S. 35, 40
$a_i^s$	Abfahrtszeit von Linie $l_i$ an Haltepunkt $s_i$ , falls der zugehörige Fahrplan im Kontext eindeutig ist	S. 28
$a(s_i, l_i, \lambda)$	Abfahrtszeit von Linie $l_i$ an Haltepunkt $s_i$ und Fahrplan $\lambda$	S. 28
$ast_{min}^{Diff}$	Mindestwert der Abfahrtszeitabweichung der überschritten werden muss, damit eine dynamische Verkürzung des Linienverlaufs in Frage kommt	S. 139
$ar_{max}^{Diff}$	Grenzwert der Abfahrtszeitabweichung ab dem eine dynamische Modifikation des Linienverlaufs in Frage kommt	S. 137
ATP	Haltestellenabkürzung für Altonaer Platz	S. 117
B	Menge aller Betriebsfahrten	S. 85
$B$	Pro Passagier zum Einsteigen benötigte Zeitdauer	S. 89
$B_f$	Pro Passagier zum Einsteigen benötigte Zeitdauer für Fahrzeug $f$	S. 90
b	Einzelne Betriebsfahrt	S. 85
BAB	Haltestellenabkürzung für Barbarossaplatz	S. 119
BFM	Haltestellenabkürzung für Bahnhof Mühlheim	S. 119
BHE	Haltestellenabkürzung für Buchheim, Herler Str.	S. 119
BNT	Haltestellenabkürzung für Bonntor	S. 116
$\beta$	Normalisierungsfaktor	S. 35, 38
$C$	Menge von Verbindungen	S. 27
$C^b$	Konstante Mindestzeit für den Einstiegsprozess	S. 89
$C_f^b$	Konstante Mindestzeit für den Einstiegsprozess von Fahrzeug $f$	S. 90
$\tilde{C}_t$	Konfiguration eines Zellulären Automaten zum Simulationszeitpunkt $t$	S. 54
$\tilde{C}_t(z)$	Zustand einer Zelle $z$ eines zellulären Automaten zum Simulationszeitpunkt $t$	S. 54
$c_i$	Einzelne Verbindung	S. 27
CATS	Computer Aided Traffic Scheduling	S. 4
CDR	Conflict Detection and Resolution Problem	S. 128
CDRFR	Conflict Detection and Resolution with Fixed Routes	S. 128
CHP	Haltestellenabkürzung für Christophstr./Mediapark	S. 120
CPM	Critical Path Method	S. 127
CWP	Haltestellenabkürzung für Chlodwigplatz	S. 116
DEVS	Discrete Event System Specification	S. 73
DHS	Haltestellenabkürzung für Dellbrück, Hauptstraße	S. 118
DLR	Deutsches Zentrum für Luft- und Raumfahrt	S. 61
$\bar{\delta}$	Obere Schranke für den Sicherheitsabstand eines Netzwerks	S. 29
$\delta^*$	Optimaler Sicherheitsabstand eines Netzwerks	S. 29
$\delta(\lambda)$	Sicherheitsabstand eines Netzwerks unter Fahrplan $\lambda$	S. 29
$\bar{\delta}_\Sigma$	Obere Schranke für die Summe der Sicherheitsabstände	S. 30
$\delta_\Sigma^*$	Optimale Summe der Sicherheitsabstände für ein Netzwerk	S. 30
$\delta_\Sigma(\lambda)$	Summe der Sicherheitsabstände unter Fahrplan $\lambda$	S. 29
$\delta(m, \lambda)$	Sicherheitsabstand eines Linienkonflikts $m$ unter Fahrplan $\lambda$	S. 30
$\bar{\delta}(s)$	Obere Schranke für den Sicherheitsabstand an Haltepunkt $s$	S. 29
$\delta(s, \lambda)$	Sicherheitsabstand an einem Haltepunkt	S. 29
$\bar{\delta}(s, l_i, l_j)$	Obere Schranke für den Sicherheitsabstand zwischen Linien $l_i$ und $l_j$ an Haltepunkt $s$	S. 35

$\delta(s, l_i, l_j, \lambda)$	Sicherheitsabstand zwischen Linien $l_i$ und $l_j$ an Haltepunkt $s$ unter Fahrplan $\lambda$	S. 28
$E$	Kantenmenge	S. 30
$E_i$	Ereignis	S. 56, 75
$e$	Einzelne Kante	S. 30
EAN	Ereignis-Aktivitäts-Netzwerk	S. 25
EBP	Haltestellenabkürzung für Ebertplatz	S. 117
EFS	Haltestellenabkürzung für Eifelstr.	S. 116
EFW	Haltestellenabkürzung für Eifelwall	S. 120
$\eta$	Durch die Nutzerin angegebener Prozentsatz auf den die im Fahrplan veröffentlichten Fahrzeiten reduziert werden, um die geschätzte Standardabweichung der Fahrzeit zu bestimmen	S. 88
$\epsilon_{c_i}^b$	Anteil der operativen Überlegungen am Haltezeitanteil der geplanten Fahrzeit von Verbindung $c_i$	S. 88
$\epsilon_{c_i}^d$	Anteil der operativen Überlegungen am Fahrzeitanteil der geplanten Fahrzeit von Verbindung $c_i$	S. 88
$F$	Menge aller Fahrzeuge	S. 85
$F(s_i, s_j, l)$	Fahrzeit zwischen den Haltepunkten $s_i$ und $s_j$ für Linie $l$	S. 31
$f, \underline{f}$	Einzelnes Fahrzeug	S. 85, 89
$\tilde{f}$	Transitionsfunktion eines Zellulären Automaten	S. 54
$\bar{f}(\omega)$	Obere Schranke für die Erfüllung der verkehrsplanerischen Vorgabe $\omega$	S. 35
$f(\omega, \lambda)$	Gewichtungsfunktion für verkehrsplanerische Vorgaben im Optimierungsmodell von Ullrich et al.	S. 34
$\phi'(\lambda)$	Regularität von Fahrplan $\lambda$	S. 34, 36
$\phi''(\lambda)$	Erfüllung der Menge der verkehrsplanerischen Vorgaben $\Omega$ unter Fahrplan $\lambda$ im Optimierungsmodell von Ullrich et al.	S. 34, 38
FEL	Future-Event-List	S. 55
$Fr_{max}^{diff}$	Maximal zulässige Fahrzeitabweichung beim Umleiten von Fahrzeugen	S. 142/143
$G$	Ein Graph	S. 28
GDEVS	Generalized Discrete Event System Specification	S. 73
GTG	Haltestellenabkürzung für Goltsteinstr./Gürtel	S. 116
$\gamma$	Durch die Nutzerin angegebener Prozentsatz auf den die im Fahrplan veröffentlichten Fahrzeiten reduziert werden um die geschätzte durchschnittliche Fahrzeit zu bestimmen	S. 88
HEU	Haltestellenabkürzung für Heumarkt	S. 154
$hsa_{min}$	Mindesthaltepunktanteil der bei einer Verkürzung des Linienverlaufs durch betroffene Passagiere noch erreicht werden muss	S. 140
HSR	Haltestellenabkürzung für Hansaring	S. 116
INRETS	Französisches Institut für Verkehrs- und Sicherheitsforschung (franz. Institut national de recherche sur les transports et leur sécurité)	S. 62
ITS	Intelligentes Transport System (engl. Intelligent Transportation System)	S. 61
$\iota$	Durch die Nutzerin anzugebender Prozentsatz, um den die abgeschätzte Standardabweichung pro Segment erhöht wird	S. 89
KLB	Haltestellenabkürzung für Klettenbergpark	S. 120
KVB	Kölner Verkehrs-Betriebe AG	S. 95
$L$	Menge von Linien	S. 28
$L(s)$	Menge der Linien die Haltepunkt $s$ bedienen	S. 29
$L(f)$	Linie von Fahrzeug $f$	S. 90
LGR	Haltestellenabkürzung für Longerich, Friedhof	S. 116
LGS	Haltestellenabkürzung für Longericher Str.	S. 117
$LKG(N)$	Linienkonflikt-Graph zu Netzwerk $N$	S. 30

$l_i$	Einzelne Linie	S. 28
$\tilde{L}$	Gitter eines Zellulären Automaten	S. 53
$\Lambda$	Menge aller Fahrpläne	S. 28
$\lambda$	Fahrplan (Vektor von Startzeiten)	S. 28
$\lambda^*$	Optimalfahrplan	S. 29
$\lambda_i$	Abfahrtszeit innerhalb des Taktes von Linie $l_i$ an ihrem Starthaltepunkt	S. 28
$M$	Kantenmenge eines Multi-Linienkonflikt-Graphen	S. 30
$M_0$	Repräsentantenmenge der Kantenmenge $M$ eines Multi-Linienkonflikt-Graphen	S. 31
MBG	Haltestellenabkürzung für Marienburg, Südpark	S. 116
Megaffic	IBM Mega Traffic Simulator	S. 74
MHS	Haltestellenabkürzung für Maria-Himmelfahrt Str.	S. 118
MILP	Gemisch-ganzzahliges lineares Programm (engl. mixed integer linear program)	S. 25
$MLKG(N)$	Multi-Linienkonflikt-Graph zum Netzwerk $N$	S. 30
$\mu_{c_i}^b$	Beobachtete durchschnittliche Haltezeit am Zielhaltepunkt von Verbindung $c_i$	S. 88
$\mu_{c_i}^d$	Beobachtete durchschnittliche Fahrzeit auf Verbindung $c_i$	S. 88
$\hat{\mu}_{c_i}^d$	Geschätzte durchschnittliche Fahrzeit auf Verbindung $c_i$	S. 88
$N$	Nachbarschaftsrelation eines Zellulären Automaten	S. 53
$N(z)$	Nachbarschaftsrelation einer Zelle $z$ eines zellulären Automaten	S. 53
$N_{f,s}$	Zahl der an Haltepunkt $s$ in Fahrzeug $f$ einsteigenden Passagiere	S. 89
$N(S, C, t, L, T)$	Ein ein Streckennetz repräsentierendes Netzwerk	S. 27
$\mathbb{N}_{T_i}$	Menge der möglichen Startzeiten $\lambda_i$ von Linie $l_i$	S. 28
NEU	Haltestellenabkürzung für Neumarkt	S. 154
NFS	Haltestellenabkürzung für Neufelder Str.	S. 118
NIS	Haltestellenabkürzung für Niehl, Sebastianstraße	S. 119
$\Omega$	Menge verkehrsplanerischer Vorgaben	S. 34
$\Omega_C$	Menge aller verkehrsplanerischen Umsteigevorgabe	S. 37
$\Omega_D$	Menge aller verkehrsplanerischer Abfahrtszeitvorgaben	S. 36
$\Omega_H$	Menge aller verkehrsplanerischen Abstandsvorgaben	S. 37
$\omega_i$	Einzelne verkehrsplanerische Vorgabe	S. 34, 36
ÖPNV	Öffentlicher Personennahverkehr	S. 5
$P$	Menge möglicher Prioritätswerte	S. 34, 36
$R_M$	Äquivalenzrelation auf der Kantenmenge $M$ eines Multi-Linienkonflikt-Graphen	S. 31
$R_\Sigma$	Äquivalenzrelation auf der Menge der Haltepunkte $S$	S. 31
$ra_{min}$	Mindestroutenanteil der bei einer Verkürzung des Linienverlaufs bedient werden muss	S. 139
RDP	Haltestellenabkürzung für Rudolfplatz	S. 116
ROMA	<u>R</u> ailway traffic <u>O</u> ptimization by <u>M</u> eans of <u>A</u> lternative Graphs	S. 128
RPP	Haltestellenabkürzung für Reichenpergerplatz	S. 120
$R_{skip}$	Menge einfacher gerichteter Wege, die eine bestimmte Menge von Haltepunkten überspringen	S. 145
$rs_{max}$	Maximal zulässige relative Verlängerung einer Route beim Umleiten von Fahrzeugen	S. 144
$\rho_i$	Durch die Nutzerin zugewiesener Prioritätswert	S. 34, 36
$S$	Menge von Haltepunkten	S. 27
$S_\Sigma$	Repräsentantenmenge der Faktormenge $S/R_\Sigma$	S. 32
$S_{\Sigma, >1}$	Repräsentantenmenge der Faktormenge $S_{>1}/R_\Sigma$	S. 32
$S_{>1}$	Menge der Haltepunkte die von mehr als einer Linie bedient werden	S. 32
$S_{=1}$	Menge der Haltepunkte die von genau einer Linie bedient werden	S. 32

$S(l_i)$	Menge der durch Linie $l_i$ bedienten Haltepunkten	S. 28
$\tilde{S}$	Menge von Zuständen eines Zellulären Automaten	S. 53
$SG(c_i)$	Menge der Segmente von Verbindung $c_i$	S. 88/89
$s_i$	Einzelner Haltepunkt	S. 27
SUMO	<u>S</u> imulation of <u>U</u> rban <u>M</u> obility	S. 61
$\sigma_{c_i}^b$	Beobachtete Standardabweichung der Haltezeit am Zielhaltepunkt von Verbindung $c_i$	S. 88
$\sigma_{c_i}^d$	Beobachtete Standardabweichung der Fahrzeit auf Verbindung $c_i$	S. 88
$\hat{\sigma}_{c_i}^d$	Geschätzte Standardabweichung der Fahrzeit auf Verbindung $c_i$	S. 88
$T$	Menge aller Takte	S. 28
$T_i$	Takt von Linie $l_i$	S. 28
$T_{f,s}$	Einstiegsdauer für Fahrzeug $f$ an Haltepunkt $s$	S. 89
$t, \tau$	(Simulations-)Zeitpunkt	S. 54, 135
$\tilde{t}$	Durch die Nutzerin als Teil einer verkehrsplanerischen Vorgabe angegebene untere Schranke	S. 36
$\bar{t}$	Durch die Nutzerin als Teil einer verkehrsplanerischen Vorgabe angegebene obere Schranke	S. 36
$t_d$	Beim Störfallmanagement eingesetzte Pufferzeit	S. 142/143
$t_q$	Abfrageintervall für Rendezvous- und Umsteigeverbindungen	S. 136
$t_t$	Minimal notwendige Umsteigezeit	S. 136
$t_w$	Mindestwendezeit	S. 85
$t(c_i)$	(Im Fahrplan veröffentlichte) Fahrzeit auf Verbindung $c_i$	S. 27
$t^b(c_i)$	Geplanter Haltezeitanteil in der im Fahrplan veröffentlichten Fahrzeit auf Verbindung $c_i$	S. 88
$t^d(c_i)$	Geplanter Fahrzeitanteil in der im Fahrplan veröffentlichten Fahrzeit auf Verbindung $c_i$	S. 88
$t(s_i, s_j)$	Fahrzeit zwischen den Haltepunkten $s_i$ und $s_j$	S. 28
$t^{dep}(f - 1, s)$	Abfahrtszeitpunkt des Vorgängerfahrzeugs von Fahrzeug $f$ an Haltepunkt $s$	S. 90
TAC	Haltestellenabkürzung für Tacitusstr.	S. 116
THB	Haltestellenabkürzung für Thielenbruch	S. 118
TRANSIMS	<u>T</u> ransportation <u>A</u> nalysis and <u>S</u> imulation <u>S</u> ystem	S. 73
Typ (b)	Menge der für Betriebsfahrt $b$ zulässigen Fahrzeugtypen	S. 85
Typ (f)	Fahrzeugtyp von Fahrzeug $f$	S. 85
UBI	Haltestellenabkürzung für Ubierring	S. 118
ULR	Haltestellenabkürzung für Ulrepforte	S. 119
VRS	Haltestellenabkürzung für Holweide, Vischeringstr.	S. 118
$wl_{max}$	Maximale Tiefe/Weglänge für Breitensuche	S. 138
$w_t$	Einem Fahrzeug durch den Dispatcher zugewiesene Wartezeit	S. 135
$w_{t_c}$	Maximale Wartezeit bei Umsteigeverbindungen	S. 135/136
$x_{ij}^s$	Abstand zwischen den Abfahrtszeiten der Linien $l_i$ und $l_j$ an Haltepunkt $s$	S. 29
$x_{ij}^{sr}$	Abstand zwischen der Abfahrtzeit von Linie $l_i$ an Haltepunkt $s$ und der Abfahrtszeit von Linie $l_j$ an Haltepunkt $r$	S. 37
$Z$	Zellulärer Automat	S. 53
$z$	Eine einzelne Zelle eines zellulären Automaten	S. 53
$z_{ij}^s$	Ganzzahlige Variable, verwendet zur Berechnung von $x_{ij}^s$	S. 29
$z_{ij}^{sr}$	Ganzzahlige Variable, verwendet zur Berechnung von $x_{ij}^{sr}$	S. 37

# 1. Einführung

## 1.1. Kontext und Motivation

Die Planung und der Betrieb von für Passagiere hochwertigen Öffentlichen Personennahverkehrssystemen unter für den Betreiber und die Öffentliche Hand zumutbaren (monetären) Bedingungen stellt eine hoch komplexe Aufgabe dar. Diese Komplexität macht den *Öffentlichen Personennahverkehr (ÖPNV)* zu einem interessanten und lebhaften Forschungsfeld für unterschiedlichste wissenschaftliche Disziplinen. Eine dieser Disziplinen ist das Operations Research, dessen Techniken und Methoden der Modellbildung, Optimierung und Simulation bereits seit Jahrzehnten erfolgreich zur besseren Entscheidungsfindung im ÖPNV eingesetzt werden (für eine Übersicht siehe z.B. [30]). Als Gesamtproblem betrachtet stellen Planung und Betrieb von ÖPNV-Systemen allerdings ein, auch mit den heutigen technischen Möglichkeiten, nicht zu beherrschendes (Optimierungs-)Problem dar. Daher werden in der Regel Teilaspekte dieses Problems in sequentieller Abfolge durch eigenständige Optimierungsprobleme gelöst. Diese Optimierungsprobleme können grob in vier, nicht notwendigerweise überschneidungsfreie, Phasen eingeteilt werden: strategische Planung, taktische Planung, operative Planung und operativer Betrieb.

Während der *strategischen Planung* werden langfristig orientierte Aufgaben, wie die Netz- und Linienplanung, betrachtet. Nach einer anfänglichen Bedarfsanalyse müssen das Netzwerk und die zugehörigen Linienverläufe so geplant werden, dass der ermittelte Bedarf unter minimalem Einsatz von Ressourcen optimal abgedeckt wird. Zusätzlich müssen aufgrund politischer oder betrieblicher Vorgaben auftretende Bedingungen berücksichtigt werden. Beispiele für solche Bedingungen sind die Sicherstellung von Anschlüssen zu vorhandener (Verkehrs-)Infrastruktur und die Einhaltung von vorgegebenen Servicefrequenzen (vgl. für diesen und die folgenden drei Absätze [30]).

In der *taktischen Planungsphase* werden vornehmlich saisonal auftretende Aufgaben adressiert. Dazu gehören unter anderem die Bestimmung der Linientakte sowie die Fahrplanerstellung. Ziel ist die Verbesserung der Servicequalität unter Berücksichtigung betrieblicher und verkehrsplanerischer Vorgaben, wie beispielsweise die Einhaltung der vorgegebenen Flottengröße oder die Sicherstellung von Umsteigerverbindungen.

Aufbauend auf den Ergebnissen der strategischen und taktischen Planung werden während der *operativen Planung* Probleme mit eher kurzfristigem Zeithorizont (von einem Monat bis zu einem Tag) betrachtet. Unter anderem muss eine kostenminimale Personal- und Fahrzeugplanung durchgeführt werden, die nicht nur die Durchführbarkeit der Ergebnisse der vorgelagerten Phasen sicherstellt, sondern weitere politische, betriebliche und tarifliche Bedingungen berücksichtigt. Beispiele für solche Bedingungen sind die Berücksichtigung tariflich bedingter Pausenzeiten oder die Integration regelmäßiger Wartungen.

Die tatsächliche im *operativen Betrieb* abrufbare Servicequalität von Verkehrssystemen ist von einer Reihe äußerer und interner Faktoren, wie Wetterbedingungen oder (Verkehrs-)Unfällen, abhängig. Um die Auswirkungen dieser Störungen auf den Betrieb möglichst gering zu halten, können Verkehrsbetreiber, abhängig von der Schwere der Störung, eine Reihe von Techniken des Verkehrsmanagements anwenden. Dazu gehören unter anderem das dynamische Anpassen von Haltezeiten oder das Überspringen von Haltepunkten.

Im Rahmen des am Lehrstuhl von Prof. Dr. Ewald Speckenmeyer am Institut für Informatik der Universität zu Köln beheimateten Projekts *Computer Aided Traffic Scheduling (CATS)* wurden bisher Methoden und Werkzeuge zur Generierung, Simulation und Bewertung von Stadtbahnfahrplänen entworfen und entwickelt, um Verkehrsplaner während der taktischen Planungsphase zu unterstützen. Die entwickelte Software soll ohne Zugang zu Großrechnern oder speziellen Maschinen auskommen und von Anwenderinnen vor Ort auf handelsüblichen Desktop-PCs oder Notebooks genutzt werden können.

Wissenschaftlich liegt das Interesse in der Erforschung und Entwicklung der zur Erfüllung dieser Aufgaben notwendigen Konzepte und Methoden. Diese liegen hauptsächlich in den Bereichen der mathematischen Optimierung und diskreten Simulation. Neben 14 Bachelor-, Master- und Diplomarbeiten sind im Rahmen des Projekts bisher drei Dissertation (siehe [39], [58] und [89]) entstanden. Diverse Forschungsergebnisse wurden bereits in einer Reihe von Artikeln und Vorträgen veröffentlicht (siehe [59], [61–63], [81], [90–95]).

Das CATS-Projekt soll im Rahmen der vorliegenden Arbeit an verschiedenen Stellen erweitert werden. Das Hauptaugenmerk liegt dabei im Bereich der Simulation. Es sollen neue Modelle und Methoden zur Erzeugung und Evaluation von Fahrplänen für multi-modale Öffentliche Personennahverkehrssysteme entworfen und entwickelt werden. Darüber hinaus sollen Verkehrsplaner nicht nur bei der taktischen Planung, sondern auch während des operativen Betriebs unterstützt werden, indem geeignete Modelle und Methoden zur (automatischen) Umsetzung und Evaluation unterschiedlicher Maßnahmen des Verkehrsmanagements entworfen und entwickelt werden.

Die resultierenden Anwendungen sollen so modular gestaltet sein, dass sie bei Bedarf um Modelle weiterer Verkehrsmodi und neuer Verkehrsmanagementmaßnahmen erweitert werden können. Zugleich müssen sie (initial) mit einer sehr spärlichen Datenbasis auskommen, aber dennoch plausibles Verhalten aufweisen.

## 1.2. Ziele der Arbeit

Die im vorhergehenden Abschnitt vermittelten Ziele lassen sich grob in drei Bereiche einteilen:

1. Zunächst soll ein Optimierungsmodul zur Generierung von Fahrplänen mit hoher Serviceregularität für multi-modale Verkehrssysteme entworfen und entwickelt werden, das zusätzlich die Möglichkeit bietet Interdependenzen innerhalb und zwischen Verkehrsmodi, wie bspw. Umsteigeverbindungen, zu berücksichtigen. Um eine Anwendbarkeit der resultierenden Fahrpläne im täglichen Betrieb zu gewährleisten, soll es darüber hinaus auch von Verkehrsplanern formulierte ökonomische, betriebliche und politische Bedingungen berücksichtigen können. Das Modul soll außerdem auch bei umfangreichen Verkehrsnetzen in akzeptabler Laufzeit Fahrpläne mit hoher Güte erzeugen.
2. Es soll ein neues Simulationsmodul für multi-modalen ÖPNV entworfen und entwickelt werden, das zur Evaluation der Güte eingesetzter Fahrpläne verwendet werden kann. Die einzelnen Bestandteile des Moduls sollen so modular gestaltet werden, dass zukünftige Erweiterungen um neue Verkehrsmodi auf einfache Art und Weise möglich sind. Zur Entwicklung des Moduls wird dabei – im Rahmen des CATS-Projekts – absichtlich ein Schritt zurückgegangen und nicht das von Ullrich in [85] vorgestellte parallele Simulationsmodul erweitert. So kann sich zunächst gänzlich auf die Korrektheit und Plausibilität der verwendeten Modelle und Verfahren konzentriert werden, ohne zeitgleich Modellierungsentscheidungen treffen zu müssen, die durch die Parallelisierung des Moduls entstehen würden.
3. Das entwickelte Simulationsmodul soll anschließend um die Möglichkeit erweitert werden, während des simulierten operativen Betriebs selbstständig Maßnahmen des Verkehrsmanagements anstoßen zu können und Anwenderinnen so in die Lage versetzen unterschiedliche Verkehrsmanagementstrategien evaluieren zu können. Auch diese Erweiterungen sollen so modular aufgebaut sein, dass zukünftige Erweiterungen mit vergleichsweise geringem Aufwand möglich sind.

Für alle Module und Erweiterungen gilt, dass sie auch mit einer sehr spärlichen Datenbasis noch anwendbar sein und plausibles Verhalten aufweisen sollen.

Jedes der resultierenden Module soll außerdem anhand von Anwendungen auf Modelle unterschiedlicher künstlicher und realer Verkehrsnetze, hinsichtlich seiner Funktionalität, Plausibilität und Leistungsfähigkeit untersucht werden.

### 1.3. Vorgehen

Nach Abschluss dieses Kapitels erfolgt in Kapitel 2 eine Einführung in das Gebiet der Fahrplanerstellung für den ÖPNV, sowie eine kurze Besprechung ausgewählter aus der Literatur bekannter Optimierungsmodelle.

In Kapitel 3 wird ein neues Optimierungsmodell zur Fahrplangenerierung für multi-modale ÖPNV-Systeme vorgestellt, das sich Elementen zweier in Kapitel 2 vorgestellter Modelle bedient. Die Korrektheit und Leistungsfähigkeit des Modells werden anhand verschiedener Beispielanwendungen nachgewiesen.

In Kapitel 4 werden daraufhin allgemeine Simulationsverfahren eingeführt und diverse speziell für Verkehrssysteme entwickelte Simulationsmodelle besprochen. Dabei wird sich vornehmlich auf diskrete Simulationsmodelle mit dynamischen und stochastischen Elementen konzentriert.

Im darauf folgenden Kapitel 5 wird ein neues Simulationsmodell für multi-modalen ÖPNV auf Basis des diskreten ereignisbasierten Simulationsansatzes vorgestellt und seine Implementierung als Java-Anwendung besprochen. Das Modell wird anschließend anhand einiger Beispielanwendung getestet und auf Plausibilität überprüft.

In Kapitel 6 werden die Grundlagen des Verkehrsmanagements im Nah- und Fernverkehr eingeführt, sowie ausgewählte Modelle aus der Literatur besprochen.

Kapitel 7 baut auf diesen Ausführungen auf und beschreibt Erweiterungen für das in Kapitel 5 vorgestellte Simulationsmodell, zur Einbeziehung von Verkehrsmanagement. Nach einer Besprechung der Implementierung werden ausgewählte Erweiterungen anhand einiger Beispielanwendungen getestet und auf Plausibilität überprüft.

In Kapitel 8 wird die Anwendbarkeit des in den Kapiteln 3, 5 und 7 vorgestellten Programmpakets abschließend demonstriert, indem ein typischer Anwendungsfall für ein Modell des Gesamt-ÖPNV-Netzes der Stadt Köln aus dem Jahr 2001 besprochen wird.

Die Arbeit schließt in Kapitel 9 mit einer Zusammenfassung des Erreichten, sowie einem Ausblick auf das mögliche weitere Vorgehen.

## 2. Erstellung von Fahrplänen für den Öffentlichen Personennahverkehr

### 2.1. Einführung und Abgrenzung

Ziel der Fahrplanerstellung ist es, die Ergebnisse der Netz- und Linienplanung mit den zur Abdeckung des Passagierbedarfs notwendigen Linientakten zu einem praktikablen Fahrplan, d.h. einer Menge von Fahrten mit zugehörigen Abfahrtszeiten, mit möglichst hoher Servicequalität zu kombinieren (siehe [30]).

Unter welchen Umständen ein Fahrplan als praktikabel gilt und was genau gute Servicequalität ist, ist dabei von den betrachteten Verkehrssystemen, den relevanten Passagier Zielgruppen und weiteren betrieblichen, politischen und gesellschaftlichen Bedingungen abhängig. Aus diesem Grund kann eine Reihe von zum Teil gegenläufigen Gütekriterien zur Fahrplanerstellung verwendet werden.

Ein häufig anzutreffendes Gütekriterium ist *Serviceregularität* (siehe z.B. [6], [39], [46], [71], [90], [93]). Sie ist ein Maß für die Gleichmäßigkeit der zeitlichen Abstände zwischen aufeinander folgenden Abfahrten, auch Sicherheitsabstand genannt. Serviceregularität kann sowohl zur statischen Bewertung potentieller Fahrpläne während des Planungsprozesses, als auch zur dynamischen Bewertung der Systemperformanz im täglichen Betrieb herangezogen werden. Wie van Ort und van Nes in [71] schreiben, kann ein Fahrplan mit hoher Regularität während des täglichen Betriebs für eine gleichmäßigere Verteilung der Passagiere auf die im Einsatz befindlichen Fahrzeuge und damit für eine effizientere Fahrzeugauslastung mit weniger starken Auslastungsspitzen sorgen. Dies ist nicht nur ein für Betreiber erstrebenswertes Ziel, sondern reduziert auch das von Passagieren unerwünschte Auftreten von überfüllten Fahrzeugen. Nach Ullrich et al. in [94] kann Serviceregularität in Verkehrssystemen in denen sich Fahrzeuge gemeinsame Ressourcen mit knapper Kapazität und wenigen Ausweichmöglichkeiten teilen, wie z.B. in Stadtbahnsystemen, außerdem dabei helfen Fahrpläne zu generieren die widerstandsfähig gegenüber kleinen, im täglichen Betrieb unausweichlich auftretenden Verspätungen sind. Bei einem angenommenen Zehn-Minuten-Takt lassen sich die Fahrzeuge zweier Linien auf einem gemeinsamen Streckenabschnitt z.B. mit jeweils fünf Minuten planmäßigem Abstand zum nachfolgenden Fahrzeug versehen. Hier kann eines der beteiligten Fahrzeuge um über vier Minuten verspätet sein, ohne dass die nachfolgende Fahrt verzögert wird. Bei einer extrem ungleichmäßigen Aufteilung in Abstände von neun Minuten und einer Minute kann das eine Fahrzeug zwar über acht Minuten Verspätung haben, ohne dass es zu weiteren Konsequenzen kommt, das andere Fahrzeug kann jedoch schon eine kleine Verspätung von einer Minute auf seinen Nachfolger übertragen.

Serviceregularität ist ein besonders geeignetes Gütekriterium zur Erstellung von Fahrplänen für Verkehrssysteme mit hoher Taktfrequenz, bei denen nur schwer zusätzliche Pufferzeiten zwischen Abfahrten eingeplant werden können und Umsteigeverbindungen innerhalb des Verkehrssystems eine untergeordnete Rolle spielen.

Ein der Serviceregularität entgegengesetztes Gütekriterium ist *Synchronisation* (siehe z.B. [20], [34], [46], [75], [103]). Sie ist ein Maß für die Zahl der (nahezu) gleichzeitig stattfindenden Ankünfte/Abfahrten an den Haltepunkten des Verkehrssystems und soll für Passagiere komfortable Umsteigeverbindungen zwischen Fahrzeugen verschiedener Linien sicherstellen. Neben der Zahl der Ankünfte/Abfahrten wird häufig auch die geplante und/oder gemessene Wartezeit der Passagiere zur Bewertung der Synchronisation verwendet. Fahrpläne mit hoher Synchronisation helfen besonders bei der Verbesserung der Angebotsqualität in Verkehrssystemen mit niedriger Taktfrequenz, in denen die Verfügbarkeit attraktiver Umsteigeverbindungen wichtiger ist als eine gleichmäßige Fahrzeugauslastung. Fahrpläne mit hoher Serviceregularität führen unter derartigen Bedingungen in der Regel zu einem höheren Modalwert der Wartezeit als auf Umsteigeverbindungen optimierte Fahrpläne. Sollen Fahrpläne für bi- oder multimodale Verkehrssysteme erstellt werden, so kann sich Synchronisation auch auf Umsteigeverbindungen zwischen verschiedenen Verkehrsmodi beziehen.

Ein universelleres Gütekriterium das in den unterschiedlichsten Forschungsbereichen Anwendung findet ist *Robustheit*. Sie misst die Fähigkeit eines Systems, dynamisch auftretenden Einflüssen ohne Änderung seiner initialen, stabilen Struktur standzuhalten. Da ein System in der Regel nicht gegen alle erdenklichen Einflüsse abgesichert werden kann, muss dabei stets genau angegeben werden gegenüber welchen Arten von Einflüssen ein System robust ist. Im Rahmen der Fahrplangenerierung finden sich entsprechend unterschiedliche Interpretationen von Robustheit. Wie bereits ausgeführt, kann ein Fahrplan mit hoher Serviceregularität in einem Verkehrssystem mit vielen Interdependenzen an zentralen Ressourcen zu einer höheren Robustheit gegenüber kleineren Störungen führen. Im Gegensatz dazu kann durch einen Fahrplan in dem Schlupfzeiten, d.h. feste Pufferzeiten zwischen geplanter Ankunfts- und Abfahrtszeit von Fahrzeugen, eingeplant werden, die Robustheit eines Systems gegenüber Störungen von Umsteigeverbindungen erhöht werden (siehe z.B. [77]). Häufig werden zur Generierung robuster Fahrpläne auch Methoden der robusten Optimierung angewendet (siehe z.B. [17], [56], [76], [77], [104]). Die in diesem Bereich betrachteten mathematischen Optimierungsprobleme zeichnen sich durch Nebenbedingungen aus die von stochastischen Unsicherheitsmengen abhängen. Unter den Lösungen des Optimierungsproblems wird dabei diejenige gesucht, die auch unter dem schlimmsten durch die Unsicherheitsmengen abgedeckten Fall sicherstellt, dass die betroffenen Nebenbedingungen eingehalten werden.

Zusätzlich zu den bei der Generierung verwendeten Gütekriterien können Fahrpläne auch anhand verschiedener Strukturmerkmale klassifiziert werden. Zu diesen Merkmalen gehört unter anderem ob es sich um einen periodischen oder aperiodischen Fahrplan handelt. Bei einem periodischen Fahrplan – auch Taktfahrplan genannt – wiederholen sich Abfahrten von Linien in regelmäßigen, durch den Linientakts vorgegebenen, periodischen Abständen, z.B. alle zehn Minuten. Bei aperiodischen Fahrplänen folgen die Abfahrten hingegen keinem vorgegebenen Takt. Diese Form des Fahrplans ist vornehmlich in Verkehrssystemen zu finden in denen sich konkurrierende Verkehrsbetriebe die gleiche Infrastruktur teilen, sodass die Etablierung eines einheitlichen Taktfahrplans erschwert wird (siehe [18]).

Ein weiteres Strukturmerkmal ist die Homogenität der verwendeten Linientakte. Hierbei kann unterschieden werden ob für alle Linien ein einheitlicher Takt verwendet wird oder ob unterschiedliche Takte zugelassen werden. Letztere finden in der Regel dann Anwendung, wenn Fahrpläne für Netze mit regional stark schwankender Nachfrage erstellt werden.

Der Fokus dieser Arbeit liegt auf der Erstellung von periodischen Gesamtfahrplänen für hochfrequente, multi-modale ÖPNV-Systeme. Dabei sollen einerseits die Einzelfahrpläne für die verschiedenen Verkehrssysteme robust gegenüber kleineren Störungen sein, also eine hohe Serviceregularität aufweisen. Andererseits soll der Gesamtfahrplan aber Umsteigeverbindungen – insbesondere zwischen den verschiedenen Verkehrssystemen – berücksichtigen, also eine gewisse Synchronisation aufweisen. Umfangreiche Störungen sollen hingegen nicht über den Fahrplan, sondern während des täglichen Betriebs über einfache regelbasierte Verfahren adressiert werden. Im Folgenden werden entsprechend nur Verfahren zur Erstellung von regulären und synchronen Fahrplänen besprochen.

## **2.2. Optimierungsmodelle zur Erstellung von Fahrplänen für den Öffentlichen Personennahverkehr**

Es existieren zahlreiche Optimierungsmodelle zur Erstellung von Fahrplänen für den ÖPNV. Sie können unter anderem anhand des Abstraktionsgrads des unterliegenden Modellierungsansatzes, der angewendeten Lösungsmethode, der verwendeten (Kombination von) Gütekriterien und der Struktur des erstellten Fahrplans unterschieden werden (für eine allgemeine Übersicht siehe [30]).

Dem Großteil der in der Literatur zu findenden Verfahren zur Fahrplangenerierung ist gemein, dass sie auf einer Repräsentation des zu untersuchenden Verkehrssystems als Graphen basieren. Sie unterscheiden sich lediglich im gewählten Abstraktionsgrad. Häufig werden Eins-zu-Eins-Abbildungen verwendet, bei denen die Haltepunkte des Streckennetzes als Knoten im Graphen repräsentiert werden, während die zwischen den Haltepunkten befahrenen Streckenabschnitte als gerichtete, gewichtete Kanten aufgefasst werden (siehe z.B. [6], [39], [90]). Sind nur spezifische Komponenten des Streckennetzes (bspw. wichtige Umsteigeknoten) Ziel der Optimierung, können zur Abbildung auch Graphen mit reduzierter Knoten- und Kantenmenge verwendet werden (siehe z.B. [20], [34], [75], [103]). Eine häufig

angewendete Repräsentierung mit stärkerer Abstrahierung vom zu untersuchenden System basiert auf *Ereignis-Aktivitäts-Netzwerken* (engl. *Event-Activity Network, EAN*) (siehe z.B. [40], [77]). Statt der physischen Abhängigkeiten zwischen den Systemkomponenten stehen hier die logischen Abhängigkeiten im Mittelpunkt. Die Knoten des Graphen repräsentieren *Ereignisse* (z.B. Ankünfte und Abfahrten an Haltepunkten), während die Kanten Abhängigkeiten – die *Aktivitäten* – zwischen einzelnen Ereignissen abbilden (z.B. Mindesthaltezeiten zwischen Ankunft und Abfahrt an einem Haltepunkt).

Basierend auf der gewählten Graphrepräsentation, den gewünschten Strukturmerkmalen des Fahrplans, sowie den zur Bewertung der zulässigen Fahrpläne anzuwendenden Gütekriterien wird dann ein mathematisches Optimierungsproblem – in der Regel ein (*gemischt-ganzzahliges*) *lineares Programm* (engl. *mixed integer linear program, MILP*) – aufgestellt und mittels standardisierter Solver oder speziell auf die jeweilige Problemstellung abgestimmter Algorithmen (bspw. auf Basis von Metaheuristiken) gelöst. Sollen mehrere Gütekriterien bei der Generierung eines Fahrplans verwendet werden, werden häufig auch Verfahren der Multikriteriellen Optimierung angewendet. Dies ist insbesondere dann der Fall, wenn mehrere gegenläufige Gütekriterien zum Einsatz kommen, die nur schwer in eine gemeinsame Zielfunktion zusammengefasst werden können.

Ein maßgebendes Optimierungsmodell für die Fahrplangenerierung (insbesondere für Bussysteme) wurde von Ceder et al. in [20] vorgestellt und später von Eranki in [34] erweitert. Es ermöglicht die Erstellung von aperiodischen Fahrplänen mit maximaler Synchronisation an dedizierten Umsteigeknoten unter gleichzeitiger Einhaltung von vorgegebenen Mindest- und Maximalabständen zwischen zu terminierenden Ankunftszeiten von Fahrzeugen der gleichen Linie. Auf Basis einer Graphrepräsentation des Streckennetzes mit reduzierter Knoten- und Kantenmenge entwickeln die Autoren ein MILP zur Maximierung der Zahl gleichzeitig stattfindender Ankünfte. Zur Lösung des Modells entwerfen Ceder et al. ein konstruktives heuristisches Verfahren, das immer den zum aktuellen Zeitpunkt vielversprechendsten Umsteigeknoten identifiziert und versucht für eine möglichst große Zahl der diesen Knoten bedienenden Linien synchrone Ankünfte festzulegen. Können keine weiteren (synchrone) Abfahrten mehr hinzugefügt werden, wird der nächste Umsteigeknoten identifiziert. Die Korrektheit und Anwendbarkeit ihrer Heuristik zeigen die Autoren anhand einiger künstlicher Anwendungsbeispiele sowie anhand eines realen Beispiels auf Basis eines Busnetzes in Israel.

Neuere Verfahren zur Erstellung von Fahrplänen mit maximaler Synchronisation werden beispielsweise von Saharidis et al. in [75] und Wu et al. in [103] vorgestellt. Saharidis et al. entwerfen ein MILP zur Generierung von aperiodischen Fahrplänen mit minimalen Passagierwartezeiten an dedizierten Umsteigeknoten. Neben einer Basisversion, in der nur Wartezeiten zum nächstmöglichen Verbindungsfahrzeug minimiert werden, stellen die Autoren verschiedene Erweiterungen ihres Modells vor, die es ermöglichen sowohl die Wartezeit zu mehr als einem Verbindungsfahrzeug zu minimieren als auch Phasen mit hoher Nachfrage zu priorisieren. Ihr Modell testen Saharidis et al., indem sie mit Hilfe des IBM ILOG CPLEX Solvers Fahrpläne für das Busnetz der griechischen Insel Kreta erstellen und die unter diesen Fahrplänen resultierenden Wartezeiten mit den unter dem tatsächlich angewendeten Fahrplan resultierenden vergleichen.

Wu et al. entwerfen ebenfalls ein ganzzahliges Programm zur Minimierung von Passagierwartezeiten an dedizierten Umsteigeknoten, erstellen im Gegensatz zu Ceder et al. und Saharidis et al. aber periodische Fahrpläne und verwenden stochastische Fahr- und Wartezeiten, die sich pro Fahrt, Linie und Umsteigeknoten unterscheiden. Außerdem unterscheiden Wu et al. zwischen Wartezeiten für neu hinzukommende Passagiere, Wartezeiten für umsteigende Passagiere und Wartezeiten für Passagiere auf der Durchreise. Um die Auswirkungen der stochastischen Einflussfaktoren auf den Betrieb aufzufangen fügen die Autoren Schlupfzeiten in den Fahrplan ein. So können Umsteigeverbindungen trotz unplanmäßiger Schwankungen in den Ankunftszeiten der Fahrzeuge gehalten und eine hohe Synchronisation erreicht werden. Um ihr Modell zu lösen entwickeln die Autoren einen Genetischen Algorithmus (siehe [32]), dessen probabilistische Suchstrategien sie durch eine Strategie mit lokaler Suche ersetzen, um ein schnelles und zuverlässiges Konvergieren zu gewährleisten. Die Güte und Anwendbarkeit ihres Verfahrens zeigen Wu et al., indem sie für Modelle zufällig erzeugter Netzwerkinstanzen sowohl Fahrpläne unter der Annahme deterministischer als auch stochastischer Fahr- und Wartezeiten erzeugen und unter realistischen, d.h. stochastischen, Bedingungen miteinander vergleichen. Die Ergebnisse zeigen, dass

die Berücksichtigung stochastischer Daten schon in der Planungsphase zu einem Fahrplan führt, der besser auf die realen Bedingungen im täglichen Betrieb abgestimmt ist. Die durch den stochastischen Optimierungsansatz von Wu et al. erzeugten Fahrpläne reduzieren die Wartezeiten der um- und zusteigenden Passagiere, erhöhen aber gleichzeitig, aufgrund der eingefügten Schlupfzeiten, die Wartezeiten der Passagiere, die den Umsteigeknoten lediglich passieren wollen.

Modelle zur Erstellung von regulären Fahrplänen werden unter anderem von Genç in [39], Bampas et al. in [6] und Ullrich et al. in [90] vorgestellt. Genç formuliert ein ganzzahliges lineares Programm zur Erstellung von periodischen Fahrplänen mit maximaler Regularität an allen Haltepunkten. Ziel ist die Erhöhung der Widerstandsfähigkeit des Systems gegenüber kleinen, im täglichen Betrieb unausweichlich auftretenden Verspätungen. Er zeigt außerdem, dass bereits die Bestimmung eines kollisionsfreien Fahrplans mit maximaler Regularität am am stärksten frequentierten Haltepunkt ein NP-schweres Problem ist, indem er das zugehörige Entscheidungsproblem auf das als NP-vollständig bekannte  $k$ -Färbungsproblem der Graphentheorie zurückführt. Zur Lösung seines Modells implementiert Genç einen Branch-and-Bound Solver (siehe [25]) dessen Güte er durch Vergleiche mit von CPLEX berechneten Lösungen bestimmt. Er kann anhand von Untersuchungen an einem Modell des Stadtbahnnetzes der Stadt Köln zeigen, dass sein Solver besonders zu Beginn (innerhalb der ersten fünf Minuten) bessere Lösungen als CPLEX liefert. Mit zunehmender Laufzeit reichen die durch CPLEX berechneten Fahrpläne allerdings näher an das tatsächliche Optimum heran.

Bampas et al. entwickeln exakte und heuristische Algorithmen zur Lösung des von Genç formulierten Optimierungsproblems in speziellen Netzwerkstrukturen, wie Ketten-, Spinnen-, Ring- und Baumnetzwerken. Dabei begrenzen sich die Autoren, im Gegensatz zu Genç, auf Fahrpläne mit einheitlichem Taktintervall für alle Linien. Bampas et al. führen den von ihnen betrachteten Spezialfall des Optimierungsproblems auf das Kantenfärbungsproblem der Graphentheorie zurück und stellen exakte Algorithmen für Ketten-, Stern- und Spinnennetzwerke vor. Darüber hinaus beweisen die Autoren, dass die Erstellung von maximal regulären Fahrplänen mit einheitlichem Taktintervall für Ring- und Baumnetzwerke NP-schwer ist und stellen heuristische Verfahren zur Approximation der Optimallösung vor.

Ullrich et al. befassen sich, wie Bampas et al., mit einem Spezialfall des von Genç formulierten Optimierungsproblems, in dem alle Linien einem gemeinsamen Takt folgen. Sie erweitern diesen Spezialfall um die Möglichkeit zur Berücksichtigung politischer, ökonomischer sowie betrieblicher Vorgaben wie z.B. die Limitierung erlaubter Abfahrtszeiten oder die Einhaltung vorgegebener Abstände zwischen Abfahrten verschiedener Linien. Als Resultat ermöglicht das Modell von Ullrich et al. nicht nur die Optimierung der Serviceregularität, sondern auch eine indirekte Berücksichtigung von Umsteigeverbindungen. Zur Lösung des Optimierungsproblems kombinieren die Autoren einen Genetischen Algorithmus und einen Branch-and-Bound Solver. Der Genetische Algorithmus liefert dabei eine initiale obere Schranke für den Branch-and-Bound Solver, sodass schneller große Teile des Suchbaums ausgeschlossen werden können. Im Verlauf des Optimierungsprozesses bestimmt der Branch-and-Bound Solver lokale untere Schranken, indem für bereits gelöste Teile des Optimierungsproblems exakte Zielfunktionswerte berechnet und für noch zu lösende Teile des Optimierungsproblems Abschätzungen für die entsprechenden Teile des Zielfunktionswerts verwendet werden. Die Anwendbarkeit ihres Verfahrens zeigen Ullrich et al. anhand diverser Simulationsstudien. Sowohl für das Stadtbahnnetz der Stadt Köln für die Jahre 2001 (siehe [89], Abschnitt 6.4.1 und [90]) und 2012 (siehe [89], Abschnitt 6.4.2) als auch für das Stadtbahnnetz der südfranzösischen Stadt Montpellier aus dem Jahr 2013 (siehe [89], Abschnitt 6.5 und [93]) zeigen die Autoren, dass der von ihnen entwickelte Optimierungsansatz Fahrpläne generiert, die helfen die Auswirkungen kleinerer Verspätungen zu reduzieren, während sie gleichzeitig verkehrsplanerische Vorgaben berücksichtigen.

Ein Modell, das sowohl Synchronisierung als auch Serviceregularität berücksichtigt, wird von Ibarra-Rojas und Rios-Solis in [46] vorgestellt. Die Autoren bauen auf der von Eranki in [34] vorgestellten Erweiterung des Modells von Ceder et al. auf, um Fahrpläne für Modelle des Busnetzes der mexikanischen Stadt Monterrey zu erstellen. Dazu erweitern sie Erankis Modell sowohl um die Möglichkeit gerichteter Synchronisierung zwischen Linien als auch um die Möglichkeit möglichst reguläre Abstände zwischen Ankunftszeiten der gleichen Linie zu berücksichtigen. Darüber hinaus beweisen die Autoren,

dass bereits das von Ceder und Eranki formulierte Optimierungsproblem NP-schwer ist, indem sie das zugehörige Entscheidungsproblem auf das NAE-3SAT-Problem zurückführen. Zur Lösung des Optimierungsproblems entwerfen die Autoren ein iteratives lokales Suchverfahren mit multiplen Startpunkten (engl. *multi-start iterated local search*). Die Güte des Verfahrens ermitteln sie durch Anwendung auf Testinstanzen unterschiedlicher Größe und anschließenden Vergleich der erhaltenen Fahrpläne mit durch CPLEX berechneten Lösungen. Ibarra-Rojas und Rios-Solis können zeigen, dass ihr Verfahren auch für große Instanzen (bis zu 200 Linien) sehr schnell gute, wenn auch nicht optimale, Fahrpläne erzeugt. Im Gegensatz dazu liefert CPLEX keine optimale Lösung in akzeptabler Zeit. Die Zielfunktionswerte der besten durch CPLEX berechneten Fahrpläne haben zudem bei größeren Instanzen einen größeren Abstand zur besten gefundenen oberen Schranke als die durch das lokale Suchverfahren berechneten Lösungen.

Durch die Berücksichtigung regelmäßiger Abstände zwischen Ankunftszeiten als auch einer möglichst starken Synchronisierung, eignet sich das Modell von Ibarra-Rojas und Rios-Solis sowohl für Verkehrssysteme mit niedriger als auch mit hoher Taktfrequenz. Im Gegensatz zu den Modellen von Genç, Bampas et al. und Ullrich et al. werden allerdings nur die Sicherheitsabstände zwischen Abfahrten gleicher Linien regelmäßig verteilt. Als Folge dessen kann es in Verkehrssystemen in denen sich mehrere Linien Ressourcen mit begrenzter Kapazität teilen dazu kommen, dass ein verspätetes Fahrzeug einen Teil seiner Verspätung auf ein die gleiche Ressource verwendendes Nachfolgefahrzeug einer anderen Linie überträgt. Im schlimmsten Fall kann sich diese Verspätung im Verlauf des Betriebstages auf weitere Fahrzeuge übertragen, die gesamte Systemleistung beeinträchtigen und umfangreiche Maßnahmen des Verkehrsmanagements notwendig machen.

Mit Ausnahme des Modells von Ullrich et al. berücksichtigt keines der vorgestellten Modelle verkehrsplanerische Vorgaben. Die Optimierung der Synchronisierung und Serviceregularität reicht unter Umständen aber nicht aus um einen im täglichen Betrieb einsatzfähigen Fahrplan zu erzeugen. Ökonomische, betriebliche und politische Gegebenheiten können die Berücksichtigung nicht unmittelbar ersichtlicher Anforderungen, wie z.B. die Koordination von Abfahrtszeiten an Haltepunkten mit den Unterrichtszeiten angrenzender Schulen, notwendig machen. Werden diese Anforderungen nicht bereits während der Optimierung berücksichtigt, können händische Modifikationen an den generierten Fahrplänen notwendig sein, wodurch die Optimalität leiden kann.

Um die im vorherigen Abschnitt bereits erwähnten regulären und synchronen periodischen Fahrpläne für multi-modale ÖPNV-Systeme zu erstellen, sollen im weiteren Verlauf dieser Arbeit Aspekte der Modelle von Genç und Ullrich et al. miteinander zu einem neuen Optimierungsmodell kombiniert werden. Sie werden deswegen im Folgenden detaillierter vorgestellt.

### 2.2.1. Ein Ansatz zur Maximierung zeitlicher Sicherheitsabstände<sup>1</sup>

Zur Erstellung von Fahrplänen mit maximaler Serviceregularität formuliert Genç zwei Optimierungsprobleme: das *Sicherheitsabstand-Problem* und das *Fahrplan-Problem*. Ersteres behandelt die Fragestellung einen Fahrplan aus der Menge aller zulässigen Fahrpläne zu bestimmen, der den netzwerkweiten minimalen Sicherheitsabstand maximiert. Dadurch wird die optimale Regularität an stark befahrenen Haltepunkten – den (vermuteten) Flaschenhälsen des Systems – erreicht, weniger stark befahrene Haltepunkte, beispielsweise in städtischen Randbezirken, werden allerdings systematisch vernachlässigt. Als Folge wird im Allgemeinen nicht die für das Gesamtsystem optimale Regularität erreicht. Das Fahrplan-Problem adressiert diese Problematik indem aus der Menge aller Fahrpläne, die das Sicherheitsabstand-Problem optimal lösen, derjenige bestimmt wird, der auch an den restlichen Stationen des Netzwerks die bestmögliche Regularität erreicht.

Zur Lösung des Fahrplan-Problems stellt Genç ein ganzzahliges lineares Programm auf. Ausgangspunkt dieses Modells ist eine Darstellung des Streckennetzes als Netzwerk  $N(S, C, t, L, T)$  mit Haltepunkten  $S = \{s_1, \dots, s_n\}$ , Verbindungen  $C = \{c_1, \dots, c_v\} \subseteq S \times S$  mit Fahrzeiten  $t: C \rightarrow \mathbb{N}_0$ , und Linien

---

<sup>1</sup> Zur besseren Vergleichbarkeit und Verständlichkeit des Modells wurden einige Variablenbezeichner angepasst und weichen von den Originalbezeichnern aus [39] ab.

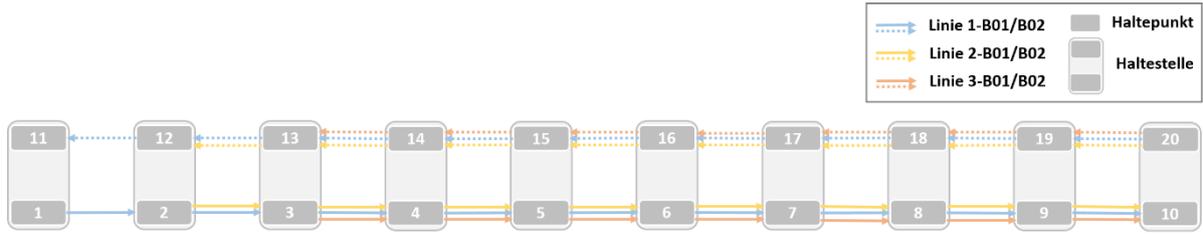


Abbildung 2.1: Netzwerk DCL 2014

$L = \{l_1, \dots, l_m\}$ . Eine Linie  $l_i = (s_{i1}, s_{i2}, \dots, s_{ik(i)})$  ist ein einfacher gerichteter Weg im gerichteten Graphen  $G = (S, C)$  mit Takt  $T_i$  aus der Menge aller Takte  $T = \{T_1, \dots, T_m\}$ . Abbildung 2.1 zeigt ein simples Beispielsnetzwerk mit sechs Linien und 20 Haltepunkten.

Für jede Linie  $l_i$  bezeichnet  $\lambda_i$  die Abfahrtszeit am Starthaltepunkt  $s_{i1}$ . Ein Vektor mit Abfahrtszeiten für alle  $m$  Linien eines Netzwerks bildet einen Fahrplan  $\lambda$  aus der Menge aller Fahrpläne  $\Lambda = \mathbb{N}_{T_1} \times \mathbb{N}_{T_2} \times \dots \times \mathbb{N}_{T_m}$ . Da ausschließlich periodische Taktfahrpläne betrachtet werden, gilt  $\mathbb{N}_{T_i} = \{0, \dots, T_i - 1\}$  und es folgt:

$$0 \leq \lambda_i \leq T_i - 1 \quad (2.1)$$

Um die Handhabbarkeit des Modells zu gewährleisten nimmt Genç an, dass die Fahrzeiten zwischen den Haltepunkten fest gegeben und unabhängig vom Fahrplan sind. Dadurch lassen sich für Fahrplan  $\lambda$  die Abfahrtszeiten  $a(s_{ip}, l_i, \lambda)$  an den von Linie  $l_i$  befahrenen Haltepunkten  $s_{ip} \in S(l_i)$  durch die Summierung von  $\lambda_i$  mit allen Fahrzeiten der bis zum Erreichen von  $s_{ip}$  befahrenen Verbindungen vollständig bestimmen (siehe Gleichung (2.2)).

$$a(s_{ip}, l_i, \lambda) := \lambda_i + \sum_{k=1}^{p-1} t(s_{ik}, s_{ik+1}) \quad 1 \leq p \leq s_{ik(i)} \quad (2.2)$$

Die Abfahrtszeiten an den Haltepunkten des Netzwerks sind folglich nur von den  $m$  Abfahrtszeiten an den Starthaltepunkten  $\lambda_1, \dots, \lambda_m$  abhängig und es lässt sich vereinfacht schreiben:

$$a(s, l_i, \lambda) = \lambda_i + a(s, l_i, \vec{0}) \quad (2.3)$$

Ist eindeutig zu welchem Fahrplan eine Abfahrtszeit gehört, wird verkürzt  $a_i^s$  geschrieben.

Bevor das Sicherheitsabstand- und Fahrplan-Problem formalisiert werden können, müssen zunächst verschiedene Sicherheitsabstandsmaße eingeführt werden.

### Definition 2.1 (Sicherheitsabstand zwischen zwei Linien, siehe [39])

Der Sicherheitsabstand  $\delta(s, l_i, l_j, \lambda)$  am Haltepunkt  $s \in S$  zwischen den Linien  $l_i, l_j \in L$  unter Fahrplan  $\lambda$  ist die minimale Differenz aller Abfahrtszeiten von  $l_i$  und  $l_j$  an Haltepunkt  $s$ :

$$\delta(s, l_i, l_j, \lambda) := \min_{r, u \in \mathbb{N}} |a(s, l_i, \lambda) + r * T_i - a(s, l_j, \lambda) - u * T_j| \quad (2.4)$$

Im Optimalfall wird das durch  $ggT(T_i, T_j)$  definierte Intervall dabei äquidistant zwischen den Abfahrten der betroffenen Linien aufgeteilt. Der Sicherheitsabstand zwischen  $l_i$  und  $l_j$  kann also wie folgt nach oben begrenzt werden (für einen ausführlichen Beweis siehe [39], S.53-54):

$$\delta(s, l_i, l_j, \lambda) \leq \left\lfloor \frac{ggT(T_i, T_j)}{2} \right\rfloor \quad (2.5)$$

Eine von Genç zur Formulierung des Fahrplan-Problems als ganzzahliges lineares Programm verwendete vereinfachte Methode zur Berechnung des Sicherheitsabstands zeigt Vince in [97]:

$$\delta(s, l_i, l_j, \lambda) := \min\{x_{ij}^s, ggT(T_i, T_j) - x_{ij}^s\} \quad (2.6)$$

Dabei sei  $x_{ij}^s = a_i^s - a_j^s - z_{ij}^s * ggT(T_i, T_j)$  und  $z_{ij}^s \in \mathbb{Z}$  so gewählt, dass  $x_{ij}^s \in \mathbb{N}_0$  gilt.<sup>2</sup>

**Definition 2.2 (Sicherheitsabstand an einem Haltepunkt, siehe [39])**

Der Sicherheitsabstand  $\delta(s, \lambda)$  an Haltepunkt  $s$  unter Fahrplan  $\lambda$  ist das Minimum der paarweisen Sicherheitsabstände der Linien aus  $L(s)$ , der Menge aller Linien die  $s$  bedienen:

$$\delta(s, \lambda) := \begin{cases} \min_{l_i, l_j \in L(s), l_i \neq l_j} \delta(s, l_i, l_j, \lambda), & |L(s)| > 1 \\ T_{L(s)}, & |L(s)| = 1 \end{cases} \quad (2.7)$$

Auf Basis der in Formel (2.5) definierten oberen Schranke kann dann für den Sicherheitsabstand an einem Haltepunkt  $s$  eine obere Schranke  $\bar{\delta}(s)$  wie folgt definiert werden (für einen ausführlichen Beweis und weitere Schranken siehe [39], S. 54-59):

$$\bar{\delta}(s) = \left\lfloor \min_{l_i, l_j \in L(s), l_i \neq l_j} \frac{ggT(T_i, T_j)}{2} \right\rfloor \quad |L(s)| \geq 2 \quad (2.8)$$

**Definition 2.3 (Sicherheitsabstand eines Netzwerks, siehe [39])**

Der Sicherheitsabstand  $\delta(\lambda)$  eines Streckennetzwerks unter einem Fahrplan  $\lambda$  ist das Minimum aller Sicherheitsabstände an den Haltepunkten des Netzwerks:

$$\delta(\lambda) := \min_{s \in S} \delta(s, \lambda) \quad (2.9)$$

Es ist unmittelbar ersichtlich, dass eine obere Schranke des Sicherheitsabstands in einem Streckennetzwerk  $\bar{\delta}$  wie folgt bestimmt werden kann:

$$\bar{\delta} = \min_{s \in S} \bar{\delta}(s)$$

Mit den Sicherheitsabständen für Linienpaare, Haltepunkte und Streckennetzwerke lässt sich das Sicherheitsabstand-Problem formal definieren.

**Definition 2.4 (Sicherheitsabstand-Problem, siehe [39])**

Sei ein Streckennetzwerk  $N(S, C, t, L, T)$  gegeben. Bestimme ein  $\lambda^* \in \Lambda$  mit  $\delta(\lambda^*) = \delta^*$ , wobei  $\delta^*$  den optimalen Sicherheitsabstand bezeichnet und wie folgt definiert ist:

$$\delta^* := \max_{\lambda \in \Lambda} \delta(\lambda) \quad (2.10)$$

Wie bereits erwähnt, profitieren von der Maximierung des minimalen Sicherheitsabstands lediglich stark frequentierte Haltepunkte, weniger stark befahrene Haltepunkte werden vernachlässigt. Diese Problematik adressiert das Fahrplan-Problem, indem unter den Lösungen des Sicherheitsabstand-Problems diejenige gesucht wird, die für den größtmöglichen Teil aller Haltepunkte den am jeweiligen Haltepunkt bestmöglichen Sicherheitsabstand erreicht. Dazu wird die Summe aller Sicherheitsabstände betrachtet.

**Definition 2.5 (Summe der Sicherheitsabstände in einem Streckennetzwerk, siehe [39])**

Die Summe der Sicherheitsabstände  $\delta_\Sigma(\lambda)$  unter Fahrplan  $\lambda$  ist definiert als:

$$\delta_\Sigma(\lambda) := \sum_{s \in S} \delta(s, \lambda) \quad (2.11)$$

Eine obere Schranke  $\bar{\delta}_\Sigma$  für die Summe der Sicherheitsabstände kann auf einfache Weise berechnet werden, indem die oberen Schranken aller Haltepunkte addiert werden (siehe Formel (2.12)).

<sup>2</sup> Genc zeigt in [39], dass die hier verwendete Gleichung für  $x_{ij}^s$  äquivalent ist zu  $x_{ij}^s = |a_i^s - a_j^s| \bmod ggT(T_i, T_j)$  und so die rechenaufwendigen Betrags- und Modulo-Berechnungen umgangen werden können.

$$\bar{\delta}_\Sigma = \sum_{s \in S} \bar{\delta}(s) \quad (2.12)$$

**Definition 2.6 (Fahrplan-Problem, siehe [39])**

Gegeben ein Streckennetzwerk  $N(S, C, t, L, T)$ . Bestimme ein  $\lambda^* \in \Lambda$  mit  $\delta_\Sigma(\lambda^*) = \delta_\Sigma^*$ , wobei  $\delta_\Sigma^*$  den optimalen Sicherheitsabstand bezeichnet und wie folgt definiert ist:

$$\delta_\Sigma^* := \max_{\lambda \in \Lambda: \delta(\lambda) = \delta^*} \delta_\Sigma(\lambda) \quad (2.13)$$

In dieser Formulierung des Fahrplan-Problems müssen alle Linien und Haltepunkte des Streckennetzes betrachtet werden, was unter Umständen zu einer hohen Zahl an Variablen und Ungleichungen führt. Mit Hilfe des (Multi-)Linienkonflikt-Graphen kann diese Zahl signifikant reduziert werden.

**Definition 2.7 (Linienkonflikt-Graph, siehe [39])**

Zu einem gegebenen Netzwerk  $N(S, C, t, L, T)$  definiert  $LKG(N) = (L, E)$  den Linienkonflikt-Graphen. Die Knoten in  $LKG(N)$  entsprechen den Linien des Streckennetzes und zwischen je zwei Linien existiert eine Kante genau dann, wenn die Linien einen Haltepunkt gemeinsam haben, d.h. es gilt:

$$e = (l_i, l_j) \in E \Leftrightarrow S(l_i) \cap S(l_j) \neq \emptyset \quad (2.14)$$

Abbildung 2.2 zeigt den Linienkonflikt-Graphen für Netzwerk DCL 2014 aus Abbildung 2.1.

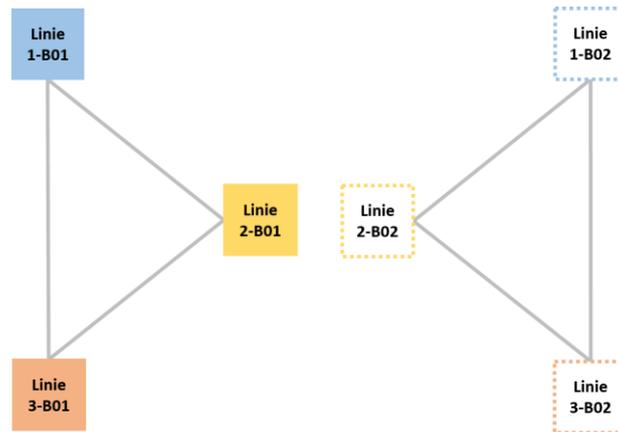


Abbildung 2.2: Linienkonflikt-Graph zum Beispielnetzwerk DCL 2014

**Definition 2.8 (Multi-Linienkonflikt-Graph, siehe [39])**

Zum Netzwerk  $N(S, C, t, L, T)$  definiert  $MLKG(N) = (L, M)$  den Multi-Linienkonflikt-Graphen. Wie beim Linienkonflikt-Graphen bilden die Linien des Streckennetzes die Knoten. In  $MLKG(N)$  existiert zwischen je zwei Linien  $l_i, l_j$  eine Kante  $(s, l_i, l_j) \in M$ , genau dann wenn beide Linien Haltepunkt  $s$  befahren, d.h. es gilt:

$$(s, l_i, l_j) \in M \Leftrightarrow s \in S(l_i) \cap S(l_j) \quad (2.15)$$

Abbildung 2.3 zeigt den Multi-Linienkonflikt-Graph für Netzwerk DCL aus Abbildung 2.1.

Bevor der Multi-Linienkonflikt-Graph verwendet werden kann um die Anzahl der Variablen und Ungleichungen des Fahrplan-Problems zu reduzieren, muss ein weiterer Sicherheitsabstand definiert werden.

**Definition 2.9 (Sicherheitsabstand eines Linienkonflikts)**

Der Sicherheitsabstand  $\delta(m, \lambda)$  einer Kante  $m \in M$  des Multi-Linienkonflikt-Graphen ist definiert als der Sicherheitsabstand zwischen den konfliktären Linien am betroffenen Haltepunkt:

$$\delta(m, \lambda) := \delta(s, l_i, l_j, \lambda) \quad m = (s, l_i, l_j) \quad (2.16)$$

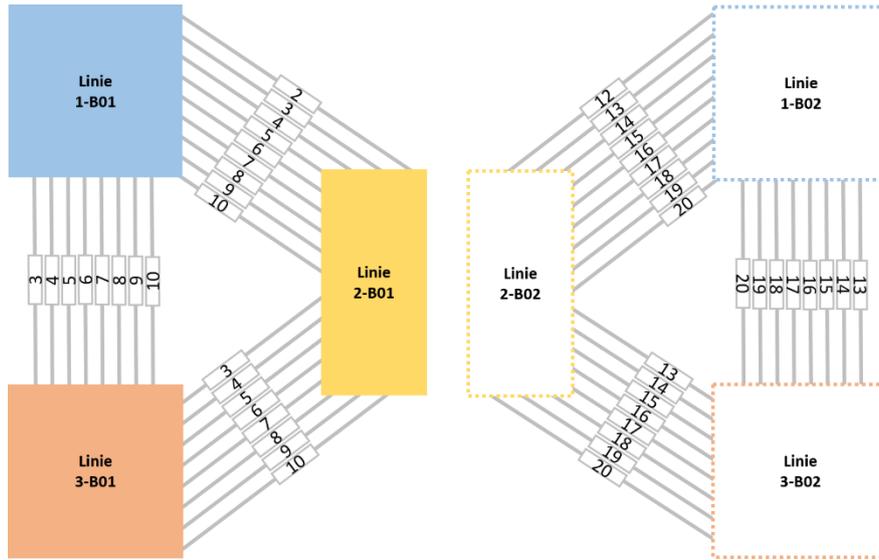


Abbildung 2.3: Multi-Linienkonflikt-Graph zum Netzwerk DCL 2014

Mit diesem Sicherheitsabstand lässt sich ebenfalls der Sicherheitsabstand eines Streckennetzwerks berechnen (siehe Formel (2.17)).

$$\delta(\lambda) = \min_{m \in M} \delta(m, \lambda) \quad (2.17)$$

Basierend auf dem Multi-Linienkonflikt-Graphen und dem in Formel (2.16) definierten Sicherheitsabstand zeigt Genç, dass zur Berechnung einer optimalen Lösung des Fahrplan-Problems lediglich eine Teilmenge der Kantenmenge des Multi-Linienkonflikt-Graphen betrachtet werden muss. Um diese Teilmenge zu erhalten wird eine Äquivalenzrelation definiert.

**Definition 2.10 (Äquivalenzrelation auf der Kantenmenge des Multi-Linienkonflikt-Graphen, siehe [39])**

Sei  $F(s_i, s_j, l)$  die Fahrzeit zwischen den Haltepunkten  $s_i$  und  $s_j$  für Linie  $l$  und sei  $M$  die Kantenmenge des Multi-Linienkonflikt-Graphen. Definiere eine Relation  $R_M$  auf  $M$  wie folgt:

$$\begin{aligned} m_1 = (s_1, l_{11}, l_{12}) \sim_{R_M} m_2 = (s_2, l_{21}, l_{22}) \\ \Leftrightarrow \{l_{11}, l_{12}\} = \{l_{21}, l_{22}\} \wedge F(s_1, s_2, l_{11}) = F(s_1, s_2, l_{12}) \end{aligned}$$

Für alle  $m_1, m_2 \in M$  für die  $m_1 \sim_{R_M} m_2$  gilt, ist dann der Sicherheitsabstand der Linienkonflikte bei gleichem Fahrplan identisch, d.h. es gilt:

$$\forall \lambda \in \Lambda: \delta(m_1, \lambda) = \delta(m_2, \lambda) \quad (2.18)$$

Sei nun  $M_0 \subseteq M$  die Repräsentantenmenge von  $M$ . Dann gilt für den Sicherheitsabstand  $\delta(\lambda)$  für ein Streckennetzwerk unter Fahrplan  $\lambda$ :

$$\forall \lambda \in \Lambda: \min_{m \in M} \delta(m, \lambda) = \min_{m \in M_0} \delta(m, \lambda) \quad (2.19)$$

Es reicht also aus zur Berechnung des minimalen Sicherheitsabstands nur die Linienkonflikte aus der Repräsentantenmenge zu betrachten.

Um die Zahl der Variablen und Ungleichungen noch weiter zu reduzieren kann eine Äquivalenzrelation auf der Menge der Haltepunkte verwendet werden.

**Definition 2.11 (Äquivalenzrelation auf der Menge der Haltepunkte, siehe [39])**

Definiere eine Relation  $R_S$  auf der Menge der Haltepunkte  $S$  wie folgt:

$$s_1 \sim_{R_S} s_2 \Leftrightarrow L(s_1) = L(s_2) \wedge \exists k \in \mathbb{Z} \forall l \in L(s_1): F(s_1, s_2, l) = k \quad (2.20)$$

Zwei Haltepunkte sind also äquivalent, wenn beide durch die gleichen Linien befahren werden und die Fahrzeiten aller Linien zwischen den beiden Haltepunkten gleich sind.

Sei nun  $S_\Sigma \subseteq S$  die Repräsentantenmenge der Faktormenge  $S/R_\Sigma$ . Da die Fahrzeiten zwischen Haltepunkten als fest angenommen werden, reicht zur Berechnung der Summe der Sicherheitsabstände  $\delta_\Sigma(\lambda)$  unter Fahrplan  $\lambda$  die Faktormenge sowie die Kardinalität der Äquivalenzklassen:

$$\forall \lambda \in \Lambda: \delta_\Sigma(\lambda) = \sum_{s \in S} \delta(s, \lambda) = \sum_{s \in S_\Sigma} |[s]_{R_\Sigma}| * \delta(s, \lambda) \quad (2.21)$$

Abbildung 2.4 zeigt beispielhaft wie ein Netzwerk durch diese Äquivalenzrelation reduziert werden kann.

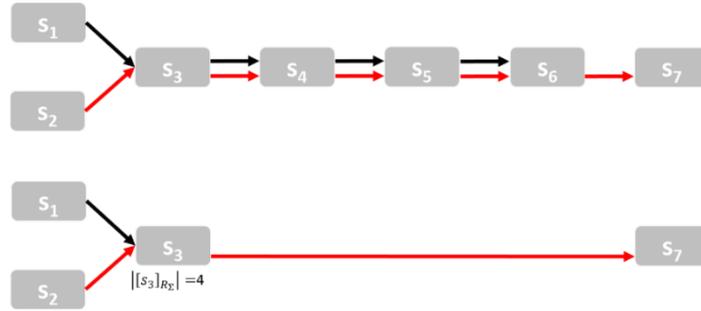


Abbildung 2.4: Beispiel der Haltepunktreduktion für einen gemeinsamen Streckenabschnitt zweier Linien. Haltepunkte  $s_4, s_5$  und  $s_6$  können durch Haltepunkt  $s_3$  repräsentiert werden. Die zugehörige Äquivalenzklasse hat eine Kardinalität von vier

Die Menge der zu betrachtenden Haltepunkte kann noch weiter reduziert werden, da für Haltepunkte die von genau einer Linie bedient werden nach Definition 2.2  $\delta(s, \lambda) = T_{L(s)} = const$  gilt. Sei  $S_{>1} \subseteq S$  die Menge der Haltepunkte, die von mehr als einer Linie bedient werden, und  $S_{=1} \subseteq S$  die Menge der Haltepunkte, die von genau einer Linie bedient werden, dann kann Formel (2.21) wie folgt umgeformt werden:

$$\forall \lambda \in \Lambda: \delta_\Sigma(\lambda) = \sum_{s \in S} \delta(s, \lambda) = \sum_{s \in S_{>1}} |[s]_{R_\Sigma}| * \delta(s, \lambda) + \sum_{s \in S_{=1}} T_{L(s)} \quad (2.22)$$

Damit kann Genç das ganzzahlige lineare Programm zur Lösung des Fahrplan-Problems formulieren:

$$\max_{\lambda \in \Lambda} \delta(\lambda) * \bar{\delta}_\Sigma + \sum_{s \in S_{>1}} |[s]_{R_\Sigma}| * \delta(s, \lambda) + \sum_{s \in S_{=1}} T_{L(s)} \quad (2.23)$$

$$a_i^s = \lambda_i + a(s, l_i, \vec{0}) \quad \forall m = (s, l_i, l_j) \in M_0 \quad (2.24)$$

$$x_{ij}^s = a_i^s - a_j^s - z_{ij}^s * ggT(T_i, T_j) \quad \forall m = (s, l_i, l_j) \in M_0 \quad (2.25)$$

$$0 \leq x_{ij}^s \leq ggT(T_i, T_j) - 1 \quad \forall m = (s, l_i, l_j) \in M_0 \quad (2.26)$$

$$\delta(m, \lambda) \leq x_{ij}^s \quad \forall m = (s, l_i, l_j) \in M_0 \quad (2.27)$$

$$\delta(m, \lambda) \leq ggT(T_i, T_j) - x_{ij}^s \quad \forall m = (s, l_i, l_j) \in M_0 \quad (2.28)$$

$$0 \leq \delta(m, \lambda) \leq \left\lfloor \frac{ggT(T_i, T_j)}{2} \right\rfloor \quad \forall m = (s, l_i, l_j) \in M_0 \quad (2.29)$$

$$\delta(s, \lambda) \leq \delta(m, \lambda) \quad \forall s \in S_{>1}, \forall m \in M(s) \quad (2.30)$$

$$\delta(\lambda) \leq \delta(s, \lambda) \quad \forall s \in S_{>1} \quad (2.31)$$

$$0 \leq \delta(s, \lambda) \leq \bar{\delta}(s) \quad \forall s \in S_{>1} \quad (2.32)$$

$$0 \leq \delta(\lambda) \leq \bar{\delta} \quad (2.33)$$

$$0 \leq \lambda_i \leq T_i - 1 \quad \forall i \in L \quad (2.34)$$

$$x_{ij}^s, \lambda_i, \delta(m, \lambda), \delta(s, \lambda), \delta(\lambda) \in \mathbb{N}_0 \quad (2.35)$$

$$z_{ij}^s \in \mathbb{Z} \quad (2.36)$$

Der erste Term in Zielfunktion (2.23) gewährleistet, dass eine optimale Lösung für das Fahrplan-Problem auch eine optimale Lösung des Sicherheitsabstand-Problems darstellt, während die letzten beiden Terme dafür sorgen, dass auch an den restlichen Haltepunkten des Netzwerks die Abfahrtszeiten möglichst gleichmäßig verteilt werden. Der minimale Sicherheitsabstand des Netzwerks wird dabei mit der oberen Schranke für die Summe der Sicherheitsabstände gewichtet, um eine gleichmäßige Gewichtung beider Zielfunktionsteile während der Optimierung zu gewährleisten.

Nebenbedingungen (2.24), (2.25), (2.26) und (2.34) definieren die Variablen für Abfahrtzeiten und Abstände zwischen Abfahrtzeiten sowie deren zulässige Wertebereiche. Nebenbedingungen (2.27) bis (2.29) stellen sicher, dass die Sicherheitsabstände zu allen relevanten Linienkonflikten im zulässigen Wertebereich bleiben. Hierbei folgen Nebenbedingungen (2.27) und (2.28) direkt aus Gleichung (2.6). Nebenbedingungen (2.30) und (2.32) sowie (2.31) und (2.33) gewährleisten die Zulässigkeit der Wertebereiche für die Sicherheitsabstände an Haltepunkten bzw. den Sicherheitsabstand des Fahrplans. Zuletzt definieren Nebenbedingungen (2.35) und (2.36) die zulässigen Wertebereiche aller Variablen. Dabei lässt Genç auch explizit Sicherheitsabstände von null – und damit simultane Abfahrten am gleichen Haltepunkt – zu, um zu gewährleisten, dass immer eine – eventuell ungültige – Lösung des Optimierungsproblems existiert.

### 2.2.2. Ein Ansatz zur Maximierung zeitlicher Sicherheitsabstände unter Berücksichtigung verkehrsplanerischer Vorgaben<sup>3</sup>

Das Modell von Ullrich et al. ist eine Erweiterung und Präzisierung eines erstmals von Franz in [36] vorgestellten Modells.

Wie bei Genç, werden auch hier zeitliche Sicherheitsabstände dazu verwendet periodische Taktfahrpläne zu generieren, die widerstandsfähig gegenüber kleinen Verspätungen sind. Ullrich et al. betrachten aber nur die Summe aller Sicherheitsabstände zwischen Linien, ohne den netzwerkweiten minimalen Sicherheitsabstand explizit zu berücksichtigen. Es wird also weder das Sicherheitsabstand- noch das Fahrplan-Problem, wie sie von Genç definiert wurden, gelöst.

Basis des Modells von Ullrich et al. ist ebenfalls ein Streckennetzwerk  $N(S, C, t, L, T)$ . Auch hier wird angenommen, dass die Fahrzeiten zwischen den einzelnen Haltepunkten fest gegeben und unabhängig vom Fahrplan sind. Dementsprechend unterscheiden sich verschiedene Fahrpläne  $\lambda \in \Lambda$  ebenfalls nur durch die  $m$  Abfahrtszeiten an den Starthaltepunkten  $\lambda_i$ . Im Unterschied zu Genç, erlaubt das Modell von Ullrich et al. keine explizite Definition unterschiedlicher Takte für verschiedene Linien. Stattdessen wird ein gemeinsamer Takt  $T$  für alle Linien verwendet. Sind unterschiedliche Takte gewünscht, so müssen Taktfrequenzen einzelner Linien mittels Verstärkerfahrten erhöht werden.<sup>4</sup>

Zur Berechnung der Regularität eines Fahrplans  $\lambda$  verwenden Ullrich et al. den Sicherheitsabstand  $\delta(s, l_i, l_j, \lambda)$  zwischen zwei Linien  $l_i, l_j \in L$  an einem Haltepunkt  $s \in S$ . Um die Anzahl der notwendigen Berechnungen zu reduzieren wird eine zeitliche Ordnung auf  $L(s)$ , der Menge der Linien die Haltepunkt  $s$  bedienen, wie folgt definiert:

$$l_j \geq_{s, \lambda} l_i \Leftrightarrow a(s, l_j, \lambda) \geq a(s, l_i, \lambda)$$

Für einen gegebenen Fahrplan  $\lambda$  sind die Linien in  $L(s)$  dann aufsteigend, basierend auf ihrer Abfahrtszeit an  $s$ , sortiert. Es müssen zur Berechnung der Regularität also nur noch die Abfahrtszeiten direkt aufeinander folgender Linien betrachtet werden. Einen Sonderfall bilden die erste und letzte Linie in  $L(s)$ . Da periodische Taktfahrpläne betrachtet werden, entspricht der Nachfolger für die letzte Linie in  $L(s)$  wieder der ersten Linie in  $L(s)$ .

Unter Verwendung von Genç' Haltepunktreduktion mittels Äquivalenzklassen (siehe Definition 2.11), ergibt sich für die Berechnung der Regularität eines Fahrplans  $\lambda$ :

<sup>3</sup> Zur besseren Vergleichbarkeit und Verständlichkeit wurden einige Variablenbezeichner angepasst und weichen von den Originalbezeichnern in [90] ab.

<sup>4</sup> Eine Definition von Verstärkerfahrten folgt auf der nächsten Seite.

$$\Phi'(\lambda) := \sum_{s \in S} \sum_{\substack{l_i, l_j \in L(s), \\ l_i \neq l_j, l_j \geq_{s,\lambda} l_i}} \frac{1}{\delta(s, l_i, l_j, \lambda)} * |[S]_{R_\Sigma}| \quad (2.37)$$

Da Ullrich et al. ein Minimierungsproblem lösen, verwenden sie den Kehrwert des Sicherheitsabstands, wodurch kleine Sicherheitsabstände an hochfrequentierten Haltepunkten automatisch bestraft werden.

Um verkehrsplanerische Vorgaben zu berücksichtigen wird dem Modell eine von der Anwenderin definierte Menge  $\Omega$  dieser Vorgaben übergeben. Eine verkehrsplanerische Vorgabe  $\omega$  bezieht sich dabei auf maximal zwei Linien sowie maximal zwei verschiedene Haltepunkte und definiert eine Reihe von Zeitintervallen mit durch die Nutzerin zugewiesenen Prioritätswerten  $\rho_{\omega_1}, \dots, \rho_{\omega_k(\omega)}$ . Diese Prioritätswerte stammen aus der Menge der möglichen Prioritäten  $P = \{1, 2, 3, \infty\}$  und bilden (aufgrund des zu lösenden Minimierungsproblems) absteigende Prioritäten ab, von 1 (hohe Priorität) über 2 (mittlere Priorität) und 3 (niedrige Priorität) bis zu  $\infty$  (unerwünscht). Mit Hilfe einer Gewichtungsfunktion  $f: \Omega \times \Lambda \rightarrow P$ , die jeder verkehrsplanerischen Vorgabe den unter dem betrachteten Fahrplan  $\lambda$  erreichten Prioritätswert zuweist, kann die Erfüllung der verkehrsplanerischen Vorgaben berechnet werden:

$$\Phi''(\lambda, \Omega) := \sum_{\omega \in \Omega} f(\omega, \lambda) \quad (2.38)$$

Zur Berücksichtigung verkehrsplanerischer Vorgaben im Modell werden diese formalisiert. Ullrich et al. verwenden dazu fünf von Walter in [101] definierte Typen von Vorgaben (siehe [36], S. 24-26 und [89], S. 120-121):

- *Abfahrtszeiten:* Für eine Linie  $l_i$  werden für einen Haltepunkt  $s \in S(l_i)$  die gewünschten Abfahrtszeiten festgelegt. Dazu weisen Verkehrsplaner jeder möglichen Abfahrtszeit  $a_i^s$  einen Prioritätswert  $\rho \in P$  zu.
- *Abstände zwischen Abfahrten:* Die Abfahrten einer Linie  $l_i$  an Haltepunkt  $s_i \in S(l_i)$  müssen in einem vorgegeben zeitlichen Abstand zu den Abfahrten einer Linie  $l_j \neq l_i$  an Haltepunkt  $s_j \in S(l_j)$  erfolgen. Jedem möglichen Abstand wird dabei ein Prioritätswert  $\rho \in P$  zugewiesen.
- *Verstärkerfahrten:* Fahrten von Linie  $l_i$  verstärken Linie  $l_j \neq l_i$  in Abschnitten mit hoher Nachfrage. Für jeden möglichen zeitlichen Abstand in dem eine Verstärkung erfolgen kann wird dabei ein Prioritätswert  $\rho \in P$  vergeben.
- *Eingleisige Streckenabschnitte:* Zur Vermeidung von Kollisionen muss zwischen zwei Fahrzeugen die einen Streckenabschnitt in unterschiedlichen Richtungen befahren ein vorgegebener zeitlicher Abstand eingehalten werden. Auch hier wird für jeden möglichen Abstand ein Prioritätswert  $\rho \in P$  vergeben.
- *Umsteigeverbindungen:* Linie  $l_i$  darf Haltepunkt  $s \in S(l_i) \cap S(l_j)$  erst  $x$  Zeiteinheiten nach Linie  $l_j$  verlassen. Für alle möglichen Werte von  $x$  werden dabei durch Verkehrsplaner Prioritätswerte  $\rho \in P$  definiert.

Mit Hilfe von Vorgaben zu Abfahrtszeiten, Abständen zwischen Abfahrten und/oder Umsteigeverbindungen können synchrone Abfahrten erzwungen und so indirekt die Fahrplansynchronisierung optimiert werden.

Wie von Ullrich in Abschnitt 6.2.3 von [89] gezeigt, bilden Vorgaben zu Abfahrtszeiten und Abständen zwischen Abfahrten elementare Vorgabentypen und alle anderen Typen können durch eine Kombination dieser elementaren Typen abgebildet werden. Bei der Optimierung kann sich folglich auf diese Vorgaben konzentriert werden. Weiterhin müssen bei der Aufstellung von Vorgaben lediglich Prioritätswerte für Zeiten aus  $\{0, \dots, T - 1\}$  definiert werden, da aufgrund der Periodizität des zu generierenden Fahrplans beim Testen der Erfüllung alle Abstände und Abfahrtszeiten modulo  $T$  berechnet werden.

Abhängig von der Größe des betrachteten Streckennetzes und der Anzahl zu beachtender Vorgaben unterscheiden sich die Wertebereiche von Gleichung (2.37) und (2.38) unter Umständen deutlich, wodurch ihre Kombination zu einer Gesamtzielfunktion erschwert wird. Um eine systematische Benachteiligung eines Terms zu vermeiden definieren Ullrich et al. deshalb einen Normalisierungsfaktor  $\beta$  für die Erfüllung der verkehrsplanerischen Vorgaben, der das Verhältnis zwischen der oberen Schranke für die Regularität und der oberen Schranke für die Erfüllung aller Vorgaben abbildet. Die obere Schranke  $\bar{\delta}(s, l_i, l_j)$  für den Sicherheitsabstand zwischen zwei Linien  $l_i, l_j$  an Haltepunkt  $s$  ergibt sich dabei, wie bei Genç, durch äquidistantes Aufteilen des vorgegeben Taktes  $T$  auf die den Haltepunkt bedienenden Linien:

$$\bar{\delta}(s, l_i, l_j) = \left\lfloor \frac{T}{|L(s)|} \right\rfloor \quad (2.39)$$

Die obere Schranke  $\bar{f}(\omega)$  für die Erfüllung der verkehrsplanerischen Vorgabe  $\omega$  entspricht, unabhängig vom konkreten Fahrplan, dem minimalen von der Nutzerin vorgegeben Prioritätswert, so das typischerweise  $\bar{f}(\omega) = 1$  gilt. Dadurch ergibt sich für den Normalisierungsfaktor<sup>5</sup>:

$$\beta := \frac{\sum_{s \in S_\Sigma} \left( \sum_{l_i, l_j \in L(s), l_i \neq l_j, l_j \geq_{s, \bar{\delta}} l_i} \left( \frac{1}{\bar{\delta}(s, l_i, l_j)} \right) \right)}{\sum_{\omega \in \Omega} \bar{f}(\omega)} \quad (2.40)$$

Mit Hilfe des Normalisierungsfaktors können Gleichungen (2.37) und (2.38) nun kombiniert und das Optimierungsmodell aufgestellt werden:

$$\min_{\lambda \in \Lambda} (1 - \alpha) * \left( \sum_{s \in S_\Sigma} \sum_{\substack{l_i, l_j \in L(s), \\ l_i \neq l_j, l_j \geq_{\alpha(s, \lambda)} l_i}} \frac{1}{\delta(s, l_i, l_j, \lambda)} * |[s]_{R_\Sigma}| \right) + \alpha * \beta * \left( \sum_{\omega \in \Omega} f(\omega, \lambda) \right) \quad (2.41)$$

$$0 \leq \lambda_i \leq T - 1 \quad \forall i \in L \quad (2.42)$$

$$\delta(s, l_i, l_j, \lambda) > 0 \quad \forall s \in S_\Sigma, \forall l_i, l_j \in L(s) \quad (2.43)$$

$$\beta = \frac{\sum_{s \in S_\Sigma} \sum_{\substack{l_i, l_j \in L(s), \\ l_i \neq l_j, l_j \geq_{\alpha(s, \bar{\delta})} l_i}} \frac{1}{\bar{\delta}(s, l_i, l_j)}}{\sum_{\omega \in \Omega} \bar{f}(\omega)} \quad (2.44)$$

$$\delta(s, l_i, l_j, \lambda) \in \mathbb{N} \quad (2.45)$$

$$0 \leq \alpha \leq 1 \quad (2.46)$$

Um eine Steuerung des Optimierungsverhaltens zu ermöglichen wird zusätzlich zum Normalisierungsfaktor der Gewichtungsfaktor  $0 \leq \alpha \leq 1$  eingeführt, mit dem festgelegt werden kann wie stark die Erfüllung der verkehrsplanerischen Vorgaben bei der Lösung des Optimierungsmodells berücksichtigt werden soll.

Um außerdem zu vermeiden, dass ein Haltepunkt zeitgleich von zwei Fahrzeugen bedient werden kann, wird in Nebenbedingung (2.43) festgelegt, dass der Sicherheitsabstand zwischen zwei Linien mindestens eine Zeiteinheit betragen muss.

<sup>5</sup> Ullrich et al. unterläuft an dieser Stelle ein Fehler. Sie vergessen den Gewichtungsfaktor  $|[s]_{R_\Sigma}|$  in den Zähler aufzunehmen, wodurch ein Teil der Güte der Abschätzung verloren geht.

### 3. Ein neuer Ansatz zur Generierung regulärer Fahrpläne unter Berücksichtigung verkehrsplanerischer Vorgaben

Während die von Ullrich et al. gewählte Problemformulierung die Berücksichtigung verkehrsplanerischer Vorgaben – und damit indirekt die Berücksichtigung synchroner Abfahrten – ermöglicht, erschwert die Nicht-Linearität der verwendeten Zielfunktion die Verwendung vieler standardisierter Softwarepakete, wie beispielsweise CPLEX. Die von Genç gewählte Problemformulierung ermöglicht hingegen die Verwendung von standardisierten Softwarepaketen, berücksichtigt aber keine verkehrsplanerischen Vorgaben. Im Folgenden soll deswegen ein neuer Ansatz vorgestellt werden, der die Vorteile der Modelle von Genç und Ullrich et al. kombiniert, um unter Verwendung von CPLEX optimale Lösungen für das Fahrplan-Problem, unter Berücksichtigung verkehrsplanerischer Vorgaben, zu berechnen. Die von Genç verwendete Darstellung des Fahrplan-Problems als ganzzahliges lineares Programm wird dafür als Basis verwendet und um die Möglichkeit erweitert verkehrsplanerische Vorgaben zu berücksichtigen. Das resultierende Optimierungsproblem ist offensichtlich NP-schwer, da wie von Genç in [39] gezeigt, das Fahrplan-Problem an sich bereits NP-schwer ist.

#### 3.1. Modell

Wie bei Genç und Ullrich et al., basiert das Modell auf einem Streckennetzwerk  $N(S, C, t, L, T)$ . Abweichend von Ullrich et al. und in Anlehnung an Genç wird dabei mit einer Menge potentiell unterschiedlicher Takte für die gegebenen  $m$  Linien gearbeitet. Es gilt also  $T = \{T_1, \dots, T_m\}$ . Ebenso werden die von Genç definierten und bereits in Abschnitt 2.2.1 beschriebenen Kenngrößen für Abfahrtszeiten und Sicherheitsabstände verwendet. Folglich wird die Regularität eines Fahrplans  $\lambda$  berechnet als:

$$\Phi'(\lambda) := \delta(\lambda) * \bar{\delta}_\Sigma + \sum_{s \in S_{\Sigma, >1}} |[S]_{R_\Sigma}| * \delta(s, \lambda) \quad (3.1)$$

Anders als im Modell von Genç wird die konstante Summe der Sicherheitsabstände an Haltepunkten mit genau einer Linie  $\sum_{s=1} T_{L(s)}$  für die Optimierung nicht berücksichtigt. Sie ist unabhängig vom Fahrplan und nur für den Vergleich verschiedener Streckennetze relevant. Folglich wird sie nach Abschluss der Optimierung zum Zielfunktionswert hinzuaddiert.

Für die Einbindung verkehrsplanerischer Vorgaben wird sich am Modell von Ullrich et al. orientiert, allerdings werden einige Anpassungen vorgenommen: Anstatt mehrere Zeitintervalle mit unterschiedlichen Prioritäten in einer Vorgabe zu vereinen, umfasst eine verkehrsplanerische Vorgabe  $\omega$  im neuen Modell nur exakt ein Zeitintervall mit genau einem zugewiesenen Prioritätswert  $\rho_\omega$ . Dieser durch die Nutzerin vorgegebene Prioritätswert stammt aus der Menge der möglichen Prioritäten  $P = \{HIGH, MEDIUM, LOW, PROHIBITED\}$ . Ein Prioritätswert von *HIGH* besagt, dass eine Bedingung hohe Bedeutung hat, während ein Prioritätswert von *PROHIBITED* eine von der Nutzerin unerwünschte Konstellation definiert. Wie bei Ullrich et al. bezieht sich eine verkehrsplanerische Vorgabe maximal auf zwei Linien sowie maximal zwei Haltepunkte.

Zur einfacheren Kombination mit Genç' Modell wird eine formale Schreibweise für verkehrsplanerische Vorgaben definiert. Dabei wird sich auf die in Abschnitt 2.2.2 eingeführten elementaren Vorgaben konzentriert. Aufgrund der gewählten Modellformulierung muss zusätzlich zwischen Abstandsvorgaben an einem gemeinsamen Haltepunkt und Umsteigevorgaben an verschiedenen Haltepunkten unterschieden werden.

#### Definition 3.1 (Abfahrtszeitvorgabe)

Sei  $\Omega_D \subseteq \Omega$  die Menge aller verkehrsplanerischen Abfahrtszeitvorgaben. Eine Abfahrtszeitvorgabe ist ein 5-Tupel  $\omega = (l_i, s, \check{t}, \bar{t}, \rho) \in \Omega_D$  und definiert für die Abfahrtszeit  $a_i^s$  von Linie  $l_i$  an Haltepunkt  $s \in S(l_i)$  eine untere Schranke  $\check{t} \in \mathbb{N}_{T_i}$  und eine obere Schranke  $\bar{t} \in \mathbb{N}_{T_i}$ , mit  $\check{t} \leq \bar{t}$ , sowie einen Prioritätswert  $\rho \in P$ :

$$\check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t} \quad z_i^s \in \mathbb{Z} \quad (3.2)$$

Da wie bei Genç und Ullrich et al. nur periodische Taktfahrpläne betrachtet werden, können die Abfahrtszeiten stets auf eine Minute innerhalb des relevanten Taktintervalls begrenzt werden. Wie bei Genç wurden auch hier bereits die rechenaufwendigen Betrags- und Modulo-Operatoren entfernt.

Mit Abfahrtszeitvorgaben können also die potentiellen Startzeiten von individuellen Linien an einzelnen Haltepunkten begrenzt bzw. priorisiert werden. Dies kann unter anderem dazu genutzt werden, um die Abfahrten des Nahverkehrs besser mit externen Diensten zu koordinieren, z.B. mit dem Unterrichtsbeginn von in der Nähe befindlichen Schulen oder den Abfahrtszeiten von Zügen des Fernverkehrs.

**Definition 3.2 (Abstandsvorgabe)**

Sei  $\Omega_H \subseteq \Omega$  die Menge aller verkehrsplanerischen Abstandsvorgaben an einen gemeinsamen Haltepunkt. Eine Abstandsvorgabe zwischen zwei Linien  $l_i, l_j$  an einem gemeinsamen Haltepunkt  $s$  ist ein 6-Tupel  $\omega = (l_i, l_j, s, \check{t}, \bar{t}, \rho) \in \Omega_H$  und definiert für den Abstand  $x_{ij}^s$  zwischen den Abfahrtszeiten von  $l_i$  und  $l_j$  an  $s \in S(l_i) \cap S(l_j)$  eine untere Schranke  $\check{t} \in \{0, \dots, ggT(T_i, T_j)\}$  und eine obere Schranke  $\bar{t} \in \{0, \dots, ggT(T_i, T_j)\}$  sowie einen Prioritätswert  $\rho \in P$ :

$$\check{t} \leq x_{ij}^s \leq \bar{t} \quad (3.3)$$

Abstandsvorgaben können unter anderem dazu verwendet werden, um die Fahrzeugauslastung einer Linie  $l_j$  ohne Einsatz von Verstärkerfahrten zu reduzieren, indem Abfahrten von Fahrzeugen einer weniger stark ausgelasteten Linie  $l_i$  mit (teilweise) gemeinsamen Linienverlauf an einem gemeinsamen Haltepunkt  $s$  so geplant werden, dass sie kurz vor den Abfahrten der Fahrzeuge von Linie  $l_j$  stattfinden. Passagiere die auf Haltepunkten des gemeinsamen Streckenabschnitts zusteigen, können so dazu veranlasst werden für diesen Teil des Linienverlaufs Fahrzeuge der weniger stark ausgelasteten Linien zu verwenden, wodurch es zu weniger starken Auslastungsspitzen kommen sollte.

**Definition 3.3 (Umsteigevorgabe)**

Sei  $\Omega_C \subseteq \Omega$  die Menge aller verkehrsplanerischen Umsteigevorgaben über verschiedene Haltepunkte. Eine Umsteigevorgabe zwischen zwei Linien  $l_i, l_j$  an zwei verschiedenen Haltepunkten  $s \in S(l_i), r \in S(l_j)$  ist ein 7-Tupel  $\omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C$  und definiert eine untere Schranke  $\check{t} \in \{0, \dots, ggT(T_i, T_j)\}$  und eine obere Schranke  $\bar{t} \in \{0, \dots, ggT(T_i, T_j)\}$  sowie einen Prioritätswert  $\rho \in P$  für den Abstand zwischen den Ankunftszeiten  $a_i^s$  und  $a_j^r$ ,  $x_{ij}^{sr}$ :

$$\check{t} \leq x_{ij}^{sr} \leq \bar{t} \quad (3.4)$$

Angelehnt an Genç sei dabei  $x_{ij}^{sr} = a_i^s - a_j^r - z_{ij}^{sr} * ggT(T_i, T_j)$  mit  $z_{ij}^{sr} \in \mathbb{Z}$  so gewählt, dass  $0 \leq x_{ij}^{sr} \leq ggT(T_i, T_j) - 1$  gilt.

Umsteigevorgaben können dazu verwendet werden, um vorgesehene Umsteigeverbindungen innerhalb eines und zwischen unterschiedlichen Verkehrssystemen bereits bei der Erstellung eines Fahrplans zu berücksichtigen. Beispielsweise kann so an einer Haltestelle für mehrere Verkehrssysteme erzwungen werden, dass die Ankunftszeit einer (Bus-)Linie  $l_i$  an ihrem Haltepunkt  $s$  zwischen  $\check{t}$  und  $\bar{t}$  Minuten vor der Ankunft einer (Bahn-)Linie  $l_j$  am Haltepunkt  $r$  liegt, um zum einen Passagieren von Fahrzeugen der Linie  $l_i$  einen komfortablen Umstieg in Fahrzeuge von Linie  $l_j$  zu gestatten. Zum anderen kann durch eine solche Vorgabe – entsprechende Kontrollstrategien vorausgesetzt – aber auch die Umsetzung einer umgekehrt gerichteten Umsteigeverbindung gesichert werden, indem ein Fahrzeug der früher ankommenden Linie  $l_i$  im laufenden Betrieb so lange verzögert wird, bis ein (Zubringer-)Fahrzeug von Linie  $l_j$  angekommen ist.<sup>6</sup>

Für jede verkehrsplanerische Vorgabe wird dem Optimierungsmodell eine der Ungleichungen (3.2), (3.3) oder (3.4) hinzugefügt. Vorgaben mit Prioritätswert *PROHIBITED* werden mit einem logischen Negationsoperator versehen. Formulieren Vorgaben des gleichen Typs Schranken für die gleichen Linien und Haltepunkte, so wird von zusammenhängenden Vorgaben gesprochen.

<sup>6</sup> Ein umfangreiches Anwendungsbeispiel für diesen Fall wird in Kapitel 8 besprochen.

**Definition 3.4 (Zusammenhängende verkehrsplanerische Vorgabe)**

Seien zwei verkehrsplanerische Vorgaben  $\omega_i = (l_i, s_i, \check{t}_i, \bar{t}_i, \rho_i)$ ,  $\omega_j = (l_j, s_j, \check{t}_j, \bar{t}_j, \rho_j) \in \Omega$ , des gleichen Typs gegeben.<sup>7</sup>  $\omega_i$  und  $\omega_j$  hängen zusammen, wenn gilt:

$$l_i = l_j \wedge s_i = s_j$$

Zusammenhängende Vorgaben beschreiben logische Beziehungen zwischen zulässigen und unzulässigen Intervallen auf dem Wertebereich einer Variable. Aus diesem Grund müssen sie während der Optimierung zusammen betrachtet werden. Dazu werden sie mit logischen Operatoren verbunden. Vorgaben mit Prioritätswert *HIGH*, *MEDIUM* oder *LOW* werden mit Oder verknüpft, während negierte Vorgaben des Prioritätswerts *PROHIBITED* mit Und verknüpft werden. Anschließend werden die beiden resultierenden Teilbedingungen mit Oder verknüpft.

Als Beispiel seien drei zusammenhängende Abfahrtszeitvorgaben für Linie  $l_1$  an Haltepunkt 1 gegeben:  $\omega_1 = (1,1,0,2, \text{HIGH})$ ,  $\omega_2 = (1,1,3,4, \text{MEDIUM})$  und  $\omega_3 = (1,1,5,9, \text{PROHIBITED})$

Die aus den Vorgaben resultierenden Ungleichungen werden dann wie folgt verknüpft:

$$((0 \leq a_1^1 - z_1^1 * T_1 \leq 2) \vee (3 \leq a_1^1 - z_1^1 * T_1 \leq 4)) \vee \neg(5 \leq a_1^1 - z_1^1 * T_1 \leq 9)$$

Nicht durch zusammenhängende Vorgaben abgedeckte Teile eines Wertebereichs werden standardmäßig als unerwünscht deklariert. D.h. insbesondere, dass Vorgaben mit Prioritätswerten *HIGH*, *MEDIUM* oder *LOW* und engen Schranken die Wertebereiche betroffener Variablen stärker einschränken können als Vorgaben mit Prioritätswert *PROHIBITED*. Würde im obigen Beispiel beispielsweise nur Vorgabe  $\omega_1$  verwendet, so wäre der Wertebereich von  $a_1^1 - z_1^1 * T_1$  auf  $[0, 2]$  beschränkt. Im Gegensatz dazu hätte die isolierte Verwendung von Vorgabe  $\omega_3$  eine Beschränkung des Wertebereichs auf  $[0, 4]$  zur Folge.

Neben den Ungleichungen (3.2), (3.3) und (3.4) wird dem Optimierungsmodell außerdem für jede verkehrsplanerische Vorgabe  $\omega \in \Omega$  eine binäre Variable  $b_\omega \in \{0,1\}$  hinzugefügt, die genau dann den Wert 1 annimmt wenn die zugehörige Nebenbedingung erfüllt ist.

Die Erfüllung der verkehrsplanerischen Vorgaben unter Fahrplan  $\lambda$  ergibt sich dann durch die Summierung der mit  $b_\omega$  multiplizierten Gewichte der zu den verkehrsplanerischen Vorgaben gehörenden Prioritätswerte:

$$\Phi''(\lambda, \Omega) := \sum_{\omega \in \Omega} b_\omega * f(\rho_\omega) \quad (3.5)$$

Abweichend zum Modell von Ullrich et al. assoziiert die Gewichtungsfunktion  $f$  hier also nicht die spezifische (erfüllte) Vorgabe direkt mit ihrem Prioritätswert, sondern den zu einer Vorgabe definierten Prioritätswert mit einem durch die Nutzerin modifizierbaren Gewicht, es gilt also  $f: P \rightarrow \mathbb{R}$ .

Wie bei Ullrich et al. können sich die Wertebereiche der Gleichungen (3.1) und (3.5), abhängig von der Größe des betrachteten Streckennetzes und der Anzahl der zu beachtenden Vorgaben, deutlich unterscheiden. Die Gesamtziel Funktion kann folglich nicht als einfache Kombination von Gleichungen (3.1) und (3.5) aufgestellt werden. In Anlehnung an Ullrich et al. wird deshalb ebenfalls ein Normalisierungsfaktor  $\beta$  für die Erfüllung der verkehrsplanerischen Vorgaben verwendet:

$$\beta := \frac{\bar{\delta} * \bar{\delta}_\Sigma + \sum_{s \in S_{\Sigma, > 1}} |[s]_{R_\Sigma}| * \bar{\delta}(s)}{|\Omega| * f(\text{HIGH})} \quad (3.6)$$

Der Normalisierungsfaktor entspricht auch hier dem Verhältnis zwischen der oberen Schranke für die Regularität eines Fahrplans und der oberen Schranke für die Erfüllung der verkehrsplanerischen Vorgaben. Zur Berechnung der oberen Schranke für die Regularität werden die oberen Schranken für die einzelnen Sicherheitsabstände, wie von Genç in [39] definiert und in Abschnitt 2.2.1 beschrieben, in Gleichung

<sup>7</sup> Der Einfachheit halber wurden hier Abfahrtszeitvorgaben verwendet. Die Definition gilt analog auch für beide Formen von Abstandsvorgaben.

chung (3.1) eingesetzt. Eine einfache obere Schranke für die Erfüllung der verkehrsplanerischen Vorgaben ergibt sich durch die Multiplikation des theoretisch bestmöglichen Gewichts  $f(HIGH)$  mit der Anzahl der definierten verkehrsplanerischen Vorgaben  $|\Omega|$ . Damit ergibt sich das Optimierungsmodell:

$$\max_{\lambda \in A} (1 - \alpha) * \left( \delta(\lambda) * \bar{\delta}_\Sigma + \sum_{s \in S_{\Sigma, > 1}} |[s]_{R_\Sigma}| * \delta(s, \lambda) \right) + \alpha * \beta * \left( \sum_{\omega \in \Omega} b_\omega * f(\rho_\omega) \right) \quad (3.7)$$

$$a_i^s = \lambda_i + a(s, l_i, \vec{0}) \quad \forall m = (s, l_i, l_j) \in M_0 \quad (3.8)$$

$$x_{ij}^s = a_i^s - a_j^s - z_{ij}^s * ggT(T_i, T_j) \quad \forall m = (s, l_i, l_j) \in M_0 \quad (3.9)$$

$$0 \leq x_{ij}^s \leq ggT(T_i, T_j) - 1 \quad \forall m = (s, l_i, l_j) \in M_0 \quad (3.10)$$

$$\delta(m, \lambda) \leq x_{ij}^s \quad \forall m = (s, l_i, l_j) \in M_0 \quad (3.11)$$

$$\delta(m, \lambda) \leq ggT(T_i, T_j) - x_{ij}^s \quad \forall m = (s, l_i, l_j) \in M_0 \quad (3.12)$$

$$1 \leq \delta(m, \lambda) \leq \left\lfloor \frac{ggT(T_i, T_j)}{2} \right\rfloor \quad \forall m = (s, l_i, l_j) \in M_0 \quad (3.13)$$

$$\delta(s, \lambda) \leq \delta(m, \lambda) \quad \forall s \in S_{\Sigma, > 1}, \forall m \in M(s) \quad (3.14)$$

$$\delta(\lambda) \leq \delta(s, \lambda) \quad \forall s \in S_{\Sigma, > 1} \quad (3.15)$$

$$1 \leq \delta(s, \lambda) \leq \bar{\delta}(s) \quad \forall s \in S_{\Sigma, > 1} \quad (3.16)$$

$$1 \leq \delta(\lambda) \leq \bar{\delta} \quad (3.17)$$

$$0 \leq \lambda_i \leq T_i - 1 \quad \forall i \in L \quad (3.18)$$

$$\left( \bigvee_{\omega, \rho_\omega \neq PROH.} \check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t} \right) \bigvee \left( \bigwedge_{\omega, \rho_\omega = PROH.} \neg(\check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t}) \right) \quad \forall \omega \text{ zshg.}, \omega \in \Omega_D \quad (3.19)$$

$$b_\omega = \begin{cases} 1, & \check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t} \\ 0, & \neg(\check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t}) \end{cases} \quad \forall \omega = (l_i, s, \check{t}, \bar{t}, \rho) \in \Omega_D, \rho \neq PROHIBITED \quad (3.20)$$

$$b_\omega = \begin{cases} 1, & \neg(\check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t}) \\ 0, & \check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t} \end{cases} \quad \forall \omega = (l_i, s, \check{t}, \bar{t}, \rho) \in \Omega_D, \rho = PROHIBITED \quad (3.21)$$

$$\left( \bigvee_{\omega, \rho_\omega \neq PROH.} \check{t} \leq x_{ij}^s \leq \bar{t} \right) \bigvee \left( \bigwedge_{\omega, \rho_\omega = PROH.} \neg(\check{t} \leq x_{ij}^s \leq \bar{t}) \right) \quad \forall \omega \text{ zshg.}, \omega \in \Omega_H \quad (3.22)$$

$$b_\omega = \begin{cases} 1, & \check{t} \leq x_{ij}^s \leq \bar{t} \\ 0, & \neg(\check{t} \leq x_{ij}^s \leq \bar{t}) \end{cases} \quad \forall \omega = (l_i, l_j, s, \check{t}, \bar{t}, \rho) \in \Omega_H, \rho \neq PROHIBITED \quad (3.23)$$

$$b_\omega = \begin{cases} 1, & \neg(\check{t} \leq x_{ij}^s \leq \bar{t}) \\ 0, & \check{t} \leq x_{ij}^s \leq \bar{t} \end{cases} \quad \forall \omega = (l_i, l_j, s, \check{t}, \bar{t}, \rho) \in \Omega_H, \rho = PROHIBITED \quad (3.24)$$

$$x_{ij}^{sr} = a_i^s - a_j^r - z_{ij}^{sr} * ggT(T_i, T_j) \quad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C \quad (3.25)$$

$$0 \leq x_{ij}^{sr} \leq ggT(T_i, T_j) - 1 \quad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C \quad (3.26)$$

$$\left( \bigvee_{\omega, \rho_\omega \neq PROH.} \check{t} \leq x_{ij}^{sr} \leq \bar{t} \right) \bigvee \left( \bigwedge_{\omega, \rho_\omega = PROH.} \neg(\check{t} \leq x_{ij}^{sr} \leq \bar{t}) \right) \quad \forall \omega \text{ zshg.}, \omega \in \Omega_C \quad (3.27)$$

$$b_\omega = \begin{cases} 1, & \check{t} \leq x_{ij}^{sr} \leq \bar{t} \\ 0, & \neg(\check{t} \leq x_{ij}^{sr} \leq \bar{t}) \end{cases} \quad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C, \rho \neq PROHIBITED \quad (3.28)$$

$$b_\omega = \begin{cases} 1, & \neg(\check{t} \leq x_{ij}^{sr} \leq \bar{t}) \\ 0, & \check{t} \leq x_{ij}^{sr} \leq \bar{t} \end{cases} \quad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C, \rho = PROHIBITED \quad (3.29)$$

$$\beta = \frac{\bar{\delta} * \bar{\delta}_\Sigma + \sum_{s \in S_{\Sigma, > 1}} |[s]_{R_\Sigma}| * \bar{\delta}(s)}{|\Omega| * f(HIGH)} \quad (3.30)$$

$$x_{ij}^s, x_{ij}^{sr}, \lambda_i \in \mathbb{N}_0 \quad (3.31)$$

$$\delta(m, \lambda), \delta(s, \lambda), \delta(\lambda) \in \mathbb{N} \quad (3.32)$$

$$z_i^s, z_{ij}^s \in \mathbb{Z} \quad (3.33)$$

$$b_\omega \in \{0, 1\} \quad (3.34)$$

$$0 \leq \alpha \leq 1 \quad (3.35)$$

Zielfunktion (3.7) kombiniert die Maximierung der Serviceregularität (durch Lösen des Fahrplan-Problems) mit der Maximierung der Erfüllung gegebener verkehrsplanerischer Vorgaben (normalisiert mit  $\beta$ ). Um den Einfluss der beiden Zielfunktionsterme auf den Optimierungsprozess zu steuern wird zusätzlich Gewichtungsfaktor  $0 \leq \alpha \leq 1$  eingeführt.

Nebenbedingungen (3.8) bis (3.18) definieren Abfahrtszeiten, Abstände zwischen Abfahrtszeiten, Sicherheitsabstände sowie zugehörige Wertebereiche und entsprechen Nebenbedingungen (2.24) bis (2.34) aus Genç' Modell. Im Gegensatz zu Genç' Modell werden jedoch keine Sicherheitsabstände von null zugelassen, zeitgleiche Abfahrten an gemeinsamen Haltepunkten sollen vermieden werden.

Nebenbedingungen (3.19) bis (3.21) behandeln Abfahrtszeitvorgaben. Dabei wird zwischen Vorgaben mit Prioritätswerten *HIGH*, *MEDIUM*, *LOW* und solchen mit Prioritätswert *PROHIBITED* unterschieden, da letztere genau dann erfüllt sind, wenn die von ihnen definierten Wertebereiche nicht angenommen werden. Der zugehörige Binärwert  $b_\omega$  nimmt also unter umgekehrten Bedingungen den Wert 1 an.

Nebenbedingungen (3.22) bis (3.24) behandeln Abstandsvorgaben und folgen dem gleichen Muster wie Abfahrtszeitvorgaben. Nebenbedingungen (3.25) und (3.26) definieren die Variablen und Wertebereiche für Abstände zwischen Abfahrten an unterschiedlichen Haltepunkten. Sie werden von Nebenbedingungen (3.27) bis (3.29) zur Abbildung von Umsteigevorgaben verwendet.

Schließlich definiert Nebenbedingung (3.30) den Normalisierungsfaktor und Nebenbedingungen (3.31) bis (3.35) die Wertebereiche der restlichen Variablen.

### 3.2. Implementierung

Zur Lösung des Modells wurde eine Anwendung in Java 8 realisiert, die die für das Modell notwendigen Daten aus einer Datenbank<sup>8</sup>, sowie verschiedene Parameter und verkehrsplanerische Vorgaben aus Steuerungsdateien liest. Nach einem Preprocessingschritt, in dem die Netz- und Liniendaten, verkehrsplanerische Vorgaben, sowie weitere Programmparameter eingelesen und mit Hilfe der Java-Bibliothek von IBM ILOG CPLEX eine geeignete Modellformulierung generiert wird, wird selbiges zur Berechnung der Optimallösung verwendet. Anschließend wird im Rahmen eines Postprocessingschritts das von CPLEX berechnete Ergebnis in eine Menge von Fahrten mit zugehörigen konkreten Abfahrtszeiten umgewandelt, die in der Datenbank gespeichert werden kann.

Die Entwicklung fand unter *Microsoft Windows 7* (64 Bit) statt, es wurden jedoch keine für das Betriebssystem proprietären Bibliotheken verwendet. Die zur Anbindung von *Microsoft Access* Datenbanken verwendeten externen Bibliotheken sind unter (z.T. unterschiedlichen) Open Source Lizenzen veröffentlicht und liegen der Anwendung bei. Die Optimierungsanwendung ist unter allen Betriebssystemen mit entsprechender Java- und CPLEX-Version lauffähig.

Zur Entwicklung wurde *Eclipse Mars* in der Version 4.5.1 verwendet, zur Datenbankankbindung wurde *UCanAccess 2.0.9.5* (siehe [65], veröffentlicht unter GNU Lesser General Public License 2.1), *Jackcess 2.1.2* (siehe [43], veröffentlicht unter Apache License 2.0), *HSQLDB 2.3.2* (siehe [44], veröffentlicht unter eigener, auf BSD basierender Lizenz) sowie *Apache Commons Lang 2.6* und *Apache Commons Logging 1.2* (siehe [4] bzw. [5], beide unter Apache License 2.0 veröffentlicht) verwendet. Die reine Optimierungsanwendung umfasst ca. 1.600 Zeilen Quellcode verteilt auf 8 Klassen und 4 Enumerations-typen. Zusätzlich werden eine Reihe allgemeiner, auch von der Simulationsanwendung verwendeter, Klassen eingesetzt, die bspw. zur Repräsentation der logischen und physischen Struktur des Streckennetzes sowie zur Datenbankankbindung dienen.<sup>9</sup>

### 3.3. Beschleunigung des Optimierungsverfahrens

Die Komplexität des für ein gegebenes Streckennetz zu lösenden Optimierungsproblems hängt von der Zahl der Linienkonflikte  $|M_0|$ , der Zahl der von mehr als einer Linie bedienten Haltepunkte  $S_{\Sigma, >1}$ , der Zahl der im Netzwerk verkehrenden Linien  $|L|$  und deren Takte  $T$ , sowie den gegebenen verkehrsplanerischen Bedingungen  $|\Omega_D|$ ,  $|\Omega_H|$ ,  $|\Omega_C|$  ab. Konkret enthält ein Optimierungsproblem für ein gegebenes Streckennetz mindestens

<sup>8</sup> Eine Beschreibung des Datenbankmodells findet sich in Anhang A.

<sup>9</sup> Detaillierte Angaben zu diesen Klassen werden in Kapitel 5 gemacht.

$$6 * |M_0| + 3 * |S_{\Sigma, >1}| + |L| + 2 * |\Omega_D| + 2 * |\Omega_H| + 4 * |\Omega_C| + 1$$

Nebenbedingungen. Der erste Term ergibt sich aus Nebenbedingungen (3.8) bis (3.13), der zweite Term aus Nebenbedingungen (3.14) bis (3.16). Term  $|L|$  kommt aufgrund von Nebenbedingung (3.18) hinzu, während die Terme vier bis sechs aufgrund von Nebenbedingungen (3.19) bis (3.29) hinzukommen. Zuletzt ergibt sich noch eine weitere Nebenbedingung aufgrund von Nebenbedingung (3.17).

Neben den Nebenbedingungen enthält das Optimierungsproblem mindestens

$$5 * |M_0| + |S_{\Sigma, >1}| + |L| + 2 * |\Omega_D| + |\Omega_H| + 4 * |\Omega_C| + 1$$

Variablen. Der Suchraum des Optimierungsproblems umfasst im schlimmsten Fall  $\prod_{l \in L} T_l$  potentielle Lösungen. Um trotz dieses Umfangs und der vielen Einflussfaktoren eine annehmbare Laufzeit des Verfahrens zu gewährleisten sollen im Folgenden zwei Ideen zur potentiellen Beschleunigung des Verfahrens vorgestellt werden, die im Rahmen des Preprocessings angewendet werden können.

### 3.3.1. Zerlegung des (Multi-)Linienkonflikt-Graphen in Zusammenhangskomponenten

Genç zeigt in [39], dass ein Streckennetz, das aus mehreren physisch unabhängigen Zusammenhangskomponenten besteht, disjunkte Linienkonflikt- sowie Haltepunkt mengen besitzt. Für diese disjunkten Mengen können unabhängige Optimierungsprobleme mit stark reduziertem Lösungsraum aufgestellt und gelöst werden. Anschließend kann ein Fahrplan für das Gesamtsystem als Kombination der unabhängigen Teilfahrpläne erstellt werden.

Beachtet werden muss, dass Zusammenhangskomponenten nur dann unabhängig voneinander optimiert werden können, falls keine verkehrsplanerischen Vorgaben existieren, deren Parameter unterschiedliche Zusammenhangskomponenten betreffen. Andernfalls sind die Teilnetze zwar nicht physisch, aber logisch miteinander verbunden und müssen gemeinsam optimiert werden.

Für die Optimierungsanwendung wurde eine einfache Version dieses Verfahrens, ohne Berücksichtigung von verkehrsplanerischen Vorgaben, implementiert: Auf Wunsch der Nutzerin werden während des Preprocessings automatisch die Zusammenhangskomponenten des gegebenen Netzwerks identifiziert und für jedes der unabhängigen Teilnetze ein eigenständiges Optimierungsproblem aufgestellt. Nach der – wahlweise sequentiellen oder parallelen – Optimierung der Teilnetze wird dann im Postprocessing der Gesamtfahrplan erstellt.

### 3.3.2. Zerlegung der Menge der verkehrsplanerischen Vorgaben

Verkehrsplanerische Vorgaben begrenzen die zulässigen Wertebereiche der von ihnen betroffenen Modellvariablen. Sie limitieren damit den Lösungsraum, wodurch die Laufzeit der Optimierung (positiv) beeinflusst werden kann.

Sei ein zusammenhängendes (Teil-)Streckennetzwerk sowie eine Menge zugehöriger verkehrsplanerischer Vorgaben  $\Omega$  gegeben. Für alle zusammenhängenden Vorgaben  $\omega_i, \omega_j \in \Omega$  gelte:

$$[\check{t}_i, \bar{t}_i] \cap [\check{t}_j, \bar{t}_j] = \emptyset \wedge \rho_{\omega_i} \neq \rho_{\omega_j} \quad (3.36)$$

Zusammenhängende Vorgaben die diese Bedingung erfüllen, können nie zeitgleich erfüllt sein. Dies kann ausgenutzt werden, um reduzierte Mengen verkehrsplanerischer Vorgaben zu bestimmen, die jeweils nur eine der Bedingung (3.36) erfüllenden zusammenhängenden Vorgaben beinhalten. Sie stellen also jeweils mögliche Kombinationen von *nicht*-zusammenhängenden verkehrsplanerischen Vorgaben dar. Für jede dieser Mengen kann ein eigenständiges Optimierungsproblem mit stark eingeschränkten Wertebereichen gelöst werden. Durch diese Beschränkung des Wertebereichs werden die Entscheidungsmöglichkeiten der Optimierungsanwendung begrenzt, wodurch schneller eine (unter der gegebenen Teilmenge verkehrsplanerischer Vorgaben) optimale Lösung gefunden werden kann. Um die globale Lösung des ursprünglichen Optimierungsproblems zu bestimmen, muss nach Abschluss der einzelnen Optimierungen noch ein Vergleich der jeweiligen Zielfunktionswerte erfolgen.

Aufgrund der Vorab-Fixierung der verkehrsplanerischen Vorgaben kann der Erfüllungswert  $\Phi''(\lambda, \Omega)$  für jedes separate Optimierungsproblem vorab bestimmt werden. Es muss folglich nur noch der Fahrplan bestimmt werden, der unter der reduzierten Menge an Vorgaben die bestmögliche Regularität erreicht.

Problematisch an diesem Verfahren ist, dass für jede mögliche Vorgabenkombination, die sich aufgrund von Bedingung (3.36) ergibt, ein Fahrplan-Problem zu lösen ist. Es ist daher im Allgemeinen nur für Fälle geeignet bei denen die Zahl der Vorgaben die Bedingung (3.36) erfüllen klein ist. Andernfalls wird der positive Effekt der Wertebereichsbegrenzung durch die notwendige Anzahl zu lösender Optimierungsprobleme aufgehoben. Eine Verbesserung dieses Ansatzes könnte erreicht werden, wenn die Menge potentieller Kombinationen nicht-zusammenhängender Vorgaben vorab weiter reduziert werden könnte. Stahlberg untersucht dazu in [82] einen vielversprechenden, aber letztlich nicht erfolgreichen, Ansatz auf Basis einer Formulierung des Fahrplan-Problems mit verkehrsplanerischen Vorgaben als Constraint Satisfaction Problem (siehe z.B. [74]).

### 3.4. Validierung

Um die Funktionsfähigkeit des Optimierungsansatzes nachzuweisen, werden mit IBM ILOG CPLEX 12.5 erste Untersuchungen an Modellen unterschiedlicher Streckennetze durchgeführt.<sup>10</sup> Bei den Streckennetzen handelt es sich um das künstliche Netzwerk DCL 2014 (siehe Abbildung 2.1), das leicht komplexere, ebenfalls künstliche Netzwerk UCL 2015 (siehe Abbildung 3.2) und das reale Busnetz der Stadt Köln aus dem Jahr 2001 (siehe Abbildung 3.3 bis Abbildung 3.6). Für alle Netzwerke werden 12-Stunden-Fahrpläne von sieben Uhr morgens bis sieben Uhr abends erstellt. Die wichtigsten Kenngrößen der Streckennetze werden in Tabelle 3.1 zusammengefasst. Alle Untersuchungen werden auf einem AMD Athlon 64 X2 Dual Core mit 2.10 GHz und 4 GB RAM durchgeführt. Sofern Fahrpläne unter Berücksichtigung von verkehrsplanerischen Vorgaben erstellt werden, werden die folgenden Gewichte für Prioritätswerte verwendet:

$$f(HIGH) = 1; f(MEDIUM) = 0,25; f(LOW) = 0,125; f(PROHIBITED) = 0,03125.$$

Netz	L	M	M <sub>0</sub>	S	S <sub>Σ&gt;1</sub>
<b>DCL 2014</b>	6	50	6	20	4
<b>UCL 2015</b>	8	64	17	40	13
<b>Köln 2001</b>	68	313	179	1.242	126

Tabelle 3.1: Kenngrößen der untersuchten Streckennetze

#### 3.4.1. Experimente anhand künstlicher Testnetze

Zunächst soll die Korrektheit des Optimierungsverfahrens überprüft werden. Dazu werden jeweils Fahrpläne mit und ohne verkehrsplanerische Vorgaben für die künstlichen Netze DCL 2014 und UCL 2015 generiert. Anschließend wird die Zulässigkeit der generierten Fahrpläne überprüft und ihre wichtigsten Kennzahlen mit den zugehörigen oberen Schranken verglichen.

##### 3.4.1.1. DCL 2014

DCL 2014 wurde ausgewählt, da seine künstliche regelmäßige Struktur eine einfache Korrektheitsüberprüfung des Optimierungsverfahrens ermöglicht. Für die Experimente werden Fahrpläne mit einem einheitlichen Takt von zehn Minuten generiert. Ohne die Hinzunahme von verkehrsplanerischen Vorgaben kann ein Fahrplan unter diesen Bedingungen höchstens einen netzwerkweiten Sicherheitsabstand von  $\bar{\delta}(\lambda) = 3$  Minuten erreichen. Dabei entsprechen die Sicherheitsabstände zwischen Linien 1-B01, 2-B01, 3-B01 bzw. 1-B02, 2-B02, 3-B02 an ihren gemeinsamen Haltepunkten (3 bis 10 resp. 20 bis 13) einer Permutation des Tupels (3,3,4). Der Zielfunktionswert eines zugehörigen Fahrplans hat die folgende obere Schranke:

$$\frac{(3 * 78)}{\bar{\delta}(\lambda) * \bar{\delta}_{\Sigma}} + \frac{(1 * 5 + 1 * 5 + 3 * 8 + 3 * 8)}{\sum_{s \in S_{\Sigma > 1}} |s|_{R_{\Sigma}} * \bar{\delta}(s)} + \frac{(10 + 10)}{\sum_{s \in S_{=1}} \bar{\delta}(s)} = 312$$

Der durch die Optimierungsanwendung berechnete optimale Fahrplan erreicht einen Zielfunktionswert von 310 und einen netzwerkweiten Sicherheitsabstand von drei Minuten. Der Unterschied von einem Prozent zwischen dem erreichten Zielfunktionswert und der berechneten oberen Schranke kann dabei durch genauere Betrachtung der Schrankenberechnung für Sicherheitsabstände an Haltepunkten

<sup>10</sup> Ausführliche Anwendungsfälle werden in den Kapiteln 5 und 8 beschrieben.

erklärt werden: Bei der Berechnung dieser Schranke wird jeder Haltepunkt isoliert betrachtet, d.h. sie ergibt sich als gleichmäßige Aufteilung des gemeinsamen Taktintervalls der am Haltepunkt abfahrenden Linien. Solange benachbarte Haltepunkte von den gleichen Linien im gleichen Takt bedient werden stimmt die obere Schranke mit dem tatsächlich erreichbaren Wert überein. Kommen aber an benachbarten Haltepunkten Linien hinzu oder fallen weg, kann am durch weniger Linien bedienten Haltepunkt unter Umständen nicht mehr die theoretische obere Schranke für den Sicherheitsabstand erreicht werden ohne vom Gesamtoptimum des Systems abzuweichen. Als Beispiel betrachte man Haltepunkte 2 und 3 von DCL 2014 (siehe Abbildung 3.1): Haltepunkt 2 wird lediglich von Linien 1-B01 und 2-B01 bedient, hat folglich eine obere Schranke für den Sicherheitsabstand von  $\bar{\delta}(2) = 5$  Minuten. An Haltepunkt 3 kommt Linie 3-B01 hinzu, es ergibt sich eine obere Schranke von  $\bar{\delta}(3) = 3$  Minuten. Da die folgenden Haltepunkte 4 bis 10 – und damit der Großteil von DCL 2014 – ebenfalls von diesen drei Linien bedient werden, wird der Optimierer das Erreichen des optimalen Sicherheitsabstands an diesen Haltepunkten – und damit des Systemoptimums – priorisieren. Als Resultat werden Linien 1-B01 und 2-B01 an diesen Haltepunkten mit einem Sicherheitsabstand von vier Minuten arrangiert. Da die Fahrzeiten zwischen Haltepunkten als fest angenommen werden, ergibt sich so ebenfalls ein Sicherheitsabstand von vier Minuten an Haltepunkt 2 (siehe Abbildung 3.1, i)). Wird stattdessen an Haltepunkt 2 die berechnete obere Schranke als Sicherheitsabstand verwendet, so kann an Haltepunkten 3 bis 10 nur noch ein Sicherheitsabstand von zwei Minuten und damit nicht mehr die für das Gesamtsystem optimale Regularität erreicht werden (siehe Abbildung 3.1, ii)).

Um die Auswirkungen der Hinzunahme von verkehrsplanerischen Bedingungen zu untersuchen, soll Fall ii) aus Abbildung 3.1 mit Hilfe der beiden Abstandsvorgaben  $\omega_1 = (1 - B01, 2 - B01, 2, 5, 5, HIGH)$  und  $\omega_2 = (1 - B02, 2 - B02, 12, 5, 5, HIGH)$  erzwungen werden. Sie legen für Linien 1-B01 und 2-B01 an Haltepunkt 2 bzw. Linien 1-B02 und 2-B02 an Haltepunkt 12 einen Sicherheitsabstand von fünf Minuten fest. Mit einem Gewichtungsfaktor von  $\alpha = 0,5$  berechnet CPLEX einen Fahrplan mit Zielfunktionswert  $0,5 * 218 + 0,5 * 146 * 2 = 255$  und einem netzwerkweiten Sicherheitsabstand von zwei Minuten. Man sieht: Die verkehrsplanerischen Vorgaben werden eingehalten. Allerdings hat die erzwungene Verwendung des theoretisch optimalen Sicherheitsabstands, wie erwartet, eine schlechtere Abfahrtsanordnung an Haltepunkten 3 bis 10 bzw. 13 bis 20 und damit eine geringere Serviceregularität zur Folge.

Da DCL 2014 aus zwei unabhängigen Teilnetzen besteht, können Fahrpläne für dieses Netz auch durch eine getrennte Optimierung der beiden Zusammenhangskomponenten generiert werden. DCL

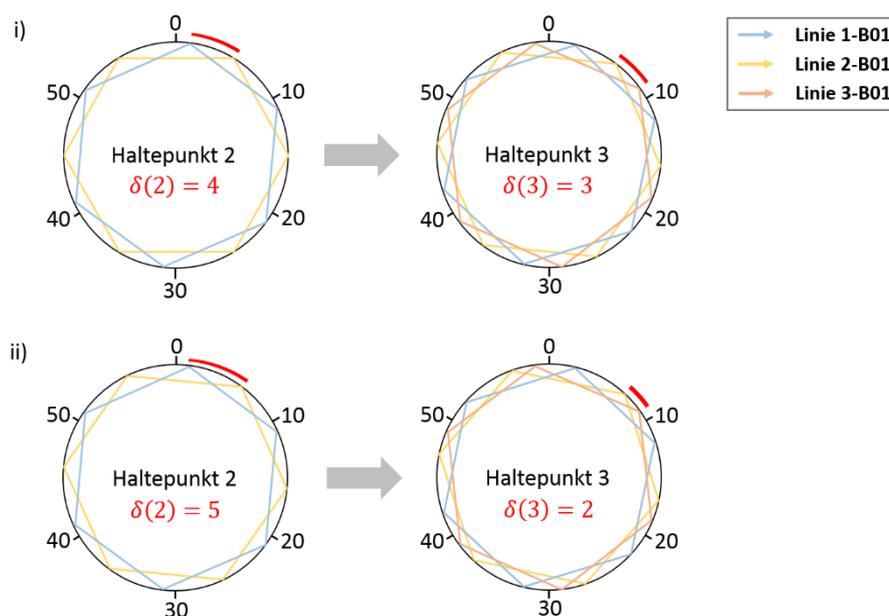


Abbildung 3.1: Abfahrten von Linie 1-B01, 2-B01 und 3-B01 an Haltepunkten 2 und 3 von DCL 2014 unter Verwendung unterschiedlicher Sicherheitsabstände. Dargestellt sind alle Abfahrten innerhalb einer Stunde

2014 wird dabei in ein Teilnetz mit den Linien 1-B01, 2-B01 und 3-B01 sowie ein Teilnetz mit den Linien 1-B02, 2-B02 und 3-B02 aufgeteilt. Ohne die Berücksichtigung verkehrsplanerischer Vorgaben haben die resultierenden, optimalen Teilfahrpläne jeweils einen Zielfunktionswert von  $155 = \frac{310}{2}$  und einen netzwerkweiten Sicherheitsabstand von drei Minuten. Wie Tabelle 3.2 und Tabelle 3.3 zeigen unterscheiden sich die mit und ohne Aufteilung erhaltenen Fahrpläne zwar in der konkreten Ausgestaltung der Abfahrtszeiten einiger Linien, die Optimalität bleibt aber erhalten.

Linie	1-B01	1-B02	2-B01	2-B02	3-B01	3-B02
<b>Ohne Aufteilung</b>	0	0	5	4	9	7
<b>Mit Aufteilung</b>	3	4	0	0	8	7

*Tabelle 3.2: Abfahrtszeiten an Starthaltepunkten mit und ohne Aufteilung, DCL 2014*

Haltepunkt	1 / 11	2 / 12	3-10 / 13-20
<b>Ohne Aufteilung</b>	10	4	3
<b>Mit Aufteilung</b>	10	4	3

*Tabelle 3.3: Sicherheitsabstände an Haltepunkten mit und ohne Aufteilung, DCL 2014*

### 3.4.1.2. UCL 2015

Netz UCL 2015 (siehe Abbildung 3.2) ist mit acht Linien und 40 Haltepunkten nur minimal umfangreicher als DCL 2014, bildet im Gegensatz aber eine realistischere, weil unregelmäßige, Netzstruktur ab und beinhaltet mit Kreislinien 2-B01 und 2-B02 außerdem einen speziellen Linienverlauf. Für die Experimente werden Fahrpläne ohne einheitlichen Takt generiert: Linien 1-B01, 1-B02, 3-B01, 3-B02, 4-B01 und 4-B02 erhalten einen Zehn-Minuten-Takt, während die Kreislinien 2-B01 und 2-B02 einen Takt von 20 Minuten erhalten. Die Taktabweichung bei den Kreislinien bietet zum einen Gelegenheit die Korrektheit des Optimierungsverfahrens unter diesen Bedingungen zu testen, zum anderen ist sie einer Entscheidung beim Entwurf des Optimierungsmodells geschuldet: Es wird nur genau eine Abfahrt pro Linie an einem Haltepunkt innerhalb des zugehörigen Taktintervalls unterstützt. Für Kreislinien bedeutet das, dass der erste Halt am Starthaltepunkt ebenfalls einem letzten Halt am Endhaltepunkt entsprechen muss. Andernfalls kommt es zu einer ungeplanten weiteren Abfahrt der gleichen Linie innerhalb des Taktintervalls: Der Abfahrt eines neu startenden Fahrzeugs, sowie der Ankunft eines seine Fahrt beendenden Fahrzeugs. Um dies zu verhindern, muss der Takt der betroffenen Kreislinie der benötigten Umlaufzeit, d.h. der Zeit um nach Abfahrt am Starthaltepunkt wieder dort anzukommen, entsprechen. Da Linien 2-B01 und 2-B02 eine Umlaufzeit von 20 Minuten haben, ergibt sich für sie so ein Takt von 20 Minuten.

Ohne verkehrsplanerische Vorgaben kann ein Fahrplan unter den gegebenen Bedingungen höchstens einen netzwerkweiten Sicherheitsabstand von  $\bar{\delta}(\lambda) = 3$  Minuten und einen Zielfunktionswert von 848 erreichen. Die Optimierungsanwendung generiert einen Fahrplan mit Zielfunktionswert 839 und einem netzwerkweiten Sicherheitsabstand von drei Minuten. Wie bei DCL 2014 erklärt sich der Unterschied von einem Prozent zwischen Zielfunktionswert und oberer Schranke durch die isolierte Betrachtung der Haltepunkte bei der Berechnung der oberen Schranken ihrer Sicherheitsabstände. Beispielsweise wird für Haltepunkt 1141 eine obere Schranke von fünf Minuten berechnet. Um aber das Systemoptimum zu erhalten, muss an Haltepunkt 1141 ein Sicherheitsabstand von vier Minuten verwendet werden.

Um ein realistischeres Anwendungsszenario zu betrachten, werden in einer weiteren Untersuchung beispielhaft die acht in Tabelle 3.4 abgebildeten Umsteigeverbindungen modelliert.

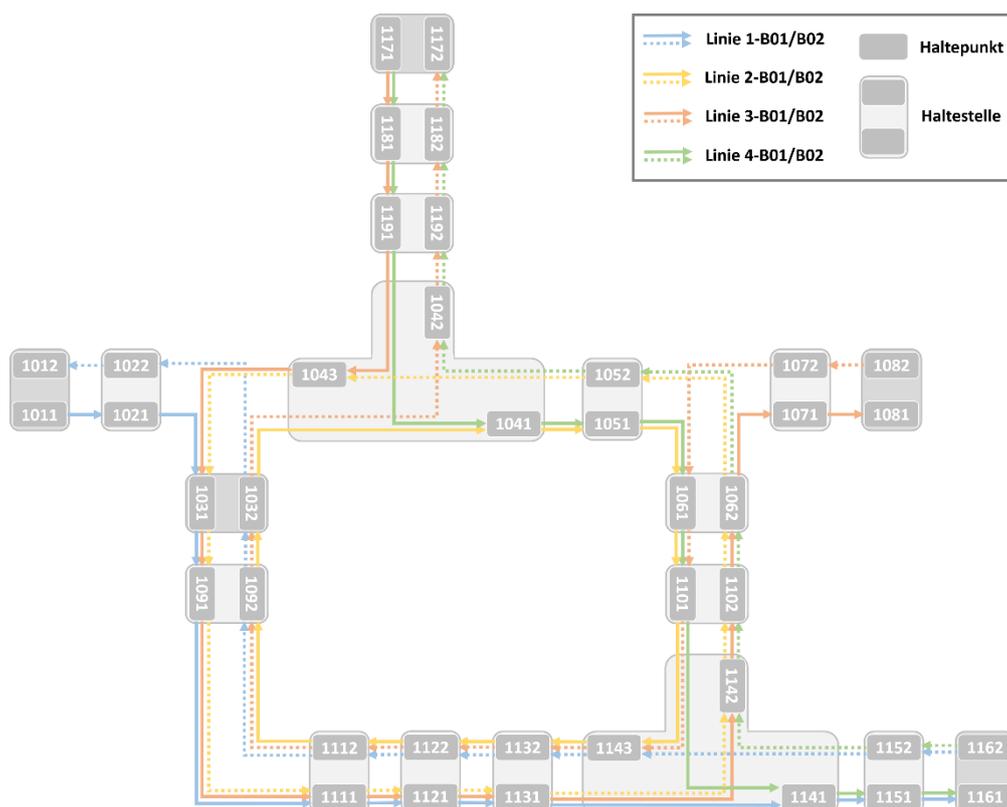


Abbildung 3.2: Beispielnetzwerk UCL 2015, dunkler unterlegte Haltestellen identifizieren Start-/Endhaltestellen der jeweiligen Linien

Nr.	Von	An	Zu	An	Abstands-Priorität									
					HIGH	MEDIUM	LOW	PROHIBITED	1	2	3	4	5	6
1	2-B01	1143	1-B01	1141	3	4	5	2	6	7	0	1	8	9
2	2-B01	1142	4-B01	1141	3	4	5	2	6	7	0	1	8	9
3	3-B01	1043	2-B01	1041	3	4	5	2	6	7	0	1	8	9
4	3-B02	1143	1-B01	1141	3	4	5	2	6	7	0	1	8	9
5	3-B02	1042	2-B01	1041	3	4	5	2	6	7	0	1	8	9
6	4-B01	1041	2-B02	1043	3	4	5	2	6	7	0	1	8	9
7	4-B02	1142	2-B01	1143	3	4	5	2	6	7	0	1	8	9
8	4-B02	1042	2-B02	1043	3	4	5	2	6	7	0	1	8	9

Tabelle 3.4: Beispielhafte Umsteigeverbindungen zwischen Linien an Haltepunkten, UCL 2015

Konkret sollen Umsteigeverbindungen modelliert werden bei denen die Wartezeit im Idealfall so gewählt ist, dass sowohl als zu lang empfundenes Warten als auch ein knappes Verpassen des Anschlusses vermieden wird. Aus diesem Grund werden Umsteigeverbindungen mit drei oder vier Minuten Wartezeit als sehr erwünscht definiert, gefolgt von Verbindungen mit einer Wartezeit von fünf Minuten. Um die Flexibilität des Optimierers zu erhöhen werden zusätzlich Umsteigeverbindungen von zwei Minuten sowie sechs bis sieben Minuten als noch tolerabel definiert. Abstände von null oder einer Minute sowie acht oder neun Minuten werden schließlich als unerwünscht markiert. Dadurch ergeben sich die 48 in Tabelle 3.4 angegebenen Umsteigevorgaben, jeweils sechs zusammenhängende für eine der acht Umsteigeverbindungen.

Mit einem Gewichtungsfaktor von  $\alpha = 0,5$  findet die Optimierungsanwendung einen Fahrplan mit Zielfunktionswert  $0,5 * 600 + 0,5 * 16 * 5,875 = 347$ . Wie Tabelle 3.5 zeigt, kann für fünf der acht gewünschten Umsteigeverbindungen der am stärksten priorisierte Abstand hergestellt werden. Lediglich die Umsteigeverbindungen von Linie 2-B02 zu 4-B01, 3-B02 zu 1-B01 und 4-B01 zu 2-B02 erhalten Abstände mit niedrigem Prioritätswert. Als Ausgleich für die Erfüllung der Vorgaben, büßt der Fahrplan an Serviceregularität ein und erreicht nur noch einen netzwerkweiten Sicherheitsabstand von zwei Minuten.

Nr.	Von	An	Zu	An	Abstand	Priorität
1	2-B01	1143	1-B01	1141	4	HIGH
2	2-B01	1142	4-B01	1141	2	LOW
3	3-B01	1043	2-B01	1041	3	HIGH
4	3-B02	1143	1-B01	1141	7	LOW
5	3-B02	1042	2-B01	1041	3	HIGH
6	4-B01	1041	2-B02	1043	2	LOW
7	4-B02	1142	2-B01	1143	3	HIGH
8	4-B02	1042	2-B02	1043	3	HIGH

*Tabelle 3.5: Generierte Abstände für Umsteigeverbindungen, UCL 2015*

### 3.4.2. Vergleich mit einer realen Testinstanz

Nachdem die Korrektheit des Optimierungsmodells nachgewiesen wurde, soll nun die Leistungsfähigkeit der Optimierungsanwendung genauer analysiert werden. Dazu werden Fahrpläne für das reale Busnetz der Stadt Köln aus dem Jahr 2001 (siehe Abbildung 3.3 bis Abbildung 3.6) generiert und die relevanten Kennzahlen mit denen der künstlichen Netze verglichen. Da keine Informationen über verkehrsplanerische Vorgaben und Linientakte für das Kölner Netz von 2001 vorliegen, werden im Weiteren Fahrpläne mit einheitlichem Zehn-Minuten-Takt und ohne Berücksichtigung verkehrsplanerischer Vorgaben generiert. Alle angegebenen Kennzahlen sind Durchschnittswerte über jeweils zehn Optimierungsläufe.

Wie Tabelle 3.7 und Tabelle 3.6 zeigen, sorgt die hohe Zahl an Haltepunkten, Linien und Linienkonflikten des Kölner Busnetzes für einen deutlichen Anstieg der Laufzeit, im Vergleich zu den beiden eher übersichtlichen künstlichen Netzen. Da die Optimierungsanwendung in ihrem Basismodus auch nach über sieben Tagen Rechenzeit nicht in der Lage war die Optimalität der berechneten Lösung sicherzustellen, musste für das Kölner Netz dabei ein anderes Abbruchkriterium verwendet werden als für die künstlichen Netze: Sobald der relative Abstand zwischen bester gefundener ganzzahliger Lösung und bester berechneter oberen Schranke 0,3 Prozent oder weniger betrug, wurde die Optimierung beendet und die bis dato beste gefundene Lösung zurückgegeben. Der Wert von 0,3 Prozent wurde ausgewählt, nachdem Beobachtungen des Optimierungsverlaufs gezeigt haben, dass an diesem Punkt in der Regel bereits eine Lösung mit bestmöglichem Zielfunktionswert gefunden wurde und nur noch Verbesserungen der Schranke stattfinden. Wie Tabelle 3.7 zeigt, konnte die Optimierungsanwendung in ihrem Basismodus mit dieser Abbruchbedingung nach ca. 2,5 Stunden einen Fahrplan für das Kölner Busnetz erzeugen.

Das Kölner Busnetz bietet daneben aber noch eine andere Möglichkeit: Es zerfällt in fünf unabhängige Zusammenhangskomponenten unterschiedlicher Größe. Dabei stellt das rechtsrheinische Netz (siehe Abbildung 3.5 und Abbildung 3.6) eine Komponente, während das linksrheinische Netz in vier Komponenten zerfällt: Das nördliche Netz um Chorweiler, Longerich, Ehrenfeld und Nippes (siehe Abbildung 3.3), das südliche Netz um Zollstock und Rodenkirchen (siehe Abbildung 3.4), sowie die Hin- und Rückrichtung der zentralen Linien 136 und 146 (siehe ebenfalls Abbildung 3.4).

Wird sich dieser Sachverhalt zu Nutze gemacht, indem das von Genç vorgestellte Zerlegungsverfahren verwendet wird, kann ein Optimalfahrplan für das Kölner Busnetz in ca. 30 Minuten erzeugt werden, was einer Reduktion um ca. 80,4 Prozent entspricht. Die Optimalität der so erhaltenen Lösung wurde sichergestellt, indem sie der Optimierungsanwendung wieder als Ausgangslösung für die Optimierung ohne Zerlegung übergeben wurde. Von dieser Startlösung ausgehend, werden keine weiteren Verbesserungen gefunden, die Lösungen mit Zerlegung entsprechen also dem Gesamtoptimum ohne Zerlegung.

Die Zeit die im Preprocessing verbracht wird, ist für alle Netze, unabhängig von der Zerlegung in Zusammenhangskomponenten, vernachlässigbar gering. Sie steigt proportional zur Zahl der Haltepunkte und Linien, da die Identifizierung der Linienkonflikte und Äquivalenzklassen zunehmend komplexer wird.

Netz	Preprocessing	CPLEX	Postprocessing	Gesamtlaufzeit	Gap [%]
<b>DCL 2014</b>	0,90	0,59	18,71	20,20	-
<b>UCL 2015</b>	0,90	0,78	22,24	23,92	-
<b>Köln 2001<sup>11</sup></b>	7,09	> 8.861,66	185,16	> 9.053,91	0,3

Tabelle 3.7: Durchschnittliche Laufzeit in Sekunden, ohne Zerlegung in Zusammenhangskomponenten

Netz	Preprocessing	CPLEX	Postprocessing	Gesamtlaufzeit	Gap [%]
<b>DCL 2014</b>	0,86	0,73	18,88	20,47	-
<b>Köln 2001</b>	7,36	1.577,68	186,49	1.771,53	-

Tabelle 3.6: Durchschnittliche Laufzeit in Sekunden, mit Zerlegung in Zusammenhangskomponenten

Die Zeit, die im Postprocessing verbracht wird steigt ebenfalls, unabhängig von der Zerlegung in Zusammenhangskomponenten, proportional mit der Zahl der im Netz verkehrenden Linien, wird insbesondere aber durch die Dauer der Planungsperiode beeinflusst. Je mehr Linien und je länger die Planungsperiode, desto mehr einzelne Fahrten müssen erstellt und mittels zeitaufwendiger Datenbankoperationen gespeichert werden. Für DCL 2014 und UCL 2015 ist die Zeit die für das Postprocessing benötigt wird besonders signifikant, sie macht dort 93% der Gesamtlaufzeit aus.

Die eigentliche Lösung des Optimierungsproblems nimmt für DCL 2014 und UCL 2015 keine signifikante Laufzeit in Anspruch, während sie für das Kölner Netz, wie bereits besprochen, deutlich steigt und durch die Zerlegung in Zusammenhangskomponenten stark reduziert werden kann.

Die Zerlegung in Zusammenhangskomponenten scheint dabei nur bei hinreichend komplexen Netzen in einen Laufzeitgewinn umgewandelt werden zu können, was sich darin zeigt, dass die Gesamtlaufzeit für DCL 2014 bei getrennter Optimierung der Zusammenhangskomponenten sogar um 1,3% steigt.

Abschließend soll die Korrektheit des für das Kölner Busnetz erstellten Optimalfahrplans überprüft werden. Unter den geschilderten Bedingungen kann er höchstens einen netzwerkweiten Sicherheitsabstand von zwei Minuten und einen Zielfunktionswert von 33.729 erreichen. Der Zielfunktionswert beinhaltet dabei einen Wert von 10.270 für die 1.072 Haltepunkte an denen lediglich eine Linie abfährt. Der von der Optimierungsanwendung mittels Zerlegung generierte Fahrplan hat einen Zielfunktionswert von 33.504 (und damit eine Abweichung von 0,67 Prozent von der berechneten oberen Schranke) sowie einen netzwerkweiten Sicherheitsabstand von zwei Minuten. Tabelle 3.8 vergleicht die nach Berechnung der oberen Schranken theoretisch maximal erreichbare Zahl an Haltepunkten mit einem spezifischen Sicherheitsabstand mit der unter dem berechneten Fahrplan tatsächlich erreichten Zahl. Wie erwartet muss zur Erreichung des Systemoptimums von den lokalen oberen Schranken der Sicherheitsabstände abgewichen werden. Allerdings kann dennoch an mindestens 116 (55%) der 210 Haltepunkte, die von mehr als einer Linie bedient werden, der theoretisch optimale Sicherheitsabstand erreicht werden.

Sicherheitsabstand	1	2	3	4	5	6	7	8	9	10
<b>Theoretisches Maximum</b>	0	13	19	0	178	0	0	0	0	1027
<b>Optimal-Fahrplan</b>	0	36	38	33	103	0	0	0	0	1027

Tabelle 3.8: Anzahl Haltepunkte, die einen spezifischen Sicherheitsabstand erreichen, Köln 2001

Tabelle 3.9 fasst die wichtigsten Kennzahlen für die verschiedenen Netze noch einmal zusammen.

Die Ergebnisse der Untersuchungen lassen darauf schließen, dass der beschriebene Optimierungsansatz in annehmbarer Laufzeit zulässige periodische Taktfahrpläne mit maximaler Serviceregularität und maximaler Erfüllung verkehrsplanerischer Vorgaben liefert. Dennoch kann die durch den Optimierer durchgeführte Bewertung der Fahrpläne nur ein erster Indikator für deren tatsächliche Anwendbarkeit sein. Während des Einsatzes im täglichen Betrieb sind Fahrpläne dynamischen und stochastischen Einflüssen ausgesetzt, die von den bisher beschriebenen Optimierungsmodellen nicht oder nur teilweise berücksichtigt werden. Diese Einflüsse haben aber signifikante Auswirkungen auf die Performanz des Fahrplans (und damit des Verkehrssystems). Eine Möglichkeit dynamische und stochastische Einflüsse

<sup>11</sup> Die Laufzeitwerte für das Kölner Netzwerk ohne Zerlegung geben Mindestlaufzeiten wieder, da die Optimierung frühzeitig beendet wurde.

des täglichen Betriebs abzubilden – und damit die Anwendbarkeit des Fahrplans unter dynamischen Bedingungen zu gewährleisten – besteht durch die Anwendung von Simulation.

Netz	Zielfunktionswert	Obere Schranke	$\delta(\lambda)$	$\bar{\delta}(\lambda)$	$\emptyset$ Gesamtlaufzeit (ohne Zerlegung) [Sek.]	$\emptyset$ Gesamtlaufzeit (mit Zerlegung) [Sek.]
DCL 2014	310	312	3	3	20,20	20,47
UCL 2015	839	848	3	3	23,92	-
Köln 2001	33.504	33.729	2	2	> 9.053,91	1.771,62

Tabelle 3.9: Kennzahlen für die Optimierung der Streckennetze

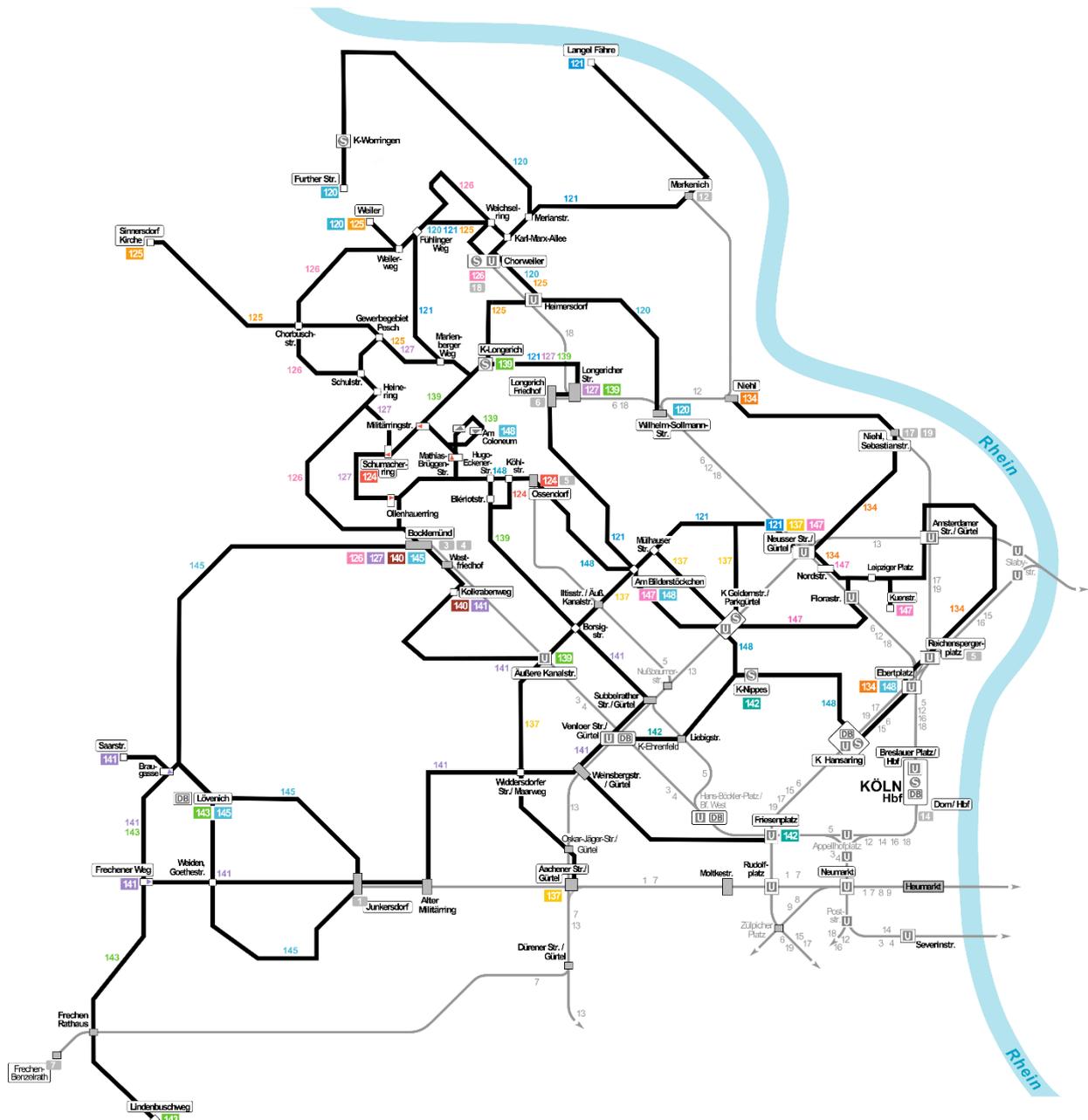


Abbildung 3.3: Nördliches, linksrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001

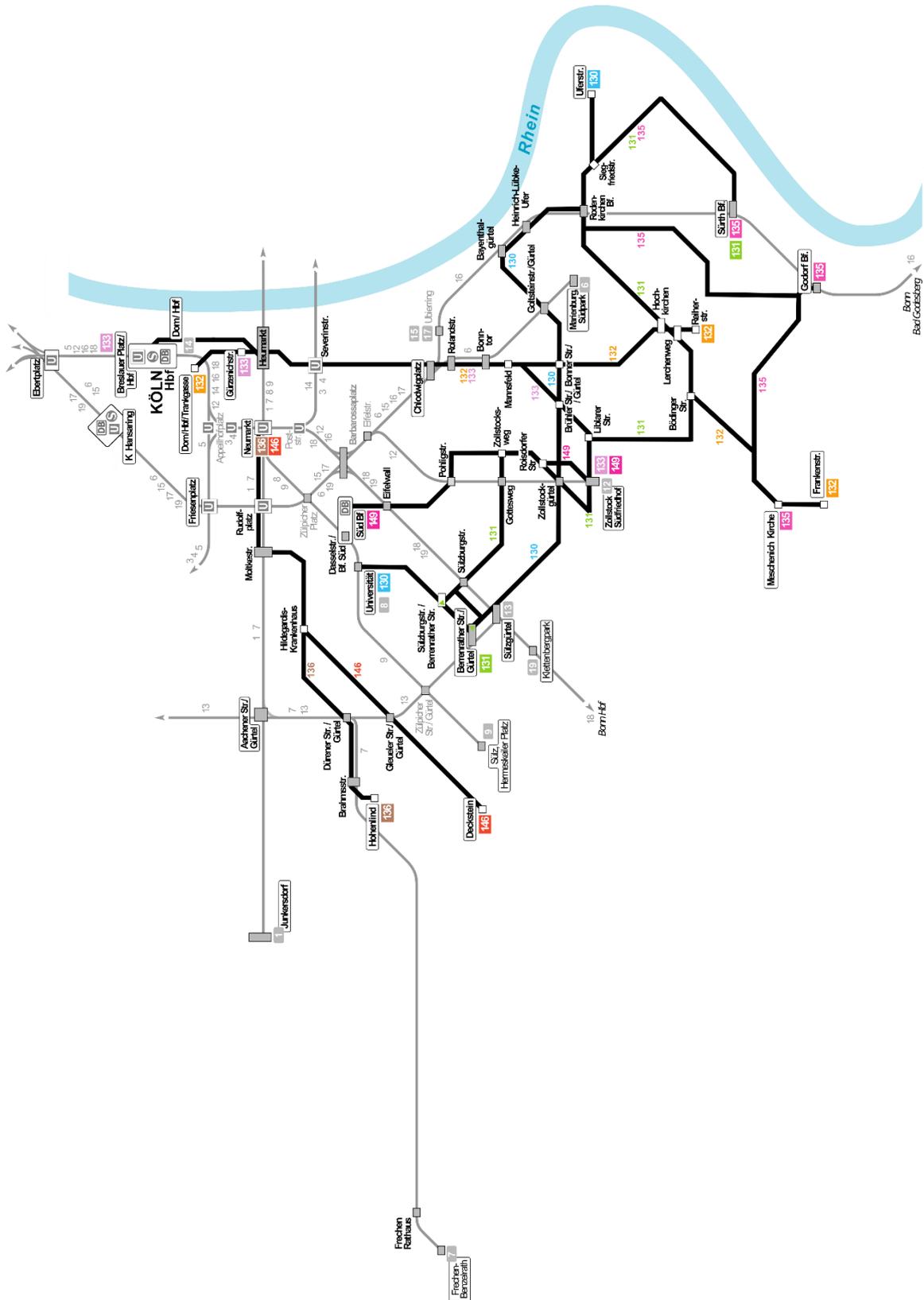


Abbildung 3.4: Südliches, linksrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001

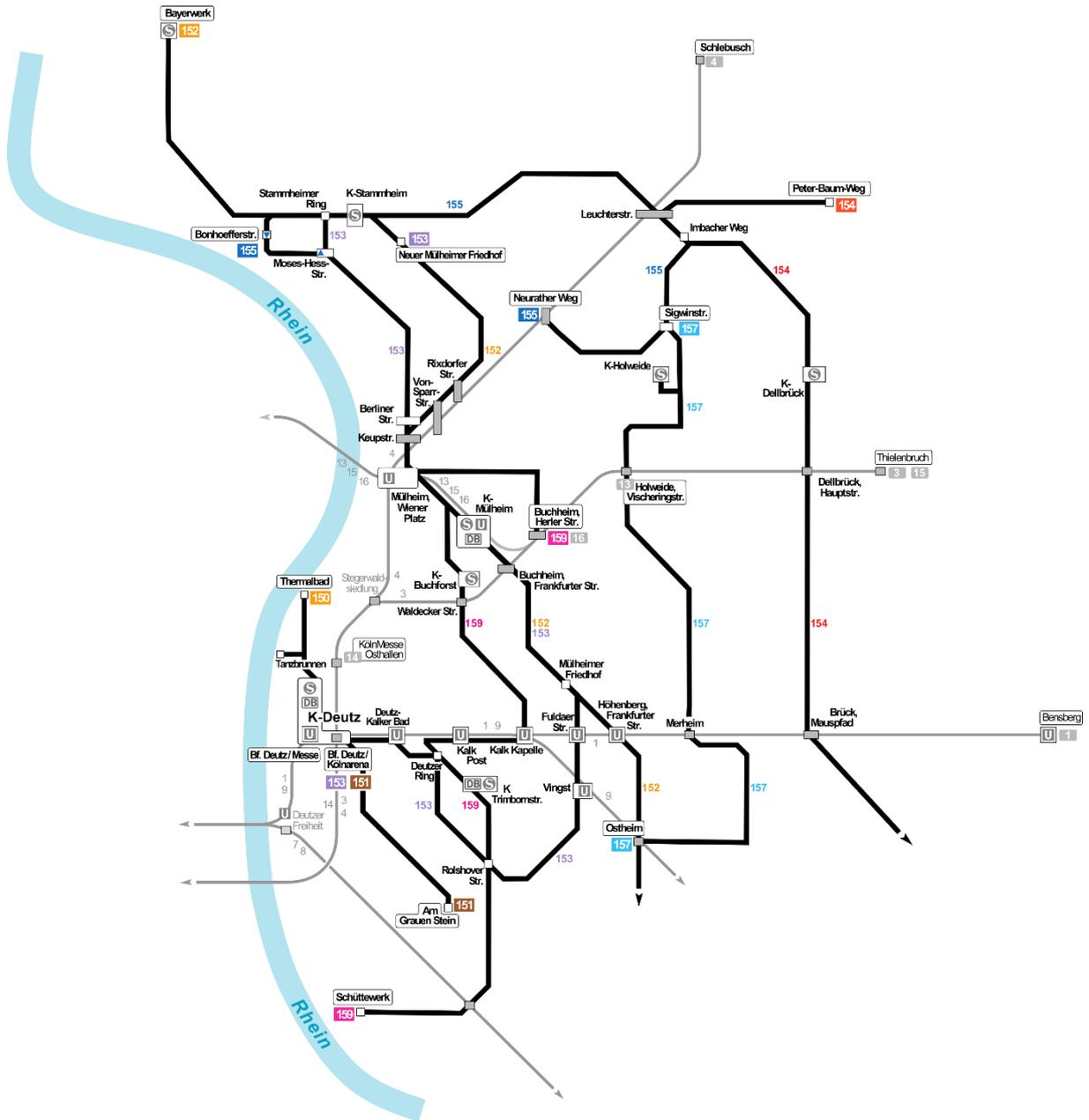


Abbildung 3.5: Nördliches, rechtsrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001

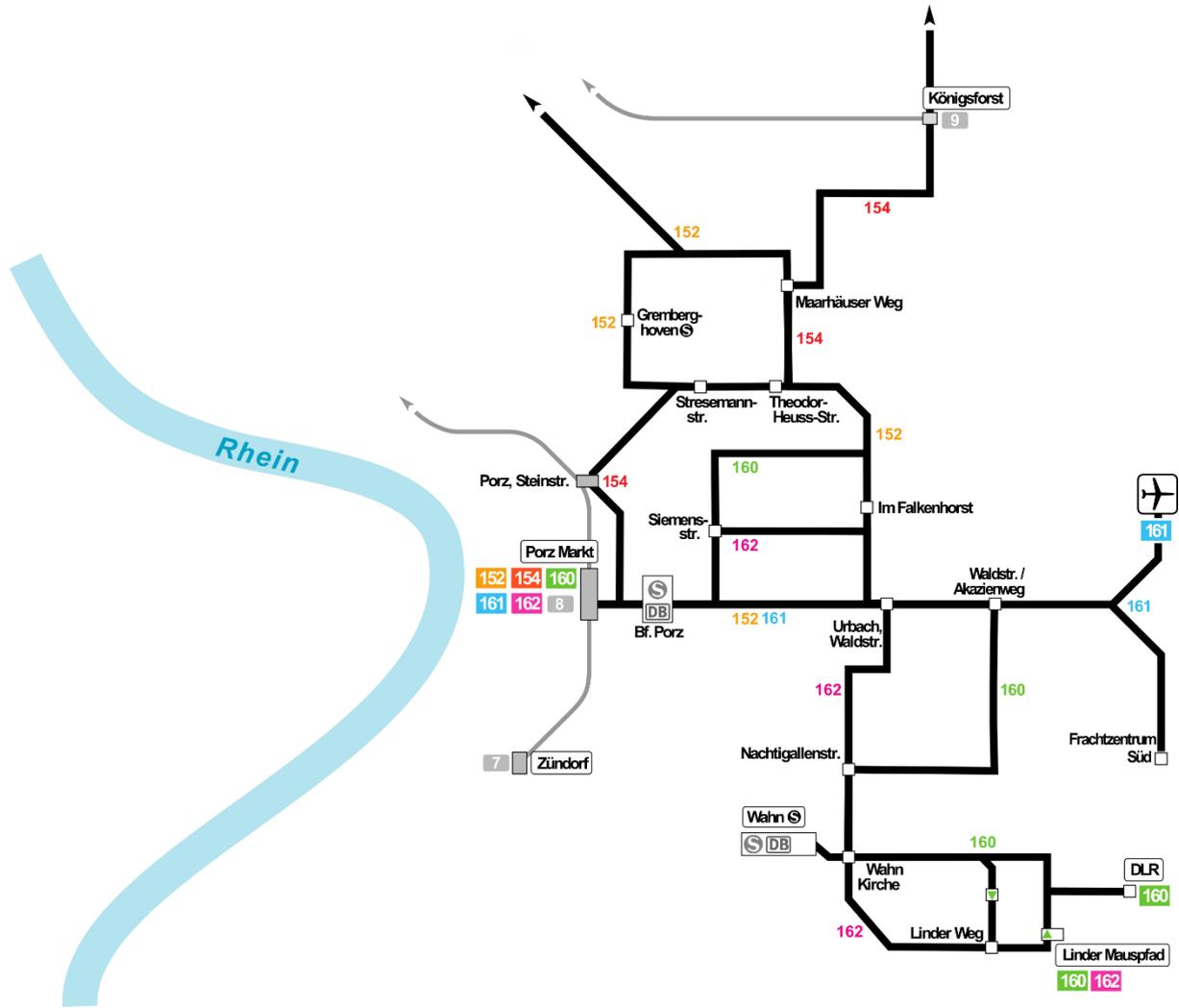


Abbildung 3.6: Südliches, rechtsrheinisches Kölner Bus- und Bahnnetz (in Schwarz bzw. Grau) aus dem Jahr 2001

## 4. Simulationsmodelle für Nahverkehrssysteme

### 4.1. Einführung und Abgrenzung

Der Begriff *Simulation* kann die unterschiedlichsten Verfahren und Modelle umfassen. In ihrer weitesten Definition bezeichnet *Simulation* schlicht die Durchführung von Experimenten an einem Modell, um Aussagen über das Verhalten eines realen Systems oder Prozesses zu tätigen (vgl. [66], S. 7).

Eine engere Definition von Simulation, die leicht adaptiert auch für diese Arbeit verwendet wird, formuliert Ingalls in [47]: Simulation ist die Durchführung von Experimenten an einem *dynamischen* und *stochastischen* Modell, mit dem Ziel Aussagen über das Verhalten eines realen *dynamischen* und *stochastischen* Systems oder Prozesses zu treffen. Dynamisch bedeutet in diesem Kontext, dass sich das Verhalten des Systems – und damit des Modells – über die Zeit verändert, während stochastisch bedeutet, dass das Verhalten von zufallsabhängigen Größen beeinflusst sein kann.

Damit können Simulationsmodelle von (mathematisch-)analytischen Modellen, wie bspw. die Optimierungsmodelle aus den vorhergehenden Abschnitten, abgegrenzt werden. Analytische Modelle versuchen Zusammenhänge realer Systeme als geschlossene mathematische Ausdrücke zu formulieren. Diese Formulierung bedingt dabei in der Regel die Festlegung starrer Vorgaben, z.B. feste Fahrzeiten zwischen Haltepunkten, um handhabbare Modelle zu erhalten. Erfüllt das betrachtete System die Vorgaben allerdings, so führt die Anwendung des analytischen Modells stets zu einer gültigen Lösung. Bei Simulationsmodellen wird hingegen versucht das Verhalten des Realsystems als komplexe Entscheidungsprozesse zu modellieren. Dadurch kann zwar unter Umständen nicht mehr gewährleistet werden, dass die erhaltene Lösung ein Optimum darstellt, allerdings gewinnt man Freiheit bei der Modellierung, wodurch auch dynamisches und stochastisches Verhalten abgebildet werden kann (vgl. für diesen Absatz [57], S. 5ff.).

Zur Gruppe dynamischer und stochastischer Systeme gehören insbesondere auch Verkehrssysteme, weswegen Simulationsmodelle ein unverzichtbares Untersuchungsinstrument für sie darstellen. Typischerweise werden zur Abbildung von Verkehrssystemen diskrete Simulationsmethoden angewendet, wie z.B. Zelluläre Automaten, ereignisbasierte oder agentenbasierte Simulation. Sie unterscheiden sich sowohl bei der Modellierung von Systembestandteilen- und zusammenhängen als auch bei der Modellierung allgemeiner Konzepte, wie bspw. des Zeitfortschritts. Entsprechend muss im Einzelfall von der ModelliererIn entschieden werden welche Systembestandteile- und zusammenhängen zur Abbildung des interessierenden Verhaltens kritisch sind und welche Modellierungsmethoden in Frage kommen (vgl. [57], S. 118). Bevor konkrete Modelle zur Simulation von Verkehrssystemen diskutiert werden, sollen deswegen zunächst wichtige Konzepte und häufig verwendete Simulationsmethoden besprochen werden.

#### 4.1.1. Mechanismen zur Abbildung von Zeit

Aus der Definition von Simulation ergibt sich, dass die Abbildung von Zeit und Zeitfortschritt ein Kernelement jedes Simulationsmodells ist. Dabei muss zwischen *Simulationszeit* (engl. *simulation time*) und *Wanduhrzeit* (engl. *wallclock time*) unterschieden werden. Während Simulationszeit die Zeit beschreibt, die während eines Simulationslaufs aus Sicht der simulierten Komponenten vergeht (und damit sowohl einen Zeitpunkt als auch eine Zeitdauer beschreiben kann), wird mit Wanduhrzeit die real vergehende Uhrzeit bezeichnet (vgl. für diesen und die folgenden Absätze [38], S. 27ff. sowie [57], S. 90ff.).

Das Fortschreiten der Simulationszeit kann in direkter Beziehung zum Fortschritt der Wanduhrzeit stehen. Vergeht die Simulationszeit in der gleichen Geschwindigkeit wie die Wanduhrzeit spricht man von *Echtzeit-Ausführung* (engl. *real-time execution*). Vergeht die Simulationszeit hingegen um einen definierten Faktor schneller oder langsamer als die Wanduhrzeit spricht man von *skalierter Echtzeit-Ausführung* (engl. *scaled real-time execution*). In Abgrenzung dazu kann der Fortschritt der Simulationszeit auch unabhängig vom Fortschritt der Wanduhrzeit erfolgen. In diesem Fall kann eine Zeiteinheit der Simulationszeit mal wenige (Milli-)Sekunden Wanduhrzeit veranschlagen, mal kann der gleiche Fortschritt in der Simulationszeit mehrere Minuten Wanduhrzeit benötigen. Man spricht dann davon, dass die Simulation *so schnell wie möglich ausgeführt* wird (engl. *as-fast-as-possible execution*).

Zusätzlich zur Beziehung zwischen Simulationszeit und Wanduhrzeit, muss bei der Aufstellung eines Simulationsmodells ebenfalls entschieden werden welche *zeitliche Auflösung* (engl. *time flow mechanism*) die Simulationszeit besitzt, d.h. wie sich der Modellzustand über die Simulationszeit verändert.

Es kann grob zwischen *diskreter* und *kontinuierlicher* Zeitauflösung unterschieden werden. Bei einem Modell mit kontinuierlicher Zeitauflösung ändert sich der Modellzustand stetig über die Simulationszeit. Das Verhalten des Systems wird dabei typischerweise als Menge von Differenzialgleichungen modelliert, die den Modellzustand in Abhängigkeit der Simulationszeit beschreiben.

Bei diskreter Zeitauflösung ändert sich der Modellzustand hingegen zu diskreten Simulationszeitpunkten, d.h. das Modell „springt“ von einem Zustand in den nächsten. Die Größe der Zeitsprünge kann fix oder variabel sein. Bei *fixem Zeitfortschritt* (engl. *time stepped execution*) schreitet die Simulationszeit in Inkrementen fester Größe voran und die Modellkomponenten können ihren Zustand zu jedem der daraus folgenden Zeitpunkte verändern. Zustandsänderungen die im gleichen Zeitschritt auftreten werden als simultan und voneinander unabhängig betrachtet, da ein Zeitschritt die kleinste Einheit ist nach der die Simulation differenzieren kann und innerhalb eines Zeitabschnitts keine weiteren Aussagen über die Reihenfolge der Zustandsveränderungen getroffen werden können. Stehen Zustandsänderungen eines Zeitschritts in kausaler Beziehung zueinander, müssen sie stattdessen in unterschiedlichen Zeitschritten simuliert werden. Die durch die ModelliererIn gewählte Größe der Zeitinkremente ist folglich eine wichtige Modelleigenschaft, da sie die zeitliche Präzision der Simulation definiert. Diese Problematik entfällt bei diskreten Simulationsmodellen mit *variablem Zeitfortschritt* (engl. *event-driven execution*). Hier finden Zustandsveränderungen zu unregelmäßigen Zeitpunkten statt, die Simulationszeit schreitet vom Zeitpunkt einer Zustandsänderung zum Zeitpunkt der nächsten voran. Die Diskretisierung der Zeit geht also implizit vom System aus und wird nicht mehr durch die ModelliererIn vorgegeben.

Im Rahmen dieser Arbeit werden vornehmlich Simulationsmodelle mit diskreter Zeitauflösung betrachtet, die so schnell wie möglich ausgeführt werden. Wie Liebl in [57] (S. 10) allerdings anmerkt, ist eine scharfe Trennung zwischen den verschiedenen Ansätzen unter Umständen kompliziert. Auf der einen Seite können kontinuierliche Systeme mit diskreten Simulationsmodellen angenähert werden. Auf der anderen Seite können Systeme kontinuierliche Komponenten beinhalten, die aber für die zu beantwortende Forschungsfrage keine Relevanz aufweisen, also nicht modelliert werden müssen.

## 4.1.2. Allgemeine Simulationsmethoden

### 4.1.2.1. Zelluläre Automaten

Besteht das abzubildende System aus einer Menge gleichförmiger Einheiten, die in einer (geografischen) Nachbarschaftsrelation zueinander stehen, können Zelluläre Automaten zur Modellierung angewendet werden.

Ein Zellulärer Automat  $Z = (\tilde{L}, \tilde{S}, N, \tilde{f})$  ist ein (unendliches) reguläres Gitter  $\tilde{L}$  von Zellen  $z \in \tilde{L}$ , von denen sich jede in einem Zustand aus der endlichen Menge von Zuständen  $\tilde{S}$  befinden kann. Jede Zelle  $z$  bildet mit einer endlichen Menge von Zellen  $N(z) = \{i \in \tilde{L} \mid z - i \in N\}$  eine Nachbarschaft, deren Ausgestaltung durch die Menge der zulässigen Nachbarschaftsindizes  $N$  bestimmt wird, die wiederum vornehmlich von der räumlichen Diskretisierung des Modells abhängt. Diese *Gittergeometrie* kann prinzipiell eine beliebige dimensionale Ausdehnung haben; in praktischen Anwendungen finden sich allerdings typischerweise ein-, zwei- oder dreidimensionale Gitter. Teil der Gittergeometrie ist auch die Form der Zellen. Jede Zellform, die die Regularität des Gitters gewährleistet ist denkbar; häufig anzutreffende Zellformen sind: quadratisch, dreieckig oder hexagonal (vergleiche zu diesem und den folgenden Absätzen [102], S. 1ff.).

Abhängig vom Untersuchungsziel, muss die Nachbarschaft nicht zwangsläufig (nur) direkte Nachbarn – d.h. Zellen mit Entfernung 1 – umfassen, sondern kann auch Zellen beinhalten die weiter entfernt liegen oder gänzlich unregelmäßig sein. Dies kann beispielsweise sinnvoll sein, wenn – wie bei der Modellierung von Nahverkehr – auch weiter entfernt liegende Hindernisse beachtet werden müssen. Abbildung 4.1 zeigt die häufig anzutreffenden von-Neumann- und Moore-Nachbarschaften im Kontext verschiedener Gittergeometrien. Wie zu sehen, gehören bei der von-Neumann-Nachbarschaft nur die Zellen zur Nachbarschaft einer Zelle  $z$  die sich eine ganze Kante des Gitters mit  $z$  teilen. Die Moore-

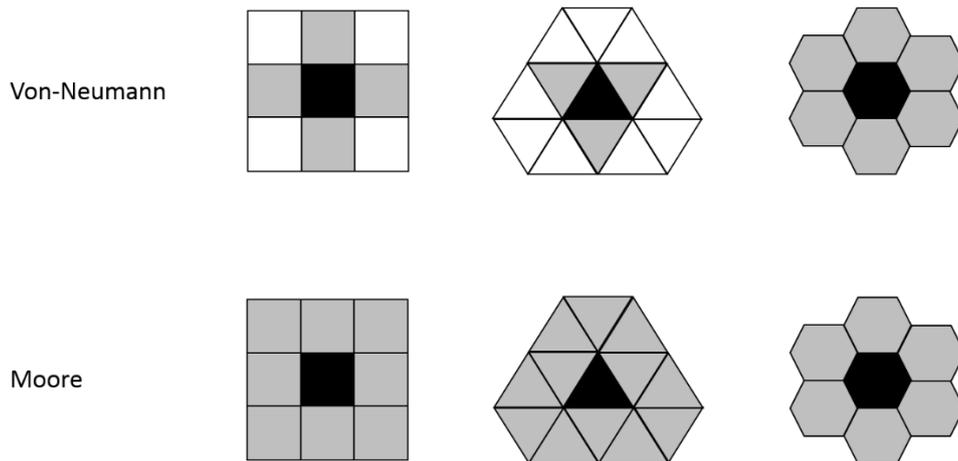


Abbildung 4.1: Von-Neumann- und Moore-Nachbarschaft (jeweils in Grau) in unterschiedlichen Gittergeometrien

Nachbarschaft umfasst hingegen zusätzlich auch alle Zellen die lediglich einen Eckpunkt mit  $z$  gemeinsam haben.

Mangels Rechen- und Speicherkapazität wird die maximale Ausdehnung des Gitters für praktische Anwendungen in der Regel begrenzt, wodurch eine spezifische Betrachtung der Randzellen notwendig wird. Üblicherweise werden drei verschiedene Begrenzungsarten unterschieden: Bei *periodischen Rändern* sind gegenüberliegende Seiten des Gitters benachbart. Bei *reflektierenden Rändern* spiegeln die Randzellen hingegen den Zustand der ihnen gegenüberliegenden inneren Zelle. Bei *Rändern mit fester Zellbelegung*, wird den Randzellen stattdessen ein fester Zustand zugewiesen, der sich über den Verlauf der Simulation nicht ändert. Die Gitterbegrenzung muss dabei nicht zwingend aus einer einzelnen Randart bestehen, sondern kann auch verschiedene Begrenzungsarten miteinander kombinieren, wenn dadurch das interessierende Systemverhalten besser abgebildet wird.

Gemeinsam mit der Gittergeometrie müssen Entscheidungen bezüglich des Zeitfortschritts und der Transitionsfunktion des Zellulären Automaten getroffen werden. Bei Zellulären Automaten schreitet die Simulationszeit in fixen Inkrementen zwischen diskreten Zeitpunkten voran. Zu jedem Simulationszeitpunkt  $t$  wird mit der Konfiguration  $\tilde{C}_t: \tilde{L} \rightarrow \tilde{S}$  jeder Zelle des Gitters ihr aktueller Zustand zugeordnet. Die Konfiguration  $\tilde{C}_0$  bezeichnet die durch die ModelliererIn vorgegebene Startkonfiguration des Zellulären Automaten. Zum Fortschreiben der Simulationszeit wird, wie in Gleichung (4.1) abgebildet, durch die Transitionsfunktion  $\tilde{f}: \tilde{S}^{|\tilde{N}|} \rightarrow \tilde{S}$  die Konfiguration  $\tilde{C}_t$  in die Konfiguration  $\tilde{C}_{t+1}$  überführt.

$$\tilde{C}_{t+1}(z) = \tilde{f}(\{\tilde{C}_t(i) | i \in N(z)\}) \quad \forall z \in \tilde{L} \quad (4.1)$$

Die Transitionsfunktion ist die Modellgröße, die maßgeblich das Verhalten des Zellulären Automaten bestimmt. Sie ist von der Gittergeometrie, der Art und Größe der Nachbarschaft sowie der Menge der möglichen Zustände abhängig und ihre konkrete Ausgestaltung muss zusammen mit diesen Modellkomponenten durch die ModelliererIn anhand des Untersuchungsziels festgelegt werden. Es werden zwei Arten von Transitionsfunktionen unterschieden: Bei einer *deterministischen* Transitionsfunktion werden Zustandsübergänge eindeutig durch den Zustand der betroffenen Zelle sowie den Zuständen ihrer Nachbarn bestimmt. Bei *probabilistischen* Transitionsfunktionen hingegen wird der anzuwendende Zustandsübergang in jedem Simulationszeitschritt zufällig aus einer Menge möglicher Zustandsübergänge ausgewählt.

Aufgrund des Zusammenhangs zwischen Gittergeometrie, Nachbarschaft und Transitionsfunktion stehen räumliche und zeitliche Diskretisierung bei Zellularautomaten in direktem Zusammenhang, sodass die Festlegung der Zellabmessungen direkten Einfluss auf die Größe der Zeitinkremente hat. Wird beispielsweise wie beim von Joisten in [49] vorgestellten Simulationsmodell für Stadtbahnssystemen eine Zelllänge von 30 Metern – etwa einer Wagenlänge – gewählt, muss ein sich bewegendes Fahrzeug innerhalb eines Simulationsschritts stets ein Vielfaches dieser 30 Meter zurücklegen. Als Folge dessen

entspricht ein Simulationszeitschritt im Modell von Joisten vier Sekunden im Realsystem. Diese Problematik entfällt bei Verfahren der ereignisbasierten und agentenbasierten Simulation.

#### 4.1.2.2. Ereignisbasierte Simulation

Ändert sich der Zustand des abzubildenden Systems nur zu diskreten Zeitpunkten und kann es als Menge von stochastisch beeinflussten, verknüpften Prozessen beschrieben werden, eignet sich die ereignisbasierte Simulation besonders gut.

Das System wird hier als Menge von miteinander in Verbindung stehenden Entitäten mit Attributen sowie Ereignissen und Aktivitäten abgebildet. Eine *Entität* ist eine Komponente des zu untersuchenden Systems, die zur Abbildung des für das Untersuchungsziel relevanten Systemverhaltens explizit modelliert werden muss. Ein *Ereignis* ist ein unmittelbar stattfindender Vorfall, also ein Zeitpunkt zu dem eine Zustandsänderung des Modells eintreten kann. Im Gegensatz dazu bezeichnet eine *Aktivität* eine Aktion, also einen Zeitraum an dessen Ende eine Änderung des Modellzustands eintreten kann. Der Modellzustand wird dabei durch die Ausprägungen der relevanten Eigenschaften – der *Attribute* – aller Entitäten definiert (vergleiche zu diesem und den folgenden Absätzen [7], S. 30ff. sowie S. 106ff.).

Aus diesem Sachverhalt folgt, dass Ereignisse und Aktivitäten immer in einer einrahmenden Beziehung zueinander stehen (siehe Abbildung 4.2). Wie genau einzelne Ereignisse und Aktivitäten in Beziehung stehen muss dabei durch die ModelliererIn anhand des Untersuchungsziels definiert werden. Dazu gehört insbesondere auch die Spezifizierung der Dauer der einzelnen Aktivitäten. Sie kann sowohl deterministisch sein als auch stochastisch anhand von Verteilungsfunktionen modelliert werden. Unabhängig davon kann die konkrete Dauer einer Aktivität vollständig bestimmt werden sobald das mit dem Start der Aktivität korrespondierende Ereignis eintritt, da der Modellzustand per Definition zwischen zwei Ereignissen konstant bleibt, eine laufende Aktivität im Allgemeinen also nicht durch den Eintritt eines Ereignisses beeinflusst werden kann.<sup>12</sup>

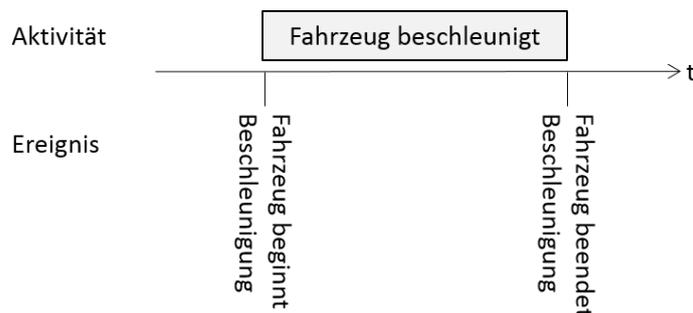


Abbildung 4.2: Beziehung zwischen Ereignis und Aktivität

Beim ereignisbasierten Simulationsansatz kann damit eine diskrete zeitliche Auflösung mit variablem Zeitfortschritt zur Fortschreibung der Simulationszeit verwendet werden. Der Ablauf der Simulation entspricht dann einer Aneinanderreihung von Momentaufnahmen des Modellzustands, von denen jede dem Eintritt eines Ereignisses entspricht.

Die Momentaufnahme des Simulationszeitpunkts  $t = 0$  nimmt, wie bei Zellulären Automaten, einen Sonderstatus ein. Sie muss durch die ModelliererIn festgelegt werden und definiert die initialen Zustände der Entitäten. Sie hat außerdem die Terminierung sogenannter *exogener* Ereignisse zur Folge. Hierbei handelt es sich um Ereignisse mit Ursprung „außerhalb“ des modellierten Systems, die dieses beeinflussen, beispielsweise die erste Ankunft einer Kundin in einem System. Abzugrenzen von den exogenen Ereignissen sind *endogene* Ereignisse und Aktivitäten, deren Ursprünge innerhalb des modellierten (Teil-)Systems liegen.

Zum Fortschreiben der Simulationszeit und zur Sicherung der zeitlich korrekten Ereignisreihenfolge wird eine Prioritätsschlange verwendet, die in diesem Kontext auch als *Future-Event-List (FEL)* be-

<sup>12</sup> Von dieser Regel kann im Einzelfall abgewichen werden, falls das durch die ModelliererIn erstellte Modell bzw. das von ihr zur Modellierung verwendete Simulationsframework Verfahren und Datenstrukturen zum Abbruch und zur Neuterminierung von Ereignissen/Aktivitäten zur Verfügung stellt.

zeichnet wird. Sie hält alle für zukünftige Simulationszeitpunkte terminierten Ereignisse in chronologisch aufsteigender Sortierung. Da sich der Inhalt der FEL permanent verändert während die Simulationszeit voranschreitet, hat ihre effiziente Verwaltung signifikanten Einfluss auf die Leistungsfähigkeit der späteren Implementierung.

Ein Simulationsschritt einer ereignisbasierten Simulation verläuft damit wie folgt: Entnahme des zeitlich nächsten Ereignisses aus der FEL, Aktualisierung der Simulationszeit auf den Eintrittszeitpunkt des Ereignisses, sowie Abarbeitung der unter Umständen durch das Ereignis ausgelösten Aktivität(en) samt Terminierung zugehöriger Folgeereignisse. Dieser Vorgang wird durch die Initialisierungsphase der Simulation gestartet und wiederholt sich so lange bis ein durch die ModelliererIn definiertes Abbruchkriterium erreicht wurde, beispielsweise wenn die FEL leer ist, eine anfangs spezifizierte maximale Laufzeit erreicht wurde oder ein zu Beginn terminiertes Stopp-Ereignis eintritt. Abbildung 4.3 verdeutlicht diesen Zusammenhang schematisch.

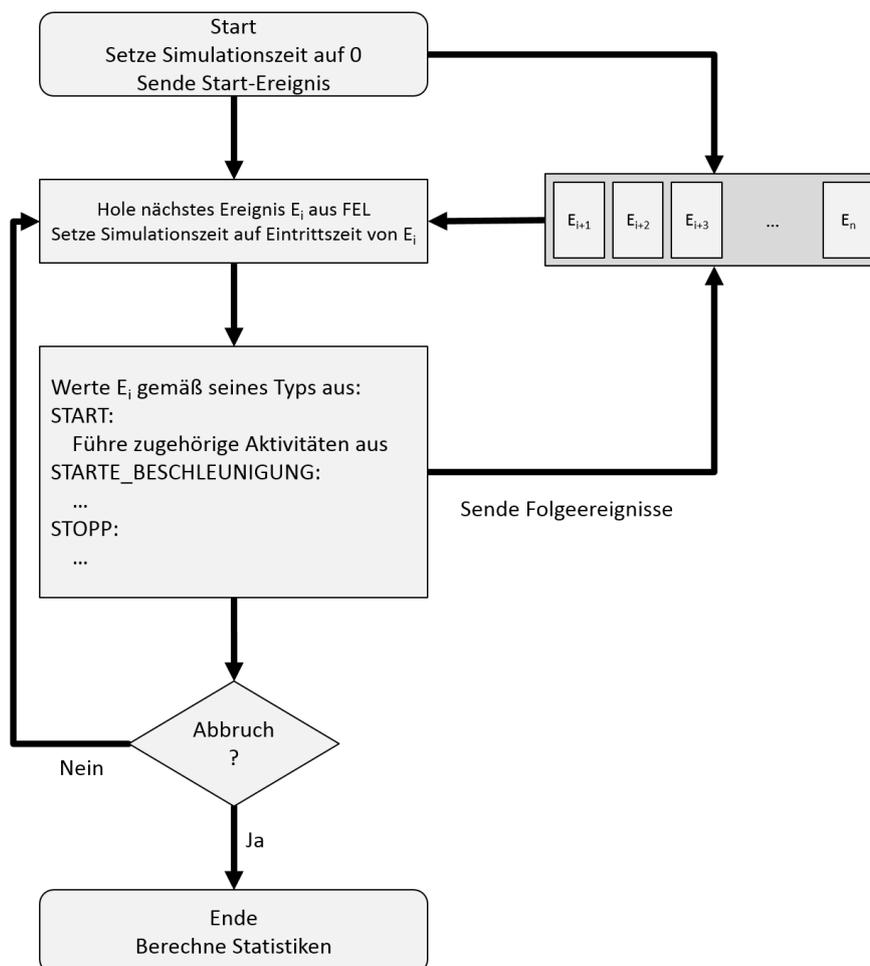


Abbildung 4.3: Schematische Darstellung des Ablaufs einer ereignisbasierten Simulation

Bei der ereignisbasierten Simulation müssen – anders als die Bezeichnung vermuten lässt – nicht zwingend die Ereignisse im Mittelpunkt der Modellierung stehen. Dies ist nur eine von drei verschiedenen Sichtweisen auf die ereignisbasierte Simulation, der so genannte *event-scheduling* Ansatz.

Im Gegensatz dazu stehen beim *process-interaction* Ansatz ganze Prozesse im Vordergrund der Modellierung. Ein *Prozess* fasst dabei eine feste Sequenz von chronologisch aufeinander folgenden Ereignissen und Aktivitäten zusammen. Der daraus folgende Ansatz eignet sich, wenn Entitäten das zu modellierende System in ähnlicher Weise „durchwandern“, d.h. wenn *transiente* Entitäten durch eine mehr oder weniger ähnliche Folge von Aktivitäten mit daraus folgenden Zustandsänderungen beeinflusst werden. Basis des *process-interaction* Ansatz ist eine Future-Event-List mit terminierten Ereignissen, sowie eine oder mehrere Warteschlangen für Entitäten deren nächste Aktivität bereits hätte starten sollen aber verzögert ist, da eine notwendige Eintrittsbedingung – etwa ausreichend Kapazität auf einer Maschine

– noch nicht erfüllt ist. Wie beim event-scheduling Ansatz wird in jedem Simulationsschritt das zeitlich nächste Ereignis aus der FEL genommen, die Simulationszeit auf den Eintrittszeitpunkt des Ereignisses gesetzt und versucht die zugehörigen Aktivitäten auszuführen. Sind die notwendigen Eintrittsbedingungen noch nicht erfüllt, wird die betroffene Entität in die entsprechende Warteschlange verschoben. Zusätzlich wird für alle bereits wartenden Entitäten geprüft, ob die Eintrittsbedingungen für ihre nächste Aktivität erfüllt sind und der zugehörige Prozess fortgesetzt werden kann.

Beim *activity-scanning* Ansatz liegt der Fokus der Modellierung auf den Aktivitäten und den zu ihrer Aktivierung und Beendigung notwendigen Bedingungen. Anders als beim event-scheduling und process-interaction Ansatz wird die Simulationszeit hier in fixen Zeitinkrementen fortgeschrieben. Es wird also keine FEL verwendet, sondern es werden nur Warteschlangen von aktuell stattfindenden Aktivitäten gehalten. In jedem Zeitschritt wird dann für alle diese Aktivitäten – und potentiell neu zu startenden Aktivitäten – geprüft ob ihre Start- bzw. Endbedingungen erfüllt sind und gegebenenfalls der Modellzustand aktualisiert.

Wie Liebl in [57] (S. 101) schreibt, beschreibt der event-scheduling Ansatz also eine Vorwärtsverkettung zwischen Ereignissen und Aktivitäten, indem nach Eintritt jedes Ereignis geprüft wird welche Aktivitäten und Folgeereignisse stattfinden können, während der activity-scanning Ansatz eine Rückwärtsverkettung beschreibt, da geprüft wird unter welchen Bedingungen eine Aktivität eintreten kann.

Verfechter des activity-scanning Ansatzes führen an, dass der event-scheduling Ansatz umso komplexer wird, je umfangreicher potentielle Konstellationen von Modellzuständen ausfallen. In diesem Fall muss zu jedem Ereigniseintritt eine große Zahl an Bedingungen überprüft werden, um zu entscheiden ob – und welche – Folgeereignisse ausgelöst werden. Im Gegensatz dazu werden beim activity-scanning Ansatz nur die direkt aus einer Aktivität resultierenden Zustandsänderungen durchgeführt, eine Terminierung von zukünftigen Ereignissen findet hingegen nicht statt. Der sich aus diesem Ansatz ergebenden Übersichtlichkeit bei der Modellierung – und Abarbeitung – von Aktivitäten, steht allerdings die höhere Komplexität bei der Auswahl der zu startenden oder zu beendenden Aktivitäten gegenüber. Im Gegensatz zum event-scheduling Ansatz muss beim activity-scanning Ansatz in jedem Zeitschritt überprüft werden welche Zustandsänderungen zulässig sind (vergleiche zu diesem Absatz [57], S. 101ff.).

Um den mit dem permanenten Scannen nach zu startenden oder zu beendenden Aktivitäten verbundenen Rechenaufwand zu reduzieren, wurde eine Erweiterung des activity-scanning approach entworfen, der *three-phase* Ansatz. Er kombiniert Eigenschaften des event-scheduling und des activity-scanning Ansatzes, um die Verwendung von variablen Zeitinkrementen zu ermöglichen und ähnelt damit dem process-interaction Ansatz. Es wird eine Future-Event-List verwendet um Ereignisse zu verwalten, die als Aktivitäten ohne zeitliche Dauer angesehen werden. Darüber hinaus werden Aktivitäten in zwei Klassen eingeteilt: Die *B-Aktivitäten* sind unbedingte („bound to happen“) Ereignisse und Aktivitäten, die ohne weiteren Test von Bedingungen eintreten können. *C-Aktivitäten* hingegen sind bedingte („happen conditionally“) Aktivitäten, für die Start- und/oder Endbedingungen spezifiziert sein müssen. Die drei Phasen des three-phase Ansatz sind dann (vergleiche für diesen und den folgenden Absatz [7], S. 115 sowie [57], S. 105-106):

- A. Entferne das zeitlich nächste Ereignis aus der FEL und setze die Simulationszeit auf die Eintrittszeit des Ereignisses.
- B. Führe alle B-Aktivitäten aus, die für den aktuellen Simulationszeitpunkt terminiert worden sind.
- C. Teste für alle C-Aktivitäten, ob ihre Start- bzw. Endbedingungen erfüllt sind. Wiederhole diesen Schritt so lange, bis keine neuen C-Aktivitäten und Ereignisse mehr eintreten.

Aufgrund der starken Ähnlichkeiten zwischen den unterschiedlichen Ansätzen und ihrer Überführbarkeit ineinander, ist die Wahl einer Sichtweise bei der Abbildung des zu untersuchenden Systems hochgradig von den Präferenzen der ModelliererIn abhängig.

### 4.1.2.3. Agentenbasierte Simulation

Die agentenbasierte Simulation ist immer dann ein geeignetes Mittel, wenn das abzubildende System aus einer (heterogenen) Menge von individuellen, autonomen und miteinander interagierenden Handelnden besteht und wird häufig zur Abbildung von sozialem Verhalten verwendet (vergleiche für diesen und die folgenden Absätze [64]).

Ein agentenbasiertes Simulationsmodell besteht aus zwei Teilen: Den Agenten mit ihren Attributen, Verhaltensregeln und Interaktionsprinzipien, sowie der Umwelt in der sich die Agenten bewegen.

*Agenten* repräsentieren die agierenden Entitäten des abzubildenden Systems, dies können sowohl individuelle Einheiten als auch Gruppen von Einheiten sein. Die genauen Bestandteile die eine Entität beinhalten muss, um als Agent zu gelten sind innerhalb des Forschungsfelds umstritten und Thema umfangreicher Diskussionen. Gemeinsam sind allen Agenten allerdings die folgenden vier Eigenschaften:

- *Modularität*: Agenten sind in sich abgeschlossen. Sie sind identifizierbare, diskrete Einheiten mit ihren eigenen Attributen, Verhaltensregeln und der Kapazität selbstständig Entscheidungen zu treffen. Es ist also eindeutig feststellbar ob ein Element des Simulationsmodells Teil eines Agenten, Teil der Umwelt oder ein zwischen allen Agenten – oder einer Teilmenge von Agenten – geteiltes Element ist.
- *Autonomie*: Agenten funktionieren unabhängig in ihrer Umwelt und in ihrem Verhalten gegenüber anderen Agenten.
- *Sozialität* (engl. *sociality*): Agenten sind sozial. Sie interagieren über definierte Protokolle mit anderen Agenten und ihrer Umwelt.
- *Konditionalität* (engl. *conditionality*): Ein Agent hat einen Zustand der sich im Verlauf der Simulationszeit verändern kann. Der Zustand eines Agenten zu einem Simulationszeitpunkt  $t$  wird durch die Ausprägungen seiner Attribute zu diesem Zeitpunkt bestimmt. Der Zustand des Gesamtmodells ergibt sich aus den Zuständen aller Agenten des Modells sowie aus dem Zustand der Umwelt.

Agenten können darüber hinausgehend oft weitere Eigenschaften besitzen, die abhängig vom Modellierungskontext als Voraussetzung für Agentenschaft gesehen werden. Zu diesen Eigenschaften kann gehören, dass Agenten *zielorientiert* sind, d.h. dass ihr Verhalten am Erreichen von Zielen ausgerichtet ist und sie die Effektivität ihrer Entscheidungen anhand dieser Ziele evaluieren. Dazu kann außerdem gehören, dass Agenten *lernfähig* sind, d.h. dass Agenten ein (dynamisches) Gedächtnis haben und dazu in der Lage sind ihr Verhalten aufgrund von Erfahrungswissen anzupassen.

Die *Umwelt* bildet den Aktionsraum der Agenten und ist ein wichtiger Bestandteil zur Modellierung der Interaktionsprinzipien. Sie ist üblicherweise durch das Vorhandensein begrenzter Ressourcen gekennzeichnet und verändert sich im Verlauf der Simulationszeit. Diese Veränderungen sind in der Regel nicht deterministisch oder vorausberechnet, sondern werden dynamisch über eine unterliegende Simulationsebene modelliert, beispielsweise auf Grundlage Zellulärer Automaten. Zur Abbildung der Umwelt finden verschiedene Topologien Anwendung, die dabei lediglich aus pragmatischen Gründen unterschieden werden, da die unterschiedlichen Topologien ineinander überführbar sind und ebenso miteinander kombiniert werden können. Bei der *Suppen-Topologie* (engl. *soup*) handelt es sich um eine Topologie ohne räumliche Ausdehnung, bei der Agenten keinerlei Standortattribute besitzen. Bei der *Gitter-Topologie* hingegen bewegen sich die Agenten über die Zellen eines Zellulären Automaten. Der *euklidische Raum* – sowohl zweidimensional als auch dreidimensional – sowie *Geoinformationssysteme* sind ebenfalls häufig anzutreffende Topologien. Zuletzt können auch *Graphen* bzw. *Netzwerke* zur Repräsentierung der Beziehungen zwischen Agenten Anwendung finden.

Essenziell für die agentenbasierte Simulation ist, dass die Wahrnehmung von Agenten lokal beschränkt ist. Agenten nehmen also stets nur eine begrenzte Zahl von anderen Agenten und Umweltzuständen wahr. Dies bedeutet nicht zwangsläufig eine geografische Beschränktheit, sondern vor allem eine nachbarschaftliche. Häufig ergibt sich aus diesem lokal orientierten Verhalten dabei *Emergenz*, also globale Verhaltensmuster, die aus den individuellen Verhaltensregeln nicht vorhersehbar sind.

Der Ablauf einer agentenbasierten Simulation besteht dann aus dem wiederholten Anstoßen der Verhaltens- und Interaktionsmethoden der einzelnen Agenten. Dies geschieht typischerweise zu diskreten Zeitpunkten und in fixen Zeitinkrementen.

### 4.1.3. Klassifizierung von Modellen zur Verkehrssimulation

Simulationsmodelle für Verkehrssysteme lassen sich anhand des verwendeten (diskreten) Modellierungsansatzes, sowie anhand der im Modell abgebildeten Verkehrsmittel klassifizieren. Darüber hinaus hat sich der Detailgrad des Modells als spezifische Klassifizierungsmethode im Forschungsfeld der Verkehrssimulation etabliert. Vornehmlich wird dabei zwischen makroskopischen, mikroskopischen und mesoskopischen Modellen unterschieden.

Bei *makroskopischen* Simulationsmodellen werden keine einzelnen Entitäten des Systems simuliert, sondern ganze Verkehrsflüsse über Flussraten, Verkehrsdichten und Geschwindigkeiten abgebildet. Dazu werden in der Regel Differenzialgleichungen verwendet, die die Entwicklung des Verkehrs (kontinuierlich) über die Zeit beschreiben und sich üblicherweise an das Verhalten von physikalischen Systemen, wie Flüssigkeiten oder Gasen, anlehnen. Aufgrund der vergleichsweise geringen Zahl an notwendigen Parametern, die darüber hinaus in der Regel direkt im Realsystem gemessen werden können, zeichnen sich makroskopische Simulationsmodelle durch einen geringen Kalibrierungsaufwand aus. Allerdings sind sie nur dann zur Modellierung geeignet, wenn die Interaktionen zwischen den Verkehrsteilnehmern und anderen Teilen des Verkehrssystems keinen signifikanten Einfluss auf das Untersuchungsziel haben. (vergleiche für diesen und die folgenden Absätze [16], S. 15 ff.)

*Mikroskopische* Modelle zur Verkehrssimulation zielen hingegen auf eine feingranulare Abbildung des zu untersuchenden Systems. Hier werden sowohl die einzelnen Verkehrsteilnehmer als auch ihre Interaktionen miteinander und mit der restlichen Verkehrsinfrastruktur detailliert modelliert. Häufig werden die Modelle der Verkehrsteilnehmer dabei in weitere Teilmodelle aufgebrochen. Beispielsweise werden für die Abbildung von Individualverkehr in der Regel drei Teilmodelle verwendet: Ein Fahrzeugfolge-Modell beschreibt das Brems- und Beschleunigungsverhalten, das aus den Interaktionen mit vorausfahrenden Fahrzeugen und anderen Teilen der Verkehrsinfrastruktur resultiert. Ein Spurwechsel-Modell beschreibt das Entscheidungsverhalten beim Spurwechsel, häufig in Abhängigkeit von individuellen Präferenzen des Fahrers. Ein Routenwahl-Modell bildet den Entscheidungsprozess bei der Wahl des übergeordneten Reisewegs ab.

Mikroskopische Simulationsmodelle erfordern aufgrund ihres hohen Detailgrads eine umfangreiche Datenbasis zur Kalibrierung aller Modellparameter und zur Validierung des Modellverhaltens. Oft liegen die notwendigen Daten dabei gar nicht, nicht in ausreichendem Umfang oder im falschen Aggregationsgrad vor und müssen folglich kosten- und zeitaufwendig erhoben werden. Hinzu kommt, dass der hohe Detailgrad eine handhabbare Abbildung umfangreicher Verkehrssysteme mit mehreren hunderttausenden – oder gar Millionen – Entitäten erschwert. Aus diesem Grund beschränken sich mikroskopische Simulationsmodelle üblicherweise auf abgegrenzte Teilbereiche von Verkehrssystemen.

Zwischen makroskopischen und mikroskopischen Simulationsmodellen liegen *mesoskopische* Modelle. Sie versuchen die Vorteile von mikroskopischen und makroskopischen Modellen zu verbinden, indem sie Teile des zu untersuchenden Systems mit mikroskopischer Genauigkeit abbilden, während andere, für das Untersuchungsziel unter Umständen weniger relevante Teilbereiche, mit makroskopischer Genauigkeit modelliert werden. Ihr Spektrum reicht dabei von der detaillierten Abbildung individueller Verkehrsteilnehmer mit stark abstrahierten Verhaltensmodellen, über die vereinfachte Modellierung der Verkehrsinfrastruktur bei gleichzeitiger detaillierter Modellierung der Verkehrsteilnehmer, bis hin zur Zusammenfassung von Verkehrsteilnehmern zu nicht weiter unterscheidbaren Gruppen mit (zeitweise) einheitlichen Verhaltensmustern. Die für die Kalibrierung und Validierung von mesoskopischen Simulationsmodellen notwendige Datenbasis ist in der Regel also größer als bei makroskopischen Modellen, aber kleiner als bei mikroskopischen Modellen. Wie auch makroskopische Modelle, sind mesoskopische Modelle nur dann zur Abbildung eines Verkehrssystems geeignet, wenn vollumfängliche, detaillierte Interaktionen zwischen den Verkehrsteilnehmern untereinander und mit der Infrastruktur des Systems keinen signifikanten Einfluss auf das Untersuchungsziel haben.

## 4.2. Simulationsmodelle für Individual- und Busverkehr

Busse sind, wenn nicht auf explizit ausgezeichneten Busspuren unterwegs, immer Teil eines größeren Individualverkehrssystems. Man kommt deswegen nicht umhin Individualverkehr zu betrachten, möchte man Bussysteme modellieren.

Das wohl bekannteste diskrete Simulationsmodell für Individualverkehr wurde 1992 von Nagel und Schreckenberg in [69] vorgestellt. Die Autoren beschreiben einen Zellulären Automaten mit eindimensionalem Gitter und periodischen bzw. offenen<sup>13</sup> Rändern zur Abbildung von Individualverkehr auf einspurigen Schnellstraßen. Jede Zelle des Automaten kann sich in einem von zwei Zuständen befinden – durch ein Fahrzeug belegt oder nicht – und Fahrzeuge können nur ganzzahlige Geschwindigkeiten zwischen 0 und  $v_{max}$  Zellen annehmen. In jedem Simulationsschritt werden parallel für alle belegten Zellen die folgenden Regeln angewendet:

1. *Beschleunigung*: Ist die aktuelle Geschwindigkeit  $v$  eines Fahrzeugs geringer als die Höchstgeschwindigkeit  $v_{max}$  und die Distanz zum nächsten vorausfahrenden Fahrzeug größer als  $v + 1$ , wird die Geschwindigkeit um eine Einheit erhöht, d.h.  $v \leftarrow v + 1$ .
2. *Abbremsen (aufgrund eines vorausfahrenden Fahrzeugs)*: Besitzt ein Fahrzeug auf Zelle  $i$  und mit aktueller Geschwindigkeit  $v$  ein Vorgängerfahrzeug auf Zelle  $i + j$  (mit  $j \leq v$ ), so wird die Geschwindigkeit auf  $j - 1$  reduziert, d.h.  $v \leftarrow j - 1$ .
3. *Randomisierung*: Ist die aktuelle Geschwindigkeit  $v$  größer 0, wird sie mit einer Wahrscheinlichkeit von  $p$ , der Trödelwahrscheinlichkeit, um eine Einheit reduziert, d.h.  $v \leftarrow v - 1$ .
4. *Bewegung*: Jedes Fahrzeug wird um  $v$  Zellen bewegt.

Nagel und Schreckenberg zeigen anhand künstlicher Beispiele mit periodischen und offenen Rändern, dass bereits dieses einfache Modell dazu in der Lage ist realistisches Verhalten abzubilden: Bei geringer Fahrzeugdichte zeigt sich laminarer Verkehrsfluss, während sich bei höherer Fahrzeugdichte aufgrund der Geschwindigkeitsfluktuationen der Fahrzeuge zufällig Staus und die damit einhergehenden Start-Stopp-Wellen bilden.

Neuere Simulationsmodelle für Individualverkehr auf Basis von Zellulären Automaten werden beispielweise von Tian in [87] und Zamith et al. in [108] vorgestellt. Tian erweitert das Modell von Nagel und Schreckenberg auf mehrspurige Straßen sowie unterschiedliche Fahrzeugklassen und ergänzt das Regelwerk um die Möglichkeit des Spurwechsels. Dazu wird für jede durch ein Fahrzeug belegte Zelle vor der Beschleunigungsentscheidung eine Spurwechselentscheidung getroffen, bei der – genügend Platz auf der potentiellen Zielspur vorausgesetzt – mit einer zell- und fahrzeugklassenabhängigen Wahrscheinlichkeit die Spur gewechselt wird. Neben der Spurwechselwahrscheinlichkeit unterscheiden sich die im Modell abgebildeten Fahrzeugklassen außerdem in Länge (d.h. Zahl der durch das Fahrzeug belegten Zellen) und Höchstgeschwindigkeit. Ihr Modell wendet die Autorin an, um die Auswirkungen von Bushaltestellen vor Signalanlagen auf den Verkehrsfluss zu untersuchen. Dazu betrachtet sie ein einfaches künstliches Beispiel einer zweispurigen Straße mit offenen Grenzen sowie einer Haltebucht für Busse und einer nachgelagerten Signalanlage. Zur Abbildung des Bussystems werden in jedem Simulationsschritt mit einer durch die Nutzerin festzulegenden Wahrscheinlichkeit Busse in das System gespeist, die sich zur Haltebucht bewegen, dort eine deterministische Zeitspanne lang verweilen, wieder aus der Haltebucht herausfahren und anschließend das System verlassen. Zwar ist das Bussystem nur rudimentär abgebildet, es reicht im Rahmen des Untersuchungsziels aber zur Modellierung von realistischem Verhalten aus: Der Verkehrsfluss nimmt mit zunehmendem Anteil von Bussen am Gesamtfahrzeugaufkommen ab, da neu ankommende Busse darauf warten müssen, dass die Haltebucht frei wird und sich in der Folge ein Rückstau am Systemeingang entwickelt. Darüber hinaus wirkt sich der Abstand zwischen Haltebucht und Signalanlage auf den Verkehrsfluss im Bereich der Haltebucht aus, da Busse, aufgrund von Rückstau vor der Signalanlage, daran gehindert werden die Haltebucht zu verlassen.

Zamith et al. erweitern das Modell von Nagel und Schreckenberg hingegen um unterschiedliche Fahrerprofile – d.h. unterschiedliches Fahrverhalten – sowie unterschiedliche Fahrzeuglängen. Die Fahrerprofile werden über betaverteilte Zufallszahlen abgebildet, die den bei der Beschleunigung angesetzten minimalen Fahrzeugabstand beeinflussen. Konkret unterscheiden die Autoren zwischen drei Profilen: Standard, konservativ und riskant. Während sich Fahrer mit Standardprofil dem aktuellen Verkehrsfluss anpassen, d.h. das Ursprungsverhalten des Nagel/Schreckenberg-Modells aufweisen, halten

<sup>13</sup> „Offen“ bedeutet in diesem Kontext, dass Fahrzeuge aus dem System entfernt werden sobald sie den Rand des Gitters erreichen, es handelt sich also um Randzellen mit fester Zellbelegung (siehe Abschnitt 4.1.2.1).

konservative Fahrer einen größeren Abstand zum vorausfahrenden Fahrzeug ein. Riskante Fahrer bevorzugen hingegen einen geringeren Fahrzeugabstand. Auch Zamith et al. wenden ihr Modell auf ein künstliches Beispiel an: Einen einspurigen Streckenabschnitt mit periodischen Rändern. Anhand dieses Beispiel untersuchen die Autoren die Auswirkungen der unterschiedlichen Fahrerprofile auf den Verkehrsfluss, wobei sie innerhalb eines Simulationslaufs nur homogene Profile zulassen, d.h. alle Fahrer das gleiche Verhalten aufweisen. Dennoch zeigt sich hier ebenfalls realistisches Verhalten: Während sich bei den Experimenten mit konservativen Fahrern aufgrund der größeren Fahrzeugabstände früher Staus bilden, kann bei den Experimenten mit riskanten Fahrern auch bei höherer Fahrzeugdichte noch ein laminarer Verkehrsfluss beobachtet werden.

Die Modelle von Tian und Zamith et al. zeigen, dass mit Hilfe von Zellulären Automaten auch komplexe Verkehrssysteme mit unterschiedlichen Teilnehmern und individuellem Verhalten abgebildet werden können. Dabei muss aber mit zunehmender Komplexität weiter vom ursprünglichen Modellierungsansatz des Zellulären Automaten abgewichen werden. So ist die von Tian für ihr Modell verwendete Transitionsfunktion nicht mehr nur vom Zustand einer Zelle und ihrer Nachbarn, sondern auch von der Klasse des die Zelle(n) belegenden Fahrzeugs und ihrer konkreten geografischen Position abhängig, damit unterschiedliches Spurwechselverhalten von Bussen und Individualverkehr vor einer Haltebuch realisiert werden kann.

Sollen Systeme modelliert werden deren Verhalten wesentlich von der Inhomogenität der Verkehrsteilnehmer beeinflusst wird, werden deswegen in der Regel agentenbasierte Modelle angewendet (siehe z.B. [10–12], [14], [31], [50], [51], [54], [55], [83], [84]). Zur (mikroskopischen) agentenbasierten Simulation von Individualverkehr wird dabei häufig das am *Deutschen Zentrum für Luft- und Raumfahrt (DLR)* entwickelte Open Source Softwarepaket *SUMO (Simulation of Urban Mobility)* verwendet (für eine allgemeine Übersicht zu SUMO siehe [11], [54], [55], für einen Überblick verschiedener Projekte siehe z.B. [10], [83], [84]). In SUMO werden Fahrzeuge als Agenten mit individuellen Eigenschaften, wie z.B. Abmessung, Beschleunigungs- und Bremsvermögen, (Höchst-)Geschwindigkeit und Kapazität modelliert, die sich durch das als Graph abgebildete Streckennetz bewegen. Das konkrete Fahrverhalten der Agenten wird durch austauschbare Fahrzeugfolge- und Spurwechsel-Modelle beschrieben, während die Routen einzelner Agenten durch das Netzwerk mit Hilfe eines zentralen Routenwahlmoduls auf Basis von Quell/Ziel-Matrizen bestimmt werden. Neben Fahrzeugen des Individualverkehrs können in SUMO auch Busse (und andere Fahrzeuge des ÖPNV) abgebildet werden (siehe [12], [52]). Alle Agenten des ÖPNV-Systems – unabhängig davon ob Bus oder Bahn – verwenden dabei die gleichen Fahrzeugfolge- und Spurwechsel-Modelle wie die Agenten des Individualverkehrs. ÖPNV-Agenten folgen allerdings festen, durch den Linienplan vorgegebenen Routen, auf denen sie an ebenfalls fest vorgegebenen Orten halten um Passagiere aufzunehmen. Die für den Passagierwechsel benötigte Zeit hängt von einer fahrzeugtypspezifischen Zustiegszeit (pro Passagier) und der Zahl der zusteigenden Passagiere ab. Ist die Passagierkapazität des ÖPNV-Agenten erreicht, müssen eventuell noch am Haltepunkt befindliche Passagiere auf das nächste Fahrzeug warten. Routen des ÖPNV-Verkehrs können darüber hinaus periodisch abgefahren werden, wobei nicht weiter ersichtlich ist ob dies auf Basis von Umläufen realisiert wird, d.h. ÖPNV-Agenten über den simulierten Betriebstag mehrere Fahrten durchführen, und sich Verspätungen damit auf Folgefahrten übertragen können oder ob lediglich periodisch neue Agenten in das Modell gespeist werden.

Reale Anwendungsbeispiele von SUMO werden bspw. von Kastner et al. in [50], Kastner und Pau in [51] sowie Bieker et al. in [14] vorgestellt. Bieker et al. wenden SUMO zur Evaluierung von Verkehrsflüssen während sportlicher Großereignisse in der italienischen Stadt Bologna an. Dazu untersuchen sie drei Szenarien im Umfeld des lokalen Fußballstadions. Diese Szenarien schwanken in der Größe ihrer abgebildeten Streckennetze zwischen 2,5 km<sup>2</sup> mit 111 Straßenzügen bzw. 65 Kreuzungen und 4,1 km<sup>2</sup> mit 268 Straßenzügen bzw. 159 Kreuzungen, sind also vergleichsweise übersichtlich. Die Autoren vergleichen die durch SUMO generierten Verkehrsflüsse mit Realdaten und kommen zu dem Ergebnis, dass das agentenbasierte Modell von SUMO das reale Verhalten hinreichend genau abbildet.

Ein umfangreicheres Anwendungsbeispiel wird von Kastner et al. in [50] besprochen. Die Autoren verwenden SUMO im Rahmen eines österreichischen Projekts zur Erstellung eines *Intelligenten Transport Systems* (engl. *Intelligent Transportation System, ITS*). SUMO soll dazu herangezogen werden eine

Echtzeit-Momentaufnahme der Gesamtverkehrssituation von Oberösterreich und Salzburg zu erstellen, ohne sämtliche Straßenzüge mit Sensoren ausstatten zu müssen. Dazu soll periodisch alle fünf Minuten ein neuer Simulationslauf mit aktualisierten Echtzeitdaten gestartet und die Verkehrssituation in den nächsten fünf Minuten prognostiziert werden. Der Umfang des Anwendungsbeispiels von 323.282 Straßenzügen, 248.749 Kreuzungen, 1,2 Millionen zulässiger Routen und 1,6 Millionen Fahrzeugen führt allerdings zu einem unzureichenden Laufzeitverhalten; das Szenario kann nicht schneller als Echtzeit simuliert werden. Kastner und Pau verwenden deswegen in [51] ein zwischenzeitlich neu in SUMO integriertes mesoskopisches Simulationsmodell auf Basis von Warteschlangen, mit dem die gewünschten Prognosen innerhalb von 20 Sekunden durchgeführt werden können.<sup>14</sup>

Ein weiteres mikroskopisches, agentenbasiertes Simulationsmodell für Individualverkehr wurde 2008 von Doniec et al. in [31] vorgestellt. Dieses Modell wurde als Teil des *ArchiSim*-Projekts des französischen *Instituts für Verkehrs- und Sicherheitsforschung* (franz. *Institut national de recherche sur les transports et leur sécurité, INRETS*) entwickelt und dient der realistischen Abbildung von Fahrerverhalten an Kreuzungen. Einzelne Fahrer werden im Modell als individuelle Agenten abgebildet, die opportunistische Verhaltensweisen und Antizipationsfähigkeiten aufweisen. Opportunistisches Verhalten wird abgebildet, indem Agenten die durch die Straßenverkehrsordnung vorgegebenen normativen Prioritätsregeln an Kreuzungen anhand ihrer individuellen Wahrnehmung der aktuellen Situation bewerten. Diese Bewertung wird durch einen Ungeduldigkeitswert und einen Anfahrtsphasenindikator beeinflusst. Ein Agent wird ungeduldig, sobald seine Wartezeit an einer Kreuzung seinen individuellen Wartezeit-Schwellenwert übersteigt und wird ungeduldiger je weiter dieser Wert überschritten wird. Der Anfahrtsphasenindikator gibt hingegen an, ob sich der Agent zum aktuellen Zeitpunkt noch in der Anfahrtsphase befindet, d.h. nach dem Stopp vor einer Kreuzung gerade erst wieder anfährt. Zusammen mit den Entfernungen zu den Agenten in der lokalen Nachbarschaft werden so alle (potentiellen) Konflikte an einer Kreuzung auf Konflikte zwischen jeweils zwei Agenten heruntergebrochen und aufgelöst. Um Unfälle und Deadlocks zu vermeiden antizipieren die Agenten dabei die Auswirkungen ihres (potentiellen) Verhaltens, indem sie den zukünftigen Zustand des Modells abschätzen. Diese Abschätzung ist ebenfalls von individuellen Eigenschaften der Agenten, wie bspw. der Länge des durch den Agenten abschätzbaren Zeitraums, abhängig und wird mittels Constraint Programming (siehe z.B. [73]) umgesetzt. Nur wenn der Agent den durch ihn geschätzten zukünftigen Modellzustand als „wünschenswert“ einstuft führt er die von ihm geplante Aktion aus, andernfalls modifiziert er sein Verhalten so lange, bis ein für ihn zulässiger Zustand erreicht wird.

Zur Validierung simulieren Doniec et al. eine Kreuzung sowie einen Kreisverkehr, sowohl mit ihrem eigenen als auch mit einem vereinfachten Simulationsmodell von Champion et al. (siehe [21]) und vergleichen die simulierten Verkehrsflüsse mit den real beobachteten. Dabei stellen die Autoren fest, dass ihr Simulationsverhalten das reale Verhalten besser abbildet als das vereinfachte Modell und zur Simulation von Fahrerverhalten an Kreuzungen herangezogen werden kann.

Die hohe Detailfülle mit der individuelle Komponenten in agentenbasierten Simulationsmodellen abgebildet werden, macht sie sehr gut zur Abbildung komplexer, von den Aktionen individueller Teilnehmer abhängiger Situationen. Bei der Modellierung umfangreicher Verkehrsnetze führt diese Detailfülle allerdings zu suboptimalem Laufzeitverhalten, wie das Anwendungsbeispiel von Kastner et al. in [50] sowie Kastner und Pau in [51] demonstriert.

In solchen Fällen sind mesoskopische Simulationsmodelle, wie sie z.B. von Eissfeldt in [33] und Burghout in [16] beschrieben werden, besser geeignet. Eissfeldt entwickelt ein mesoskopisches ereignisbasiertes Modell zur Verkehrsflusssimulation auf Basis der Warteschlangentheorie. Auch in diesem Modell werden einzelne Fahrzeuge als Entitäten mit individuellen Eigenschaften, wie z.B. Länge oder Höchstgeschwindigkeit abgebildet, ihr Fahrverhalten wird aber stark abstrahiert modelliert.

Eissfeldt teilt das Straßennetz in verkettete, gerichtete Segmente, die als Warteschlangensysteme modelliert werden. Betritt eine Fahrzeug-Entität ein Segment, wird ihr auf Basis der Segmentlänge und der

---

<sup>14</sup> Zum Zeitpunkt der Anfertigung dieser Arbeit sind keine detaillierten Informationen über das mesoskopische Simulationsmodell von SUMO öffentlich zugänglich, weswegen eine weitergehende Besprechung an dieser Stelle ausbleibt.

dort zulässigen Höchstgeschwindigkeit ein potentieller Austrittszeitpunkt zugewiesen – d.h. ein neues Ereignis generiert – der ihre Position in der Warteschlange definiert. Zum Zeitpunkt dieses Ereignisses wird die betroffene Fahrzeug-Entität durch einen Server auf ihr anvisiertes Folgesegment transferiert, falls dort genug physische Kapazität vorhanden ist und die Verkehrsflusskapazität zwischen den beiden betroffenen Segmenten durch den Transfer nicht überschritten wird. Letzteres wird durch stochastische Serviceraten gewährleistet, die sowohl vom Ausgangssegment als auch der dort vorzufindenden Verkehrssituation abhängen und z.B. unterschiedliche Grünzeitanteile von Ampelanlagen abbilden. Um unrealistisches Modellverhalten, wie z.B. ortsfeste Staufronten oder unverzügliches Schließen von Lücken, zu vermeiden, sind die Serviceraten zusätzlich von der Verkehrssituation auf dem Zielsegment abhängig.

Eine explizite Abbildung einzelner Spuren findet in Eissfeldts Modell nicht statt. Stattdessen werden Spuren implizit über Verkehrsflussraten und Kapazitätsbeschränkungen von Segmenten modelliert. Überholvorgänge werden mit Hilfe von Überholwahrscheinlichkeiten modelliert: Eine neu auf einem Segment ankommende Fahrzeug-Entität überholt die ihr direkt vorgelagerte Entität mit einer von der aktuellen Fahrzeugdichte abhängenden Wahrscheinlichkeit, wenn

- die neu ankommende Fahrzeug-Entität eine höhere Höchstgeschwindigkeit als ihr Vorgänger besitzt
- der zeitliche Abstand zwischen den beiden Fahrzeug-Entitäten klein genug ist und
- eine gewisse Zeitspanne seit dem letzten Überholvorgang der neu ankommenden Fahrzeug-Entität vergangen ist.

Eissfeldt validiert sein Modell, indem er es auf künstliche Beispiele, sowohl mit periodischen als auch offenen Rändern, anwendet und die resultierenden Verkehrsflüsse mit denen eines mikroskopischen Fahrzeugfolge-Modells vergleicht. Die Ergebnisse zeigen, dass das Warteschlangenmodell ähnlich realistisches Verhalten wie das mikroskopische Fahrzeugfolge-Modell aufweist. Darüber hinaus zeigt Eissfeldt, dass sein Modell auch zur Simulation von Szenarien realistischer Größe anwendbar ist, indem er es dazu verwendet den Verkehrsfluss auf den Kölner Hauptverkehrswegen und Autobahnen zu simulieren und die resultierenden Abgasemissionen zu prognostizieren.

Ein alternativer Modellierungsansatz wird von Burghout in [16] vorgestellt. Der Autor entwickelt ein mesoskopisches, ereignisbasiertes Simulationsmodell namens *Mezzo*, das er mit dem proprietären, mikroskopischen Simulationsmodell *MITSIMLab* (siehe z.B. [105]) zu einem hybriden Modell kombiniert, sodass Bereiche von besonderem Interesse mit mikroskopischer Genauigkeit simuliert werden können, während die umgebenden Bereiche mesoskopisch abgebildet werden.

Wie Eissfeldts Modell in [33], werden in *Mezzo* einzelne Fahrzeuge als Entitäten mit individuellen Eigenschaften und stark abstrahiertem Fahrverhalten abgebildet. Auch hier wird das zu untersuchende Straßennetz als System verketteter Warteschlangensysteme modelliert. Dazu wird es in einen Graphen überführt, dessen Knoten die Server der Warteschlangensysteme beherbergen und Orte des Netzes darstellen, an denen mehrere Verkehrsflüsse zusammen- und/oder auseinanderlaufen. Straßenabschnitte zwischen diesen Orten werden durch gerichtete Kanten modelliert, die einzelne Warteschlangen verwalten. Wie beim Modell von Eissfeldt, wird neu auf einer Kante ankommenden Fahrzeug-Entitäten ein potentieller Austrittszeitpunkt zugewiesen, anhand dessen die Reihenfolge der Fahrzeug-Entitäten auf der Kante bestimmt wird. Im Gegensatz zu Eissfeldts Modell, wird dabei aber eine Geschwindigkeit/Dichte-Funktion verwendet, d.h. die aktuelle Verkehrssituation bei der Berechnung berücksichtigt. Die Anwendung solcher Funktionen setzt dabei homogene Verkehrsbedingungen auf dem modellierten Straßenabschnitt voraus; eine Annahme, die durch Kapazitätslimitationen am Ende eines Straßenabschnitts (z.B. aufgrund von Ampelanlagen) ungültig wird. Deswegen teilt Burghout Kanten in einen *Warteschlangenabschnitt* („*queue part*“) sowie einen *Fahrabschnitt* („*running part*“) mit dynamischer Grenze ein (vgl. Abbildung 4.4) und verwendet zur Bestimmung potentieller Austrittszeitpunkte nur die Fahrzeugdichte auf dem Fahrabschnitt. Dieser Abschnitt beinhaltet die Fahrzeug-Entitäten, die auf dem

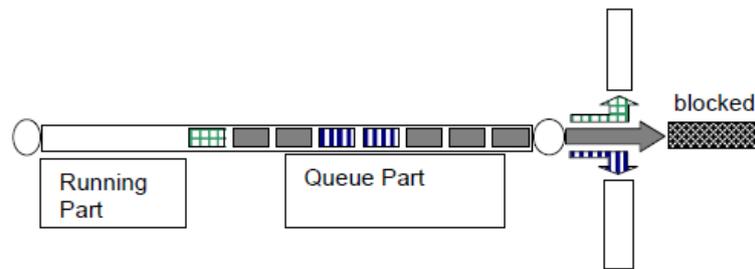


Abbildung 4.4: Modellierung von Kanten und Abbiegebewegungen in Mezzo.  
Entnommen aus [16], S. 50

Weg zum Zielknoten der Kante sind und (noch) nicht durch ein flussabwärts befindliches Kapazitätslimit verzögert sind. Im Gegensatz dazu beinhaltet der Warteschlangenabschnitt zu Simulationszeitpunkt  $t$  alle Fahrzeug-Entitäten, die einen potentiellen Austrittszeitpunkt kleiner als  $t$  besitzen, d.h. die Entitäten, die die Kante bereits verlassen haben sollten, dazu aber aufgrund eines Kapazitätslimits flussabwärts (noch) nicht in der Lage waren. Mit zunehmender (abnehmender) Zahl verzögerter Fahrzeug-Entitäten verschiebt sich so die Grenze zwischen den beiden Abschnitten und der Fahrabschnitt wird kürzer (länger), wodurch die minimale Fahrzeit auf der Kante beeinflusst wird.

Wie beim Modell von Eissfeldt, müssen Fahrzeug-Entitäten zum Transfer zwischen Kanten von einem Server verarbeitet werden. Burghout weist dazu jedem Knoten des Graphen für jede potentielle Verbindung zwischen ihren inzidenten Kanten eine *Abbiegebewegung*<sup>15</sup> („turning movement“) zu (vgl. ebenfalls Abbildung 4.4). Diese Abbiegebewegungen bilden die Server des Warteschlangensystems und besitzen spezifische, stochastische Serviceraten. Sie teilen sich den Warteschlangenbereich ihrer eingehenden Kante und verarbeitet ausschließlich Fahrzeug-Entitäten deren Route die entsprechende Zielkante beinhaltet. Während bei der Modellierung der Straßenabschnitte, wie im Modell von Eissfeldt, unterschiedliche Spuren nicht explizit abgebildet werden, werden sie bei den Übergängen zwischen Straßenabschnitten explizit berücksichtigt. Für jede Spur eines eingehenden Straßenabschnitts besitzt der Zielknoten der entsprechenden Kante im Modellgraphen eine eigenständige Abbiegebewegung. So kann abgebildet werden, dass mehrere Fahrzeuge zeitgleich eine Kreuzung überqueren können, falls sie sich auf unterschiedlichen Spuren befinden. Um darüber hinaus auch Interaktionen zwischen den Wartebereichen verschiedener Abbiegebewegungen abzubilden – etwa wenn der Zugang zu einer Linksabbiegerspur blockiert ist, weil viele Fahrzeuge geradeaus fahren wollen – besitzt jede Abbiegebewegung ein individuelles Lookback-Limit, d.h. eine maximale Anzahl an Fahrzeug-Entitäten, die sie in der gemeinsamen Schlange zurückschauen kann, um zu verarbeitende Fahrzeuge zu finden.

Sein Modell validiert Burghout sowohl durch Plausibilitätsuntersuchungen anhand eines künstlichen Netzes als auch durch Anwendung seines Modells auf zwei reale Szenarien unterschiedlicher Größe. Der künstliche Testfall besteht aus 15 hintereinander geschalteten, zweispurigen Straßenabschnitten, von denen einer künstlich blockiert wird, um einen Unfall zu simulieren. Dabei zeigt sich, dass das Modell dazu in der Lage ist fundamentales Verkehrsverhalten, wie den bei einer Blockade zu erwartenden Rückstau und die damit einhergehenden Start-Stopp-Wellen abzubilden. Die beiden realen Szenarien werden hingegen dazu verwendet die durch das Simulationsmodell berechneten Kennzahlen mit Realdaten zu vergleichen. Dazu wird zum einen ein 2,5 km langer Autobahnabschnitt samt Ab- und Auffahrt sowie ein Teil eines 5 km<sup>2</sup> großen, gemischten Straßennetzes – d.h. mit und ohne Signalisierung – nördlich von Stockholm simuliert. Auch bei diesen Anwendungsbeispielen kann das Modell die Verkehrsflüsse hinreichend präzise abbilden, wobei dies beim kleineren Autobahnscenario aufgrund einer besseren Datengrundlage – und in Folge dessen einfacheren Kalibrierung – besser gelingt, als beim umfangreicheren gemischten Straßennetz.

Weder Eissfeldt in [33] noch Burghout in [16] machen detaillierte Aussagen über das Laufzeitverhalten der von ihnen entwickelten Simulationsmodelle. Burghout erwähnt allerdings, dass Mezzo auf einem Pentium Prozessor mit 1 GHz und 256 MB Ram unter Linux dazu in der Lage ist Netze mit über 1000 Knoten und 2000 Kanten dreimal schneller als Echtzeit zu simulieren.

<sup>15</sup> Das Überfahren einer Kreuzung ohne Abbiegen wird ebenfalls als Abbiegebewegung deklariert.

Wie die vorangegangene Besprechung exemplarisch verdeutlicht, zielen Simulationsmodelle für Individualverkehr in der Regel auf die Abbildung von Verkehrsflüssen (vgl. [11], [12], [14], [16], [33], [50–52], [54], [55], [69], [108]) oder spezifischer Verkehrssituationen (vgl. [31], [87]). Um zur Bewertung der Leistungsfähigkeit von ÖPNV-Systemen herangezogen werden zu können, müssen sie deswegen häufig aufwendig erweitert werden. Für Mezzo wird eine Erweiterung zur Simulation von Bussystemen (*BusMezzo*) von Toledo et al. in [88] vorgestellt. Dazu führen die Autoren, z.T. abgeleitete, Klassen für unterschiedliche Bustypen, individuelle Busfahrzeuge, Haltepunkte, Linien, Routen und Fahrten ein (siehe Abbildung 4.5). *Bustypen* kapseln für alle Busse eines bestimmten Typs identische Eigenschaften, wie die Passagierkapazität oder die Fahrzeugabmessung. *Busfahrzeuge* verwalten hingegen fahrzeugspezifische, dynamische Eigenschaften, wie die aktuelle Anzahl an Passagieren oder die durch das Fahrzeug zu bedienenden Fahrten. Letzteres ermöglicht die Modellierung von Fahrzeugumläufen und in der Folge die Berücksichtigung von Wende- und Ruhezeiten, sowie Verspätungsübertragungen zwischen aufeinander folgenden Fahrten.

Das Liniennetz wird über Buslinien, Busrouten und Haltepunkte abgebildet. *Buslinien* bestehen aus einem Quell/Ziel-Paar, d.h. einem Start- und Zielknoten im übergeordneten Straßennetz-Modell von Mezzo, sowie einer Liste zwischenliegender, anzufahrender Haltepunkte. Darüber hinaus verwalten Buslinien alle derzeit aktiven sowie noch durchzuführenden Fahrten und eine Liste von Haltepunkten an denen bei Bedarf korrigierend in den laufenden Betrieb eingegriffen werden kann. Auf welchem Weg eine Fahrzeug-Entität von einem Haltepunkt zum nächsten gelangt wird hingegen durch die *Busrouten* über eine Liste abzufahrender Kanten des Modellgraphen definiert. Die Haltepunkte sind diesen Kanten zugeordnet, verwalten Informationen über ihre physischen Charakteristiken (wie z.B. Länge und Typ), eine Liste der sie bedienenden Linien und deren letzte Ankunftszeiten, sowie die Zahl der aktuell an ihnen wartenden Passagiere. Da Haltepunkte an einer beliebigen Position auf ihrer zugehörigen Kante verortet sein können, Fahrzeiten in Mezzo aber stets nur für ganze Kanten bestimmt werden, wird die Fahrzeit zum nächsten Haltepunkt als Anteil der Gesamtfahrzeit berechnet, indem die Entfernung zum Haltepunkt ins Verhältnis zur Gesamtlänge der Kante gesetzt wird.

Den für ÖPNV-Systeme zentralen Halteprozess modellieren Toledo et al. mit Hilfe dynamischer, linien- und haltepunktspezifischer Passagierankunfts- und Ein-/Ausstiegsraten sowie unter Berücksichtigung eventuell anzuwendender Kontrollmaßnahmen, d.h. zusätzlichen Wartezeiten. Zur Abbildung der Passagierankünfte verwenden die Autoren die Poissonverteilung, während die Ausstiegsrate mit Hilfe

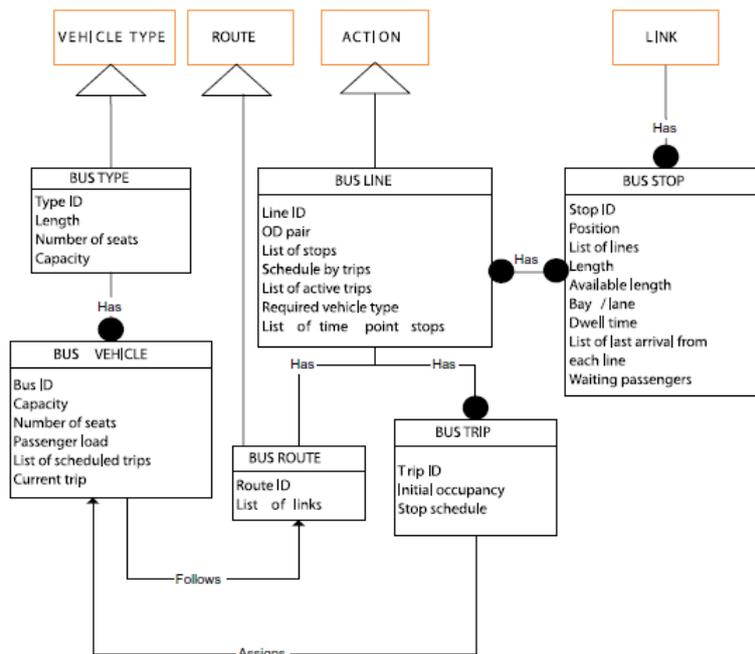


Abbildung 4.5: Klassendiagramm zur Erweiterung von Mezzo um Busse, entnommen aus [88], S. 898

der Binomialverteilung modelliert wird. Die für den Ein-/Ausstiegsprozess notwendige Zeitdauer wird anhand der Zahl ein-/aussteigender Passagiere sowie durch von der Nutzerin zu definierende Ein-/Ausstiegszeiten (pro Passagier) bestimmt. Für jede Fahrzeugtür wird dabei eine separate Ein-/Ausstiegszeit berechnet und die Gesamthaltezeit als Maximum der Einzelzeiten bestimmt. Die Haltezeit kann anschließend noch durch die am Haltepunkt durchzuführenden Kontrollmaßnahmen weiter erhöht werden.

Zusätzlich zur Einführung neuer Klassen, muss auch die Ereignisverarbeitung von Mezzo erweitert werden. Die in der ursprünglichen Version verwendeten Ereignistypen zur Generierung und Entfernung von Fahrzeugen, zur Initiierung der Verarbeitung einer Abbiegebewegung und zur Änderung der Verkehrsnachfrage reichen zur Abbildung von ÖPNV nicht aus. Toledo et al. führen deswegen die fünf Ereignistypen *Ankunft auf Kante* („Enter link“), *Einfahrt in Haltepunkt* („Enter stop“), *Ausfahrt aus Haltepunkt* („Exit stop“), *Fahrtbeginn* („Trip departure“) und *Fahrtende* („Trip end“) ein. Abbildung

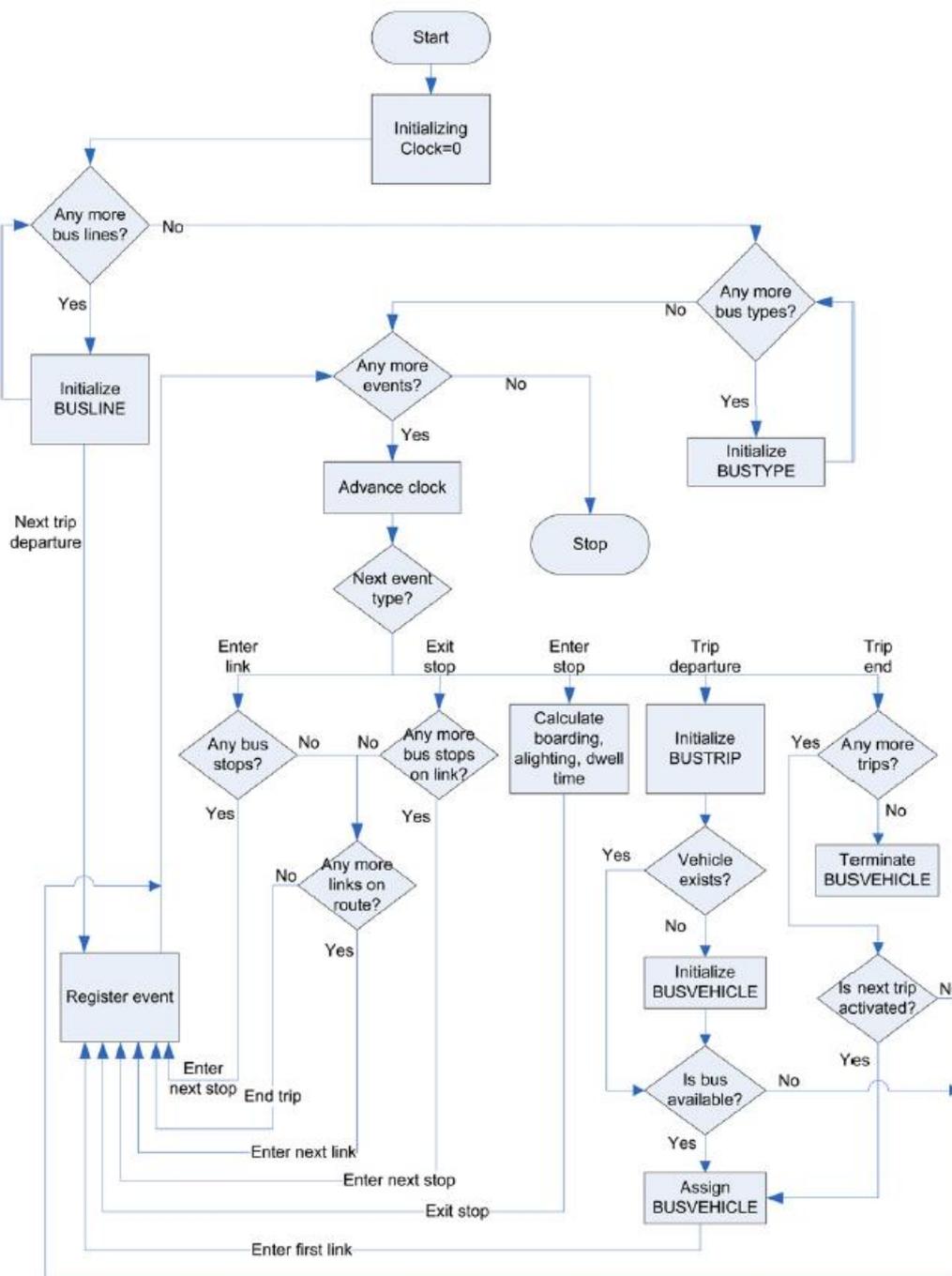


Abbildung 4.6: Flussdiagramm des Simulationsablaufs für den ÖPNV in Mezzo, entnommen aus [88], S. 900

4.6 zeigt den Zusammenhang zwischen den verschiedenen Ereignistypen sowie deren Stellung im Simulationsablauf als Flussdiagramm.

Ihre Modellerweiterung testen die Autoren anhand der Hin- und Rückrichtung einer einzelnen Buslinie in der israelischen Stadt Tel-Aviv, wobei sie neun verschiedene Szenarien mit jeweils unterschiedlicher Verkehrsnachfrage und Fahrzeitschwankungen untersuchen. Die Ergebnisse ihrer Untersuchungen lassen auf plausibles Verhalten schließen, wobei das Modell – so die Autoren – Schwankungen der Sicherheitsabstände an später auf der Linie befindlichen Haltepunkten zu unterschätzen scheint. Toledo et al. vermuten die Ursache dafür darin, dass sie bei ihren Experimenten den Individualverkehr nicht explizit simulieren. Stattdessen parametrisieren sie die Auswirkungen des Individualverkehrs durch die bei der Fahrzeitbestimmung verwendeten Geschwindigkeit/Dichte-Funktionen. Dies könnte zur Folge haben, dass die Auswirkungen von Interaktionen des Individualverkehrs, wie bspw. spontane Staubildung und sich über adjazente Kanten übertragender Rückstau, unterschätzt werden.

Modelle die explizit zur Simulation von Busverkehr entwickelt wurden, werden unter anderem von Andersson et al. in [3] und Meignan et al. in [67] vorgestellt. Bei dem von Andersson et al. bereits 1979 vorgestellten Modell handelt es sich um eine mesoskopische, ereignisbasierte, interaktive Simulation, die es Nutzerinnen ermöglicht im laufenden Simulationsbetrieb Verkehrsmanagementstrategien, wie z.B. Fahrtverkürzungen oder Leerfahrten, zu veranlassen und zu vergleichen. Dazu bilden die Autoren das Busnetz als Menge von Linien ab, die Sammlungen verketteter Haltepunkte darstellen. Jeder dieser Haltepunkte besitzt wiederum für jede ihn bedienende Linie einen eigenständigen Haltebereich, d.h. im Simulationsmodell werden physische Interaktionen zwischen Fahrzeugen unterschiedlicher Linien an gemeinsamen Haltepunkten nicht direkt abgebildet. Dies geschieht hingegen indirekt über den Ein-/Ausstiegsprozess, da Passagiere an einem Haltepunkt durch unterschiedliche Buslinien bedient werden können. Wie beim Modell von Toledo et al., wird der Ein-/Ausstiegsprozess vornehmlich durch die Zahl ein-/aussteigender Passagiere, die durch die Nutzerin zu definierende Ein-/Ausstiegszeiten (pro Passagier), sowie eventuelle Kontrollmaßnahmen beeinflusst. Auch beim Modell von Andersson et al. wird für die Modellierung der Passagierankünfte die Poissonverteilung herangezogen, während zur Bestimmung der Zahl aussteigender Passagiere die Binomialverteilung verwendet wird. Im Gegensatz zu Toledo et al. wird die für den Ein-/Ausstiegsprozess benötigte Gesamtzeit bei Andersson et al. aber als gammaverteilte Zufallszahl modelliert, deren Mittelwert und Standardabweichung aufgrund statistischer Überlegungen zu den Zusammenhängen zwischen den Ein-/Ausstiegsströmen an den unterschiedlichen Türen der Fahrzeuge bestimmt werden.

Die Fahrzeiten zwischen zwei Haltepunkten modellieren Andersson et al. über tageszeitabhängige, lognormalverteilte Zufallszahlen. Ihre Wahl begründen die Autoren damit, dass eine Verteilung für Fahrzeiten rechtsschief sein sollte, da eine (geringe) Wahrscheinlichkeit sehr hoher Fahrzeiten existiert (bspw. aufgrund eines Staus), während Geschwindigkeitsbegrenzungen und Fahrzeugkapazitäten zu einer positiven unteren Grenze für Fahrzeiten führen.

Zur Validierung ihres Modells wenden es die Autoren zur Simulation eines kurzen, von mehreren Buslinien bedienten, Abschnitts von Stockholms Busnetz an und vergleichen die so generierten Kennzahlen durch statistische Anpassungstests mit real beobachteten Daten. Dabei kommen sie zu dem Ergebnis, dass die Verteilungen für Fahrzeiten und Ein-/Ausstiegsprozess zur qualitativen Abbildung der im Realsystem beobachteten Phänomene geeignet sind, für eine genauere Abbildung allerdings umfangreichere Realdaten notwendig sind.

Ein neueres Modell zur Simulation von Busverkehr wird von Meignan et al. in [67] vorgestellt. Die Autoren entwickeln ein agentenbasiertes, mesoskopisches Simulationsmodell, das Busse und Reisende beinhaltet. Sowohl Busse als auch Reisende werden dabei als Agenten abgebildet, während Individualverkehr makroskopisch über Verkehrsflüsse modelliert wird.

Basis des Modells von Meignan et al. sind drei zusammenhängende (Teil-)Netzwerke: Das *Straßennetzwerk*, auf dem die Verkehrsflüsse des Individualverkehrs berechnet werden und auf dem sich die Bus-Agenten bewegen, das *Busnetzwerk*, das die Routen der Bus-Agenten durch das Straßennetzwerk definiert sowie ein *Fußgängernetzwerk*, das alle fußläufigen Verbindungen zwischen Orten des Gesamtnetzwerks beinhaltet.

Das Busnetz wird durch Routen, Linien, Haltepunkte und Haltestellen modelliert. Eine *Route* ist ein gerichteter Pfad durch das Straßennetzwerk, während eine *Linie* ähnliche Routen, bspw. Hin- und Rückrichtung, zusammenfasst. Ein *Haltepunkt* ist ein Ort auf einer Route an der ein Bus für den Passagierwechsel hält. Geographisch zusammengehörende Haltepunkte werden durch eine *Haltestelle* gruppiert.

Fußgänger- und Busnetzwerk sind über die Haltepunkte miteinander verbunden. Darüber hinaus ist das Fußgängernetzwerk mit dem Straßennetzwerk verknüpft und beinhaltet auch weitere, nur fußläufig erreichbare Orte des abzubildenden Gesamtnetzes.

Bus-Agenten bewegen sich anhand der Vorgaben des Busnetzwerks durch das Straßennetzwerk, wobei die von ihnen benötigten Fahrzeiten aufgrund von Geschwindigkeit/Dichte-Funktionen bestimmt werden. Meignan et al. machen keine weiteren Angaben darüber, ob bei der Modellierung des Fahrprozess Interaktionen zwischen Bussen berücksichtigt werden. Die Bewegung von Bus-Agenten über einen Betriebstag hinweg wird durch den Fahrplan vorgeschrieben. Wie Toledo et al. in [88], modellieren auch Meignan et al. dazu Umläufe und sind folglich in der Lage Warte- und Ruhezeiten sowie Verspätungsübertragungen zwischen aufeinanderfolgenden Fahrten zu modellieren.

Erreicht ein Bus-Agent einen Haltepunkt wird ihm eine von der Zahl der dort ein-/aussteigenden Passagiere abhängende Haltezeit zugewiesen. Auch hier machen die Autoren keine weiteren Angaben über die konkrete Ausgestaltung des Teilmodells.

Reisende-Agenten können mehrmals täglich unterschiedliche Reisen unternehmen und sich dabei durch alle drei Netzwerke bewegen, also auch Individualverkehr zur Fortbewegung nutzen. Welche(s) (Kombination von) Fortbewegungsmittel(n) ein Reisender-Agent nutzt, wird durch ein Routenwahl-Modell entschieden, bei dem stochastische Anteile in der Nutzenfunktion für unterschiedliche Routenalternativen einer Extremwertverteilung folgen (multinomiales Logit-Modell, siehe z.B. [78], S. 231ff.).

Meignan et al. validieren ihr Modell, indem sie Plausibilitätsuntersuchungen anhand eines Modells des Busnetzes der französischen Stadt Belfort durchführen. Ihre qualitativen Untersuchungen des simulierten Passagieraufkommens und der simulierten Passagierwartezeit zeigen dabei, dass das Simulationsmodell sinnvolles Verhalten aufweist.

### 4.3. Simulationsmodelle für schienengebundenen Verkehr

Im Gegensatz zu Individual- und Busverkehr weist schienengebundener Verkehr eine weitestgehend von anderen Verkehrsarten isolierte Infrastruktur auf. Darüber hinaus zeichnet er sich sowohl durch homogenere Infrastrukturbausteine als auch Verkehrsteilnehmer aus. Aus diesen Gründen werden schienengebundene Verkehrssysteme häufig mit Zellulären Automaten oder ereignisbasierter Simulation modelliert (siehe z.B. [45], [49], [58], [60], [89]).

Ein Simulationsmodell auf Basis eines Zellulären Automaten wird von Joisten in [49] vorgestellt und später von Lückemeyer in [58] verfeinert. Das Modell basiert auf dem von Nagel und Schreckenberg in [69] vorgestellten Modell für Individualverkehr, modifiziert dieses aber signifikant. Im Gegensatz zum Ursprungsmodell von Nagel und Schreckenberg verwendet das Zellularautomatenmodell von Joisten ein irreguläres Gitter mit unterschiedlichen Zelltypen und inhomogenen Nachbarschaften. Darüber hinaus können Zellen in Joistens Modell mehrere Zustandsvariablen aufweisen und auch mit nicht unmittelbar benachbarten Zellen interagieren.

Wie beim Modell von Nagel und Schreckenberg, teilt Joisten das Streckennetz in disjunkte, äquidistante, nur in eine Richtung befahrbare Zellen auf, die entweder belegt oder frei sein können. Zusätzlich speichert jede Zelle die Ursache für ihre Belegung, z.B. durch eine Bahn oder aufgrund einer roten Ampel. Dies hat zum einen zur Folge, dass Abbremsvorgänge allgemein, aufgrund von belegten Zellen, modelliert werden können. Zum anderen kann eine Zelle so zeitgleich aus mehreren Gründen belegt sein, wodurch eine Freigabe erst erfolgen darf, wenn alle Belegungsursachen aufgehoben wurden.

Die Zellen des Automaten unterteilt Joisten in drei Typen: Haltepunkte, Weichen und Gleise. An *Haltepunkt-Zellen* müssen Bahnen mindestens so lange verweilen bis der Passagierwechsel abgeschlossen ist, aber auf jeden Fall bis zum im Fahrplan angegebenen Abfahrtszeitpunkt. Die für den Passagierwechsel notwendige Zeit bildet Joisten über eine, durch die Nutzerin definierbare, Mindesthaltezeit ab, die mit einer, ebenfalls durch die Nutzerin wählbaren, Wahrscheinlichkeit um einen zufälligen Wert überschrit-

ten wird.<sup>16</sup> Den eigentlichen Halt veranlasst Joisten durch Blockade der auf die Haltepunkt-Zelle folgenden Zelle. Sie wird erst wieder freigegeben, nachdem die für den Passagierwechsel veranschlagte Zeit verstrichen ist.

*Weichen-Zellen* werden von Joisten in zusammenführende und teilende Weichen-Zellen, mit jeweils unterschiedlichem Verhalten, unterschieden. Zusammenführende Weichen-Zellen verbinden zwei eingehende mit einer ausgehenden Gleis-Zelle und dürfen zu jedem Simulationszeitpunkt nur über eine der beiden eingehenden Gleis-Zellen befahrbar sein. Dies geschieht, indem zusammenführende Weichen-Zellen in jedem Simulationsschritt die Belegungszustände der (Nachbar-)Zellen ihrer eingehenden Gleise kontrollieren. Nähert sich auf einem der eingehenden Gleise ein Fahrzeug, wird die Vorgängerzelle der Weichen-Zelle auf dem anderen eingehenden Gleis blockiert. Nähern sich auf beiden Gleisen Bahnen, wird die Vorgängerzelle des Gleises mit der weiter entfernten Bahn blockiert. Bei teilende Weichen-Zellen ist keine explizite Blockade des Eingangs notwendig, allerdings muss bei ihrer Überfahrt eine Entscheidung bezüglich der zu verwendenden Folgezelle getroffen werden. Zu diesem Zweck speichert Joisten in teilenden Weichen-Zellen Informationen über die Ziele – d.h. Haltepunkte – der nachfolgenden Gleise. Alle Weichen-Zellen besitzen darüber hinaus eine ihnen eigene Höchstgeschwindigkeit, sodass Bahnen, auch bei einer für sie freigegebenen Weichen-Zelle, unter Umständen einen Bremsvorgang einleiten müssen.

Neben Gleisen, Haltepunkten und Weichen bildet Joisten in ihrem Modell auch Ampeln ab. Sie regeln den Verkehrsfluss an Verknüpfungspunkten mit dem Individualverkehr und sind von Signalanlagen, die die Einfahrt zu Streckenabschnitten im Schienennetz regeln, zu unterscheiden. Mit Ampeln besetzte Zellen speichern sowohl die aktuelle Ampelphase (Rot oder Grün) sowie die noch verbleibende Zeitdauer bis zum Umschalten in die nächste Phase. Eine Ampel in der Rotphase wird abgebildet, indem die ihr vorgelagerte Zelle blockiert wird.

Weiterhin bildet Joisten in ihrem Modell auch Störungen ab. Kleinere Störungen (z.B. durch verzögertes Handeln des Fahrers) werden, ähnlich wie beim Modell von Nagel und Schreckenberg, über eine Trödelwahrscheinlichkeit abgebildet, während größere Störungen durch langfristige Blockierungen von Zellen modelliert werden.

Zur Modellierung des Fahrplans bildet auch Joisten Umläufe, die den einzelnen Fahrzeugen zu Beginn der Simulation zugewiesen werden. Dabei werden Leerfahrten, d.h. Fahrten vom letzten Haltepunkt einer gerade beendeten Fahrt zum ersten Haltepunkt der nächsten Fahrt, mangels vorliegender Daten ignoriert. Bahnen werden stattdessen nach Abschluss einer Fahrt von der entsprechenden Haltepunkt-Zelle entfernt und zu Beginn ihrer nächsten Fahrt auf den zugehörigen Starthaltepunkt gesetzt.

Damit ergibt sich das folgende modifizierte Regelwerk zur Aktualisierung der Zellzustände:

1. Aktualisierung von
  - a. Ampeln
  - b. Blockierungen/Störungen
  - c. Haltepunkt-Zellen
  - d. Weichen-Zellen
2. Entfernung von Fahrzeugen, die ihre aktuelle Fahrt beendet haben
3. Einsetzen von Fahrzeugen die eine neue Fahrt beginnen
4. Beschleunigung
5. Abbremsen (aufgrund einer blockierten Zelle)
6. Randomisierung
7. Bewegung

Dabei ist zu beachten, dass Joisten, zur Abbildung der langen Bremswege von Bahnen, beim Beschleunigen und Abbremsen in jedem Zeitschritt nur Veränderungen um eine Einheit zulässt. In Folge dessen kommt Regel 6 nur zur Anwendung, falls ein Fahrzeug nicht bereits abgebremst hat.

Ihr Modell wendet Joisten zur Simulation des Kölner Stadtbahnnetzes aus dem Jahr 2001 an. Dabei wählt sie eine Zellgröße von 30 Metern, d.h. eine Bahnwagenlänge, womit Bahnen jeweils zwei Zellen des Automaten belegen. Durch diese Wahl der Zellabmessung ergibt sich für die simulierten Bahnen

---

<sup>16</sup> Leider gibt Joisten in [49] nicht an welche konkrete Verteilung an dieser Stelle verwendet wird.

ein minimales Beschleunigungsvermögen von 30 Metern pro Simulationsschritt. Da die in Köln vornehmlich zum Einsatz kommenden Stadtbahnwagen der Vossloh Kiepe GmbH von Typ K4000 über ein maximales (Not-)Bremsvermögen von  $3 \text{ m/s}^2$  verfügen (siehe [98]), entspricht in der Folge ein Simulationszeitschritt vier Sekunden im Realsystem und es können vier verschiedene Geschwindigkeitsstufen abgebildet werden: 0 km/h (0 Zellen), 30 km/h (1 Zelle), 50 km/h (2 Zellen) und 75 km/h (3 Zellen).

Joistens Untersuchungen anhand des Kölner Stadtbahnnetzes zeigen die prinzipielle Tauglichkeit des Modellierungsansatzes, verdeutlichen aber auch die Auswirkungen der sehr groben Gitterauflösung: Bereits leichte Erhöhungen in der Trödelwahrscheinlichkeit  $p$  haben extrem starke Auswirkungen auf die in der Simulation gemessenen Verspätungen. Positiv zu erwähnen ist hingegen, dass der Zellularautomatenansatz dazu in der Lage ist einen vollständigen Betriebstag auf einem Rechner aus dem Jahr 2002 in weniger als zwei Minuten zu simulieren.

Lückemeyer stellt in [58] eine Verbesserung von Joistens Modell vor, indem er die Zelllänge auf zehn Meter reduziert. Dadurch erhöht er die Zahl der abbildbaren Geschwindigkeiten auf acht, von 0 km/h bis 80 km/h, reduziert die während eines Simulationszeitschritts verstreichende Zeit im Realsystem von vier auf drei Sekunden und ermöglicht die Abbildung normaler Bremsvorgänge. Die erneute Anwendung auf das Kölner Stadtbahnnetz aus dem Jahr 2001 zeigt zwar verbessertes Verhalten, allerdings zeigen sich auch hier noch Simulationsartefakte. So entstehen auf den langen Überlandstrecken außerhalb von Köln zu starke Verspätungseffekt, was – so Lückemeyer – Folge der zu geringen Höchstgeschwindigkeit ist.

Um diese Artefakte weiter zu reduzieren entwirft Lückemeyer, ebenfalls in [58], ein mikroskopisches, ereignisbasiertes Simulationsmodell für Stadtbahnverkehr. Dazu bildet er das Stadtbahnnetz als gerichteten Graphen ab, in dem Gleise als Kanten und Haltepunkte sowie Weichen als Knoten repräsentiert werden. Knoten haben dabei keine geographische Ausdehnung, sondern befinden sich stattdessen immer am Anfang bzw. Ende von Gleisen und sind mit diesen verknüpft. Haltepunkte entsprechen demnach Haltekoordinaten auf Gleisen und keinen vollständigen Bahnsteigen. Wie auch Joisten, inkludiert Lückemeyer in seinem Modell Ampeln aber keine Signalanlagen. Ampeln werden von den Kanten ihrer zugehörigen Gleise verwaltet, haben – ebenfalls wie bei Joisten – feste Schaltzeiten und ihre Phasenwechsel werden als Ereignisse modelliert.

Fahrzeuge werden im Modell von Lückemeyer als Entitäten abgebildet, die sich über die Kanten des Graphen bewegen und deren Fahrverhalten ereignisbasiert modelliert ist. Dazu identifiziert der Autor die vierzehn in Tabelle 4.1 aufgelisteten Ereignistypen.

Ereignistyp	Beschreibung
<b>TRAIN_START</b>	Bahn startet eine Fahrt
<b>TRAIN_END_ROUTE</b>	Bahn beendet eine Fahrt
<b>STANDING</b>	Bahn steht
<b>ACCELERATE</b>	Bahn leitet Beschleunigungsvorgang ein
<b>MOVE</b>	Bewegung mit gleichbleibender Geschwindigkeit
<b>BRAKE</b>	Bahn leitet Bremsvorgang ein
<b>EMERGENCY_BRAKE</b>	Bahn leitet Notbremsvorgang ein
<b>CRASH</b>	Bahn verunfallt
<b>PLACE_LOAD_TRAIN</b>	Bahn leitet Passagierwechsel ein
<b>ADV_TRACK</b>	Bahn transferiert auf Folgleis des Modellgraphen
<b>RESERVE_SWITCH</b>	Bahn reserviert Weiche
<b>TRACK_BLOCK</b>	Blockierung eines Gleises
<b>TRACK_UNBLOCK</b>	Gleisblockierung aufheben
<b>TRAFFICLIGHT_SWITCH</b>	Umschalten in die nächste Ampelphase

*Tabelle 4.1: Von Lückemeyer in [58] identifizierte Ereignistypen*

Wie zu sehen, bildet Lückemeyer in seinem Modell das Fahrverhalten der Bahnen mikroskopisch mit Hilfe von Ereignistypen zum Stehen, Beschleunigen, Bewegen und Abbremsen ab. Dabei werden explizit auch Interaktionen zwischen auf demselben Gleis fahrenden Fahrzeugen abgebildet; das Fahrverhalten eines Vorgängerfahrzeugs beeinflusst also die Bewegung des Nachfolgerfahrzeugs, wobei Lü-

ckemeyer einige Vereinfachungen vornimmt: Bahn-Entitäten beschleunigen immer auf die für die aktuelle Verkehrssituation maximal mögliche Geschwindigkeit, um anschließend mit maximaler Kraft auf die notwendige Geschwindigkeit abzubremsen, anstatt weniger stark zu beschleunigen und anschließend mit konstanter Geschwindigkeit weiterzufahren. Das konkrete Fahrverhalten der Bahn-Entitäten in Lückemeyers Modell basiert auf Approximationen der Fahrverlaufskurven des Niederflur-Stadtbahnwagens K4000 der Vossloh Kiepe GmbH, wie sie in [98] angegeben sind. Anders als Joisten, modelliert Lückemeyer dabei kein Trödeln während des Fahrens, sondern nimmt an, dass alle Fahrer den gleichen Fahrstil aufweisen.

Wie Joisten, bildet Lückemeyer den Fahrplan über Umläufe ab, die den jeweiligen Fahrzeug-Entitäten zu Beginn der Simulation zugeordnet werden. Auch in Lückemeyers Modell werden dabei, aufgrund fehlender Daten, keine Leerfahrten abgebildet, sondern Fahrzeug-Entitäten nach Beendigung einer Fahrt aus dem Graphen entfernt und zu Beginn ihrer nächsten Fahrt am entsprechenden Haltepunkt-Knoten wieder hinzugefügt.

Erreichen Fahrzeug-Entitäten während ihrer Bewegung durch den Modellgraphen einen Weichen-Knoten, so müssen sie ihn vor der Überquerung reservieren und anschließend wieder freigeben. Ist ein Weichen-Knoten bereits durch eine andere Fahrzeug-Entität reserviert, müssen neu ankommende Entitäten vor ihm warten bis er durch die andere Entität wieder freigegeben wurde.

Ähnlich wie in Joistens Modell, wird der Passagierwechsel als stochastischer Prozess modelliert. Dabei wird zunächst mit Hilfe einer gleichverteilten Zufallszahl entschieden, ob die Mindesthaltezeit überschritten wird oder nicht. Falls dies der Fall ist, wird die zuzurechnende Überschusszeit auf Basis einer dreiecksverteilten Zufallszahl bestimmt. Zusätzlich wird die resultierende Haltezeit mit einem haltepunktspezifischen Beladungsfaktor multipliziert. Unabhängig von der so erhaltenen Haltezeit, verweilen Fahrzeug-Entitäten, ebenfalls wie beim Modell von Joisten, immer so lange am Haltepunkt-Knoten bis der im Fahrplan angegebene Abfahrtszeitpunkt erreicht wurde.

Auch Lückemeyer wendet sein Modell zur Validierung auf das Kölner Stadtbahnnetz aus dem Jahr 2001 an. Dabei ergibt sich, wie zu erwarten, aufgrund der feingranulareren Auflösung, im Vergleich zum Modell von Joisten, eine höhere Laufzeit. Mit ca. fünf Minuten für die Simulation eines kompletten Betriebstages fällt diese Erhöhung jedoch nicht sehr stark aus.

Die Anwendung auf das Kölner Stadtbahnnetz zeigt, dass das Simulationsmodell plausibles Verhalten aufweist und eine Mehrheit der Artefakte des Zellulären Automaten eliminiert. Leider zeigen sich dafür jedoch andere Anomalien. So wird nur selten ein Warten von Fahrzeug-Entitäten vor Ampeln beobachtet. Dies erklärt Lückemeyer durch den unrealistisch hohen Lookahead den Fahrzeug-Entitäten aufweisen und durch den sie ihr Fahrverhalten derart anpassen, dass sie Ampeln möglichst unter Grün erreichen.

Darüber hinaus zeigen sich auch bei Lückemeyers Modell unrealistisch hohe Verspätungen an den Haltepunkten der außerstädtischen Überlandstrecken, während die eigentlich vermuteten Flaschenhälse in der Innenstadt auch bei Ignorieren der Überlandhaltepunkte kaum Verspätung aufweisen. Die hohen Verspätungen auf den Überlandstrecken erklärt Lückemeyer durch eine in der Bahnspezifikation der K4000 zu gering ausgewiesene Höchstgeschwindigkeit von 80 km/h. Werden anstatt Verspätungen hingegen verpasste Verbindungen, d.h. Abweichungen zwischen der geplanten und simulierten Abfahrtsreihenfolge an Haltestellen, betrachtet, so zeigt sich erwartetes Verhalten.

Um die verbleibenden Artefakte zu beseitigen, wurde ein weiteres, erstmals durch Lückerath in [60] vorgestelltes, von Lückerath et al. in [62] sowie von Ullrich et al. in [90] verfeinertes und von Ullrich in [89] parallelisiertes, mikroskopisches, ereignisbasiertes Simulationsmodell entworfen.

Basis dieses Modells ist die Abbildung des Streckennetzes als dünn besetzter, gerichteter Graph, in dem Haltepunkte, Weichen und Gleise als Knoten modelliert werden und Kanten lediglich Nachbarschaftsrelationen zwischen diesen Knoten darstellen. Im Gegensatz zum Modell von Lückemeyer, werden Haltepunkte dabei mit expliziter geographischer Ausdehnung abgebildet, stellen also ganze Bahnsteige dar. In Anlehnung an Lückemeyers Modell werden Haltezeiten als haltepunkt- und fahrzeugtypspezifische stochastische Prozesse modelliert: Mit einer durch die Nutzerin zu spezifizierenden Wahrscheinlichkeit wird die fahrzeugtypspezifische Mindesthaltezeit – i.d.R. die Zeit zum Öffnen und Schließen der Fahrzeugtüren – überschritten. Wie hoch diese Überschreitung ausfällt wird dabei durch eine

haltepunktspezifische, dreiecksverteilte Zufallszahl bestimmt. Auch in Lückerraths Modell verweilen Fahrzeug-Entitäten stets so lange am aktuellen Haltepunkt-Knoten bis die durch den Fahrplan angegebene Abfahrtszeit erreicht wurde.

Weichen-Knoten stellen auch in Lückerraths Modell Punkte ohne geographische Ausdehnung dar, werden in Nullzeit überfahren und müssen vor dem Überfahren reserviert, sowie hinterher freigegeben werden. Anders als im Modell von Lückemeyer, kann die Weichenreservierung dabei nicht beliebig früh erfolgen. Stattdessen reservieren Fahrzeug-Entitäten einen Weichen-Knoten erst sobald sie sich nicht mehr weiter mit der aktuellen Geschwindigkeit fortbewegen können, d.h. unmittelbar vor dem letztmöglichen Zeitpunkt zu dem sie noch gefahrlos auf 0 km/h abbremmen können. Dadurch werden Situationen vermieden in denen eine langsame Fahrzeug-Entität eine schnellere Entität auf einem Parallelgleis vor einem zusammenführenden Weichen-Knoten blockiert, weil sie ihren zugehörigen Gleis-Knoten zuerst betreten hat.

Fahrzeuge bilden auch im Modell von Lückerrath transiente Entitäten, die sich durch den Modellgraphen bewegen. Anders als beim Modell von Lückemeyer weicht das Modell von Lückerrath aber vom klassischen ereignisbasierten Ansatz einer zentralen Ereignisverarbeitungsroutine ab. Dies ist möglich, da das Systemverhalten derart modelliert ist, dass Ereignisse immer nur eine einzige Entität betreffen und auch stets exakt ein Folgeereignis auslösen. Dadurch kann die Verarbeitung der Ereignisse durch die jeweils betroffene Entität erfolgen und in diese verlagert werden. Die zentrale Simulationsroutine verwaltet in der Folge nur noch die derzeit aktiven sowie zukünftig aktiv werdenden Fahrzeug-Entitäten und veranlasst zum Zeitpunkt ihres nächsten Ereignisses dessen Verarbeitung. Die von Lückerrath zur Modellierung identifizierten Ereignistypen zeigt Tabelle 4.2.

Ereignistyp	Beschreibung
<b>TRIP_START</b>	Bahn startet eine Fahrt
<b>TRIP_END</b>	Bahn beendet eine Fahrt
<b>TRAM_STANDING</b>	Bahn steht
<b>MOVEMENT_START</b>	Bewegung mit gleichbleibender Geschwindigkeit
<b>BRAKING_START</b>	Bahn leitet Bremsvorgang ein
<b>EMERGENCY_BRAKE</b>	Bahn leitet Notbremsvorgang ein
<b>CRASH</b>	Bahn verunfallt
<b>ACCELERATION_START</b>	Bahn leitet Beschleunigungsvorgang ein
<b>RESERVE_SWITCH</b>	Bahn reserviert Weiche
<b>FREE_SWITCH</b>	Bahn gibt Weiche frei
<b>RESERVE_BIDIRECTIONAL_TRACK</b>	Bahn reserviert eingleisigen Streckenabschnitt
<b>FREE_BIDIRECTIONAL_TRACK</b>	Bahn gibt eingleisigen Streckenabschnitt frei
<b>PASSENGER_EXCHANGE_START</b>	Bahn leitet Passagierwechsel ein
<b>TRANSFER_TO_NEXT_NODE</b>	Bahn transferiert zum nächsten Knoten des Modellgraphen

*Tabelle 4.2: Von Lückerrath in [60] identifizierte Ereignistypen*

Eine Folge des gewählten Modellierungsansatzes ist, dass bei der Abbildung der Fahrdynamik angenommen wird, dass Fahrer ein konservatives Fahrverhalten besitzen, d.h. sich so verhalten, dass sie in jeder Situation noch ausreichend Platz haben um notfalls zum Stehen zu kommen. Ein (weiterer) Vorteil dieses Modellierungsansatzes ist, dass das in Lückemeyers Modell nicht berücksichtigte Trödeln bei Fahrbewegungen wieder abgebildet werden kann. Dies geschieht, indem jeder Beschleunigungsvorgang mit einer durch die Nutzerin spezifizierbaren Wahrscheinlichkeit um einen, ebenfalls durch die Nutzerin spezifizierbaren, Zeitfaktor verlängert wird.

Wie Tabelle 4.2 zeigt, werden Ampelschaltung nicht über Ereignisse abgebildet. Da die durch ihre zugehörigen Gleis-Knoten verwalteten Ampeln deterministische Phasendauern aufweisen und sich lediglich durch ihren, zu Beginn der Simulation zufällig bestimmten, Zeitpunkt des ersten Phasenwechsels unterscheiden, können ihre aktuellen Zustände über eine einfache mathematische Funktion bestimmt

werden. Dadurch kann die Zahl der während eines Simulationslaufs zu verarbeitenden Ereignisse deutlich reduziert werden.<sup>17</sup>

Das Modell wird validiert, indem Plausibilitätsuntersuchungen anhand von Modellen diverser Iterationen des Stadtbahnnetzes von Köln (siehe [62], [89–91], [94]) als auch des Stadtbahnnetzes der südfranzösischen Stadt Montpellier (siehe [41], [89], [91], [93], [94]) durchgeführt werden. Dabei können die in Lückemeyers Modell auftretenden Simulationsartefakte, wie erhofft, weitgehend beseitigt und plausibles Modellverhalten nachgewiesen werden. Ein Vergleich mit Realdaten kann, wie bei Joisten und Lückemeyer, hingegen mangels fehlender Daten nicht erfolgen.

Ein aktuelles, alternatives Simulationsmodell für schienengebundenen Verkehr wird von Huang et al. in [45] vorgestellt. Die Autoren verwenden für ihr mikroskopisches, ereignisbasiertes Modell die *Discrete Event System Specification (DEVS)* und bilden das Streckennetz ebenfalls als gerichteten Graph ab. Dieser Graph besteht aus miteinander verknüpften Infrastrukturmodellen, die Gleise, Weichen, Sensoren oder Signale sein können. An diese Infrastrukturmodelle werden die Fahrzeugmodelle während des Simulationsverlaufs dynamisch gekoppelt. Der Bewegungsablauf der Fahrzeugmodelle wird mit Hilfe des *generalized DEVS (GDEVS)* Formalismus abgebildet, der die Modellierung kontinuierlicher und hybrider Systeme mit Hilfe diskreter Ereignisse unterstützt. Konkret wird die Bewegung von Fahrzeugen über stückweise polynomielle Funktionen abgebildet, zwischen denen zu diskreten Zeitpunkten, an denen sich die Beschleunigung verändert, gewechselt wird. Interaktionen zwischen Entitäten des Modells – sowohl Fahrzeug zu Fahrzeug als auch Fahrzeug zu Infrastruktur und Infrastruktur zu Infrastruktur – werden über Nachrichten modelliert. Ändert ein vorausfahrendes Fahrzeugmodell bspw. seine Beschleunigung, übermittelt es eine Nachricht an das Infrastrukturmodell an das es gekoppelt ist. Das Infrastrukturmodell reicht die Nachricht wiederum an ein eventuell nachfolgendes Fahrzeugmodell weiter. Entweder indem es das nächste an sich gekoppelte Fahrzeugmodell benachrichtigt oder indem es sein vorgelagertes Infrastrukturmodell benachrichtigt, das wiederum genauso verfährt. Dies geschieht so lange bis das nächste Fahrzeugmodell erreicht wurde. Huang et al. machen keine weiteren Angaben darüber, ob ihr Modell die Abbildung ganzer Fahrpläne, inklusive Umläufen, ermöglicht. Darüber hinaus, gehen die Autoren auch nicht weiter darauf ein, wie der Passagierwechsel in ihrem Modell abgebildet wird.

Ihr Modell wenden Huang et al. zur Simulation des Verkehrs auf einem (sehr) kurzen Tunnelabschnitt sowie des gesamten Stadtbahnnetzes von Den Haag an und vergleichen die resultierenden Laufzeiten mit denen eines kontinuierlichen DEVS-Modells. Dabei zeigen sie, dass ihr Modell einen vollständigen Betriebstag des Stadtbahnnetzes von den Haag auf einem „handelsüblichen“ Laptop innerhalb von ca. acht Minuten simulieren kann. Weitere Angaben zur Validität ihres Modells machen die Autoren nicht.

#### 4.4. Simulationsmodelle für multi-modalen Verkehr

Zur Abbildung (fast) vollständiger Verkehrssysteme werden multi-modale Simulationsmodelle verwendet. Modelle für multi-modalen Verkehr bilden häufig einzelne Reisende inklusive ihrer Bewegung mit und durch die unterschiedlichen Verkehrssysteme ab, vereinfachen aber unter Umständen einzelne Aspekte dieser Systeme (siehe z.B. [19], [52], [68], [85], [86]). Nagel skizziert in [68] den Modellierungsansatz des mikroskopischen Simulationsframeworks *TRANSIMS (Transportation Analysis and Simulation System)*, in dem, vergleichbar zum Modell von Maignan et al. in [67], das Gesamtverkehrsnetz in ein ÖPNV-Netzwerk, ein Fußgängernetzwerk und ein Individualverkehrsnetzwerk aufgeteilt wird, durch die sich die jeweiligen Verkehrsteilnehmer bewegen. Verbunden sind diese Netzwerke durch das Fußgängernetzwerk, in dem sich Transferpunkte zu den jeweils anderen Netzwerken finden, bspw. zu einem Haltepunkt des ÖPNV-Netzwerks oder zu einem Parkplatz im Individualverkehrsnetzwerk. Einzelne Reisende besitzen Tagespläne mit Aktivitäten, die wiederum an spezifischen Orten im Fußgängernetzwerk ausgeführt werden. Diese Orte erreichen die Reisenden dann durch eine Route durch die drei verlinkten Netzwerke.

---

<sup>17</sup> Zum Vergleich: Im Modell von Lückemeyer machen Ampelschaltungen 60 Prozent der Gesamtzahl der zu verarbeitenden Ereignisse aus.

Kendziorra und Weber führen in [52] eine ähnliche Erweiterung für SUMO ein, während Cats in [19] einen Ansatz zur Erweiterung von BusMezzo vorstellt. Sowohl Kendziorra und Weber als auch Cats unterscheiden dabei das Verkehrsnetz nicht weiter, sondern modellieren ein Gesamtnetzwerk durch das sich alle Verkehrsteilnehmer bewegen. Die von Cats vorgestellte Erweiterung von BusMezzo bildet dabei keine fußläufigen Routen ab, Reisende bewegen sich also entweder per Individualverkehr oder mit Hilfe von Fahrzeugen des ÖPNV.

Ein aktuelles, sehr ambitioniertes, multi-modales Simulationsmodell namens *M3* wird von Suzumura und Kanezashi in [85] sowie Suzumura et al. in [86] vorgestellt. Die Autoren beschreiben ein verteiltes, paralleles, agentenbasiertes Simulationsmodell auf Basis des *IBM Mega Traffic Simulator (Megaffic*, siehe [72]). In diesem Modell werden alle Verkehrsteilnehmer, seien es Fußgänger, PKWs, Busse, Bahnen, Fahrräder, etc., als Agenten modelliert, die sich über, z.T. gemeinsame, Graphenmodelle ihrer jeweiligen Verkehrsnetze bewegen. Während die Bewegung von Fahrzeugen des ÖPNV auf Basis von Linienverläufen und Fahrplänen erfolgen, werden Reisen einzelner Personen auf Grundlage von Quell/Ziel-Matrizen modelliert. Zu entsprechenden Simulationszeitpunkten werden neue *Personen-Agenten* erzeugt, denen über ein zentrales Routenwahl-Modul eine multi-modale Route zwischen ihrem Start- und Zielort zugewiesen wird. Die Bewegung mit Hilfe der unterschiedlichen Verkehrsmodi wird durch *Transport-Agenten* abgebildet, die zu Beginn einer Teilroute erzeugt werden und an die der entsprechende Personen-Agent bis zum Ende der Teilroute gekoppelt wird. Ist die Teilroute beendet, wird der Transport-Agent aus dem System entfernt. Ausgenommen hiervon sind Fahrzeuge des ÖPNV; ihre entsprechenden Transport-Agenten werden zu den im Fahrplan angegebenen Abfahrtszeiten erzeugt und erst am Ende ihrer Fahrt aus dem System entfernt. Um an Transport-Agenten des ÖPNV gekoppelt zu werden, muss sich ein Personen-Agent an einem Haltepunkt befinden und so lange dort warten, bis ein ÖPNV-Agent der in seiner Gesamtroute angegebenen Linie mit ausreichend Kapazität ankommt. Der Halteprozess selber scheint in *M3* nicht stochastisch modelliert zu sein. Ein Fahrzeug-Agent des ÖPNV wartet laut den Autoren lediglich so lange an seinem aktuellen Haltepunkt, bis zur im Fahrplan angegebenen Abfahrtszeit. Wie sich der Halteprozess verhält, wenn ein Fahrzeug-Agent über seinen geplanten Abfahrtszeitpunkt hinaus verspätet ist, wird von den Autoren hingegen nicht erwähnt.

Während Bus-Agenten im Modell von Suzumura et al. das gleiche grundlegende Fahrverhalten wie PKW-Agenten aufweisen – d.h. ihr Fahrverhalten aufgrund eines Fahrzeugfolge- und Spurwechsel-Modells anpassen – wird das Fahrverhalten von Bahnen stark vereinfacht abgebildet: Es werden keine Interaktionen zwischen auf dem gleichen Gleis fahrenden Fahrzeug-Agenten abgebildet und die Fahrzeit zwischen zwei Haltepunkten ist deterministisch und entspricht der im Fahrplan ausgewiesenen Zeit.

Zur Validierung ihres Modells wenden Suzumura et al. es in [86] zur Simulation des Verkehrsnetzes der irischen Stadt Dublin an. Das Netz umfasst 61.341 Straßenabschnitte, 19.261 Kreuzungen, 4.700 Haltepunkte für Busse mit 121 unterschiedlichen Routen, 54 Haltepunkte für Stadt- bzw. Straßenbahnen mit 2 Routen und ca. 50 Haltepunkte für Bahnen des Fernverkehrs mit ungefähr 16 Routen. Für diesen Anwendungsfall simulieren die Autoren ein vierstündiges Zeitfenster von sechs bis zehn Uhr Morgens und vergleichen die simulierten Reisezeiten mit den real gemessenen. Dabei stellen die Autoren fest, dass die simulierten Reisezeiten für Fahrzeuge des Individualverkehrs kürzer ausfallen als die beobachteten. Sie führen dies unter anderem darauf zurück, dass ihr Simulator zur Bestimmung von (Teil-)Reisepfaden für die keine Angaben in den Quell/Ziel-Matrizen der Input-Daten vorhanden sind, stets die Route wählt, die die kürzeste Reisezeit aufweist. Die simulierten Reisezeiten der Fahrzeuge des Busverkehrs zeigen hingegen eine bessere Übereinstimmung, was die Autoren darauf zurückführen, dass Busse in der Simulation stets bis zur geplanten Abfahrtszeit an den Haltepunkten warten.

Allen in diesem Abschnitt besprochenen Simulationsmodellen ist gemein, dass sie dazu herangezogen werden können, um die Performanz von Verkehrssystemen als Ganzes, bspw. auf Basis von Reisezeiten und Verkehrsflüssen, zu bewerten. Allerdings scheint keines der genannten Modelle die detaillierte Bewertung von (multi-modalen) ÖPNV-Systemen zum Ziel zu haben, weswegen sie nur bedingt zur Bewertung von Linienführung und Fahrplänen herangezogen werden können.

## 5. Ein neues Simulationsmodell für den Öffentlichen Personennahverkehr

### 5.1. Modell

Im Folgenden wird ein modulares, mesoskopisches Simulationsmodell vorgestellt, das sowohl für mono- als auch multi-modale (Nah-)Verkehrssysteme geeignet ist und auf dem bereits in Abschnitt 4.1.2.2 beschriebenen ereignisbasierten Paradigma basiert. Abweichend zum dort vorgestellten Standardansatz bei dem Ereignisse und damit potentielle Zustandsänderungen von Entitäten in einer zentralen Routine verarbeitet werden (vgl. Abbildung 4.3), ist die Ereignisverarbeitung im hier vorgestellten Modell dezentral organisiert: Die in das Modell eingebundenen Entitäten können als Ziele von Ereignissen ausgewählt werden und führen die sich für sie aus einem spezifischen Ereignis ergebenden Aktivitäten selbstständig aus. Hierdurch wird eine Kapselung der die einzelnen Entitäten betreffenden Simulationslogik und damit eine stärkere Modularisierung und einfachere Erweiterbarkeit des Simulationsmodells erreicht. Die sich dadurch ergebenden Änderungen an der Ereignisverarbeitungsroutine zeigt Abbildung 5.1.

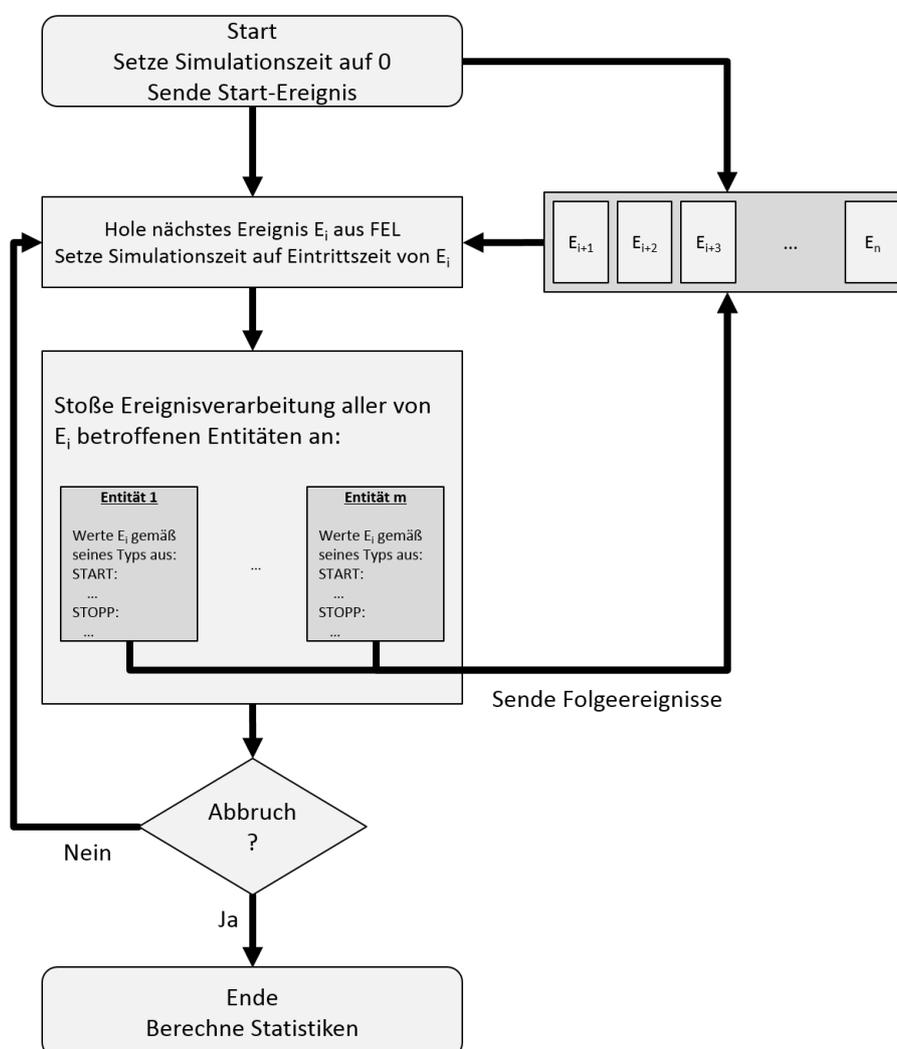


Abbildung 5.1: Modifizierte Ereignisverarbeitungsroutine

Die Mehrheit der im vorhergehenden Abschnitt beschriebenen Simulationsansätze – insbesondere die Modelle für Bus- und Individualverkehr – stellen die Modellierung eines Individualverkehrssystems in den Mittelpunkt und integrieren Nahverkehrssysteme erst in einem nachgelagerten Schritt auf Basis dieser Modellierungsentscheidungen. Als Folge dessen fällt die Abbildung (von Teilen) dieser Systeme

häufig unnötig komplex (vgl. Meignan et al. in [67]) aus oder benötigt bei Anwendungsbeispielen realistischer Größe eine nur schwer zu akzeptierende Laufzeit (vgl. bspw. die Ausführungen zu SUMO im vorhergehenden Kapitel). Aus diesem Grund verfolgt das hier beschriebene Modell den umgekehrten Ansatz und stellt die abzubildenden Nahverkehrssysteme in den Mittelpunkt aller Modellierungsentscheidungen. Das Modell soll dabei jedoch so gestaltet werden, dass eine zukünftige Erweiterung um die detaillierte Abbildung von Individualverkehr, im Kontext des Nahverkehrssystems, möglich ist.

Da die im Rahmen dieser Arbeit verfügbare Datengrundlage, wie bei den vorangegangenen Arbeiten im Rahmen des CATS-Projekts, nur als spärlich zu bezeichnen ist, muss das Simulationsmodell darüber hinaus so gestaltet sein, dass es dennoch plausibles Verhalten aufweist. Zu diesem Zweck wird in der hier vorgestellten Version des Modells das Fahrverhalten einzelner Fahrzeuge stark abstrahiert, d.h. mesoskopisch, abgebildet. Die zukünftige Einbindung umfangreicherer Datensätze wird allerdings auch bei diesem Modellierungsaspekt bereits berücksichtigt.

Um diese Ziele zu gewährleisten, ist das Simulationsmodell stark modularisiert und die Mehrzahl der mit empirischen Daten arbeitenden Modellbestandteile so ausgelegt, dass sie auch mit feingranulareren statistischen Daten arbeiten können. Stehen empirische Daten nicht oder nur in einer groben Auflösung zur Verfügung, greifen die Modellkomponenten hingegen auf durch die Anwenderin parametrisierbare allgemeine Verteilungsfunktionen zurück.

### 5.1.1. Das physische Netz

Basis des Modells ist, wie beim vom Autor in [60] entworfenen Simulationsmodell für Stadtbahnssysteme, die Abbildung des physischen Verkehrsnetzes als gerichteter Graph. Haltepunkte, Verbindungen und Segmente werden als Knoten dieses Graphen modelliert, die über Nachbarschaftsrelationen in Beziehung zueinander stehen. Jeder dieser Knoten besitzt eine geographische Position, ihn identifizierende Attribute (Bezeichner, Identifikationsnummer, etc.), sowie eine maximale Fahrzeugkapazität. Darüber hinaus stellt jeder Knoten auch eine Entität im Sinne des ereignisbasierten Simulationsparadigmas dar, kann also Quelle und Ziel von Ereignissen sein. So können temporäre Änderungen von Attributsausprägungen, etwa zur Modellierung von Störungen, auf einfache Weise über Ereignissen und Aktivitäten abgebildet werden (für eine detaillierte Ausführung siehe Abschnitt 7.1.1).

*Haltepunkte* sind Orte im Verkehrssystem an denen Fahrzeuge des ÖPNV halten, um Passagieren den Ein- und Ausstieg zu ermöglichen. Sie gehören immer zu exakt einer Haltestelle und besitzen tageszeit- und ortsspezifische Haltezeiten. Die zu ihnen korrespondierenden Knoten des Modellgraphen verwalten entsprechend sowohl einen Verweis auf die die zugehörige Haltestelle abbildende Modellkomponente als auch tageszeit- und ortsspezifische empirische Daten zu Haltezeiten (siehe dazu auch Abschnitt 5.1.5.2). Unterschiedliche räumliche Abmessungen von Haltepunkten werden mit Hilfe der maximalen Fahrzeugkapazität von Knoten abgebildet. So können an einem größeren Haltepunkt innerhalb eines Busbahnhofs in der Regel mehrere Fahrzeuge gleichzeitig halten. Wohingegen an einem Haltepunkt ohne gesonderte Haltebuch an einer vielbefahrenen Straße ein ankommendes Fahrzeug unter Umständen zunächst auf ein bereits haltendes Fahrzeug warten muss bevor es an den Haltepunkt heranfahren kann.

*Haltestellen* gruppieren geographisch zusammengehörende Haltepunkte und geben diesen einen einheitlichen Namen. Sie bilden eine zusätzliche Informationsschicht innerhalb des Modells, werden entsprechend zwar vom Graphen verwaltet, sind aber nicht als Knoten sondern als gesonderte Datenstruktur modelliert.

*Verbindungen* sind gerichtete Wege durch das Verkehrsnetz die zwei Haltepunkte miteinander verbinden. Sie besitzen sowohl eine spezifische Länge als auch tageszeit- und ortsspezifische durchschnittliche Fahrzeiten. Darüber hinaus wird ihnen durch den Fahrplan eine geplante Fahrzeit zugewiesen. Die sie abbildenden Knoten des Modellgraphen verwalten folglich Angaben über ihre Länge, der im Fahrplan für sie angegebenen geplanten Fahrzeit, sowie tageszeit- und ortsspezifische empirische Daten zu durchschnittlichen Fahrzeiten (siehe dazu auch Abschnitt 5.1.5.1). Abhängig vom zu modellierenden Verkehrssystem und dem Detailgrad der vorliegenden Datenbasis verwalten Verbindungs-Knoten darüber hinaus auch Modellkomponenten der zu ihrer Verbindung gehörenden Segmente, Weichen und Signale.



Soweit vorhanden, wird im Modell zur Bestimmung der Fahrzeit einer neu auf einem Gleis-Knoten ankommenden Fahrzeug-Entität immer auch die direkt vorausfahrende Entität berücksichtigt. Der berechnete Simulationszeitpunkt zu dem die neue Fahrzeug-Entität am Ende des Gleis-Knotens ankommt darf dabei nie früher als der der direkten Vorgängerentität sein (siehe auch hier Abschnitte 5.1.2 und 5.1.5.1). Ohne Möglichkeiten für Überholmanöver muss die Bildung von Rückstau – auch über benachbarte Knoten hinweg – im Modell berücksichtigt werden. Dies geschieht mit Hilfe der Fahrzeugkapazität von Knoten. Im Gegensatz zu Haltepunkt-Knoten wird bei Gleis-Knoten dazu deren Länge herangezogen: Die kombinierte Länge aller zu einem gegebenen Zeitpunkt auf dem Gleis-Knoten verkehrender Fahrzeug-Entitäten muss stets kleiner oder gleich der Gleislänge sein. Würde die kombinierte Fahrzeuglänge durch eine neu auf dem Gleis-Knoten ankommende Entität größer als die Gleislänge werden, muss die ankommende Fahrzeug-Entität auf dem dem Gleis-Knoten vorgelagerten Nachbarknoten warten bis ausreichend Platz vorhanden ist (siehe dazu auch Abschnitt 5.1.2).

*Weichen* sind Orte in Schienenverkehrssystemen an denen Gleisübergänge ohne Fahrtunterbrechung stattfinden, besitzen also eine eindeutige geographische Position und stehen mit mindestens zwei Gleisen – einem eingehenden und einem ausgehenden – in Beziehung. Sie werden als Transferpunkte ohne räumliche Ausdehnung modelliert und in Nullzeit überfahren. Weichen können mehrere Gleise zusammenführen und müssen vor dem Überfahren entsprechend gestellt werden. Dies wird im Modell abgebildet, indem Weichen vor dem Überfahren durch Fahrzeug-Entitäten reserviert und nach erfolgreichem Transfer wieder freigegeben werden müssen. Wie Haltestellen bilden Weichen lediglich eine weitere Informationsschicht des Modells und sind nicht als Knoten modelliert, sondern mit Hilfe einer eigenständigen Datenstruktur.

*Signale* stellen Ampeln des Straßenverkehrs sowie Lichtsignalanlagen des Schienenverkehrs dar. Sie bilden in der Regel eine Signalgruppe mit anderen Signalen und besitzen Attribute wie Schaltzeit oder Signalstatus (z.B. Grün, Gelb, Rot). Wie Haltestellen und Weichen, bilden auch sie eine zusätzliche Informationsschicht und sind nicht als Knoten des Graphen, sondern eigenständige Datenstruktur modelliert.

Sowohl Weichen als auch Signale sind im Modell immer nur an Start- oder Endpositionen von Segment-Knoten zu finden und können anhand diese Positionen mit den entsprechenden Knoten in Verbindung gebracht werden. Um Entitäten der Simulation einen einfacheren Zugriff auf sie zu gewähren, werden sie über die ihnen übergeordneten Verbindungs-Knoten verwaltet.

Abbildung 5.2 zeigt eine stark verkürzte Version des sich ergebenden UML-Klassendiagramms für das Modell des physischen Netzes.

### 5.1.2. Fahrzeuge

Fahrzeuge werden als transiente Entitäten des Modells abgebildet, die einen signifikanten Teil der ereignisbasierten Simulationslogik kapseln und sich während eines Simulationslaufs über den Modell-Graphen bewegen. Als transiente Entitäten des Modells halten sie zu jedem Zeitpunkt einen Verweis auf den aktuell durch sie belegten Knoten des Graphen, d.h. ihre derzeitige Position. Außerdem besitzt jede Fahrzeug-Entität einen Verweis auf die aktuell von ihr bediente Fahrt, hat also zu jedem Simulationszeitpunkt stets nur Zugriff auf die unmittelbar für ihre derzeitige Tätigkeit relevanten Informationen. Alle darüberhinausgehenden Informationen, wie z.B. zum Fahr- und Umlaufplan oder zur Fahrzeugflotte werden über spezialisierte Verwaltungsmodule administriert (siehe dazu Abschnitt 5.1.4).

Fahrzeuge werden im Modell anhand ihrer Verkehrsart, ihres Fahrzeugtyps und ihrer konkreten Fahrzeugeigenschaften klassifiziert. Während über die *Verkehrsart* lediglich eine grobe Kategorisierung in z.B. Bus-, Bahn- oder Individualverkehr vorgenommen wird, wird mittels des *Fahrzeugstyps* eine detailliertere Unterteilung der einzelnen Verkehrstypen vorgenommen. So sind im Kölner Stadtbahnnetz unterschiedliche Fahrzeugtypen der Vossloh Kiepe GmbH im Einsatz, darunter unter anderem Niederflurfahrzeuge vom Typ K4000 und K4500 (siehe [98], [100]) als auch Hochflurfahrzeuge vom Typ K5000 (siehe [99]). Die detaillierteste Klassifizierung findet anhand der *Fahrzeugeigenschaften* statt. Sie beziehen sich immer auf individuelle Fahrzeug-Entitäten und kapseln Attribute wie Passagierkapazität, Fahrzeuglänge, Höchstgeschwindigkeit, minimale Haltezeit oder Einstiegsrate (siehe zu Haltezeit

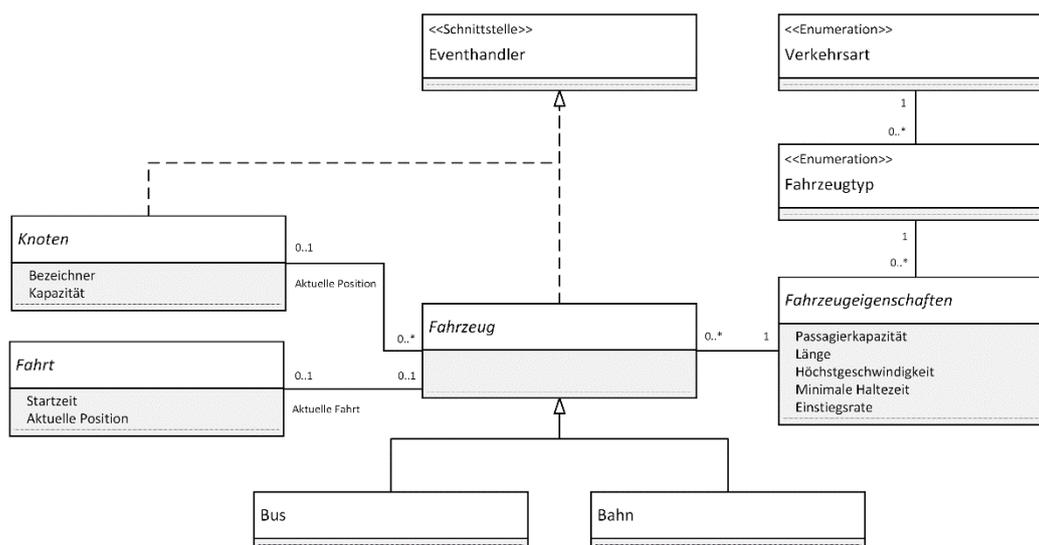


Abbildung 5.3: Verkürztes UML-Klassendiagramm für Fahrzeuge

und Einstiegsrate auch Abschnitt 5.1.5.2). Diese Art der Modellierung ermöglicht zum einen die Abbildung von Fahrzeug-Entitäten des gleichen Typs mit unterschiedlicher Ausstattung, zum anderen können dadurch auch Störungen individueller Fahrzeuge (z.B. eine erhöhte minimale Haltezeit aufgrund einer defekten Tür) abgebildet werden. Abbildung 5.3 verdeutlicht das sich ergebende Fahrzeugmodell anhand eines verkürzten UML-Klassendiagramms.

Die zum Zeitpunkt der Anfertigung dieser Arbeit vorliegende Version des Simulationsmodells umfasst generische Fahrzeugmodelle für Busse sowie Stadtbahnen. Auf eine Modellierung von Fahrzeugen des Individualverkehrs wurde mangels belastbarer Daten verzichtet. Eine spätere Erweiterung des Simulationsmodells um weitere Verkehrsmodi ist aber auf einfache Weise möglich, indem neue, vom allgemeinen Fahrzeugmodell abgeleitete, Fahrzeugmodelle erzeugt und mit individuellen Fahrzeugeigenschaften versehen werden.

Zur Abbildung des Verhaltens von Fahrzeugen des Bus- und Stadtbahnverkehrs wird zwischen neun verschiedene Simulationsereignistypen unterschieden (siehe Tabelle 5.1). Abbildung 5.4 und Abbildung 5.5 zeigen die zwischen den einzelnen Ereignis- und Aktivitätstypen bestehenden Zusammenhänge für Bus- bzw. Bahnfahrzeuge anhand der zugehörigen Ereignisprozessketten.

Ereignistyp	Beschreibung
<b>ROTATION_START</b>	Das Fahrzeug beginnt seinen Dienst für den Betriebstag
<b>ROTATION_END</b>	Das Fahrzeug beendet seinen Dienst für den Betriebstag
<b>DEADHEAD_START</b>	Das Fahrzeug beginnt eine Leerfahrt
<b>SERVICE_TRIP_START</b>	Das Fahrzeug beginnt eine Betriebsfahrt
<b>SERVICE_TRIP_END</b>	Das Fahrzeug beendet eine Betriebsfahrt
<b>BOARDING_START</b>	Das Fahrzeug beginnt mit dem Passagierwechsel
<b>BOARDING_END</b>	Das Fahrzeug beendet den Passagierwechsel
<b>DRIVING_START</b>	Das Fahrzeug (versucht) eine Fahrt auf einer Verbindung oder einem Segment zu beginnen
<b>DRIVING_END</b>	Das Fahrzeug (versucht) eine Fahrt auf einer Verbindung oder einem Segment zu beenden

Tabelle 5.1: Simulationsereignistypen zur Simulation von Fahrzeugen des ÖPNV

Die Ereignistypen ROTATION\_START und ROTATION\_END behandeln Betriebsbeginn bzw. -ende und die sich für Fahrzeug-Entitäten damit ergebenden Tätigkeiten. Dazu gehört unter anderem die Mitteilung der Fahrzeug-(De)Aktivierung an alle interessierten Entitäten des Simulationsmodells (wie z.B. das im nächsten Abschnitt beschriebene Fahrzeugverwaltungs-Modul). Der Ereignistyp ROTATION\_END bildet in der Regel das Ende des planmäßigen Diensts eines Fahrzeugs ab. Nach Verarbeitung der verbundenen Aktivitäten kann die betroffene Entität aber im Rahmen unplanmäßiger Sonder-einsätze (z.B. zur Ersetzung eines defekten Fahrzeugs) erneut verwendet werden.

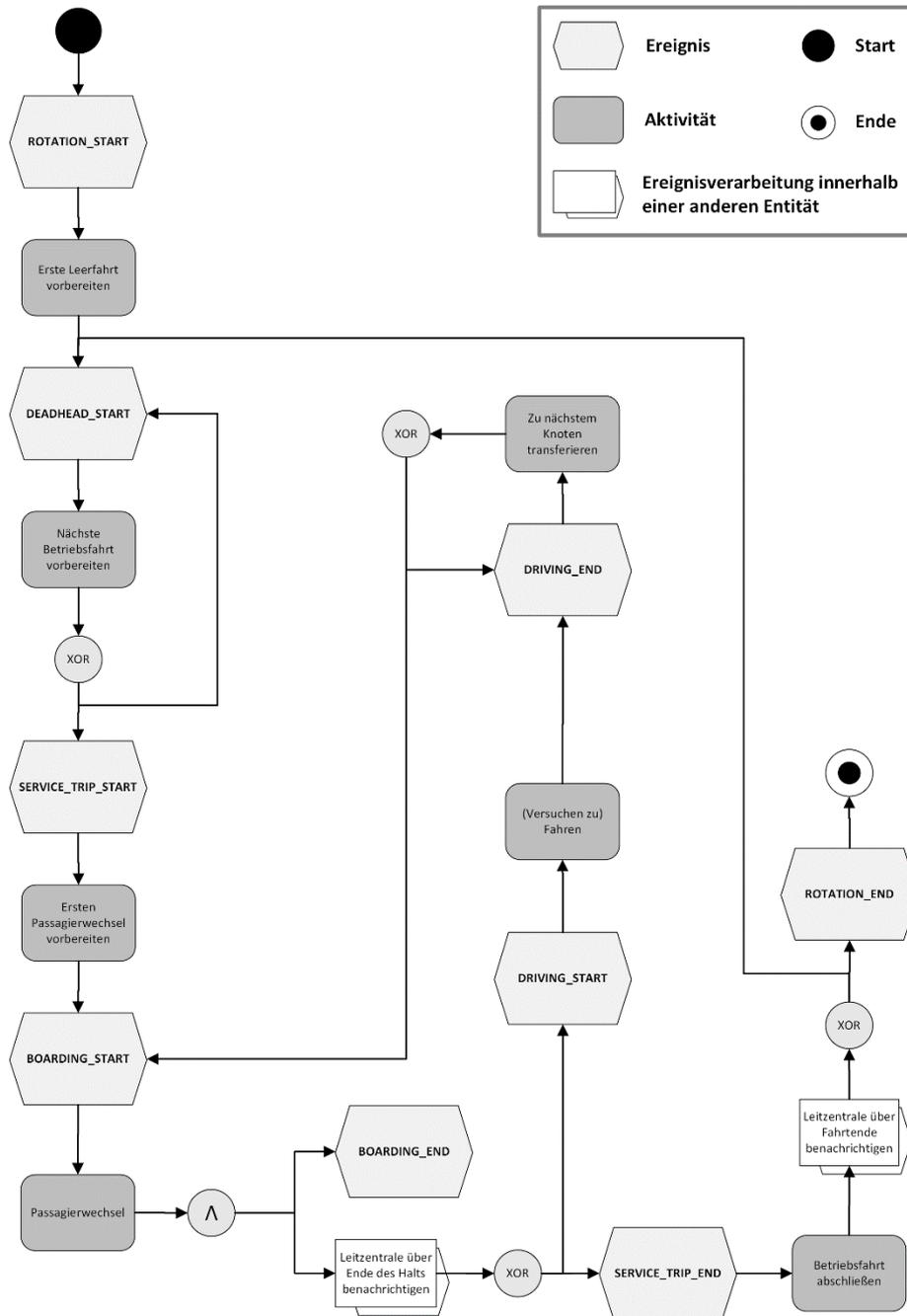
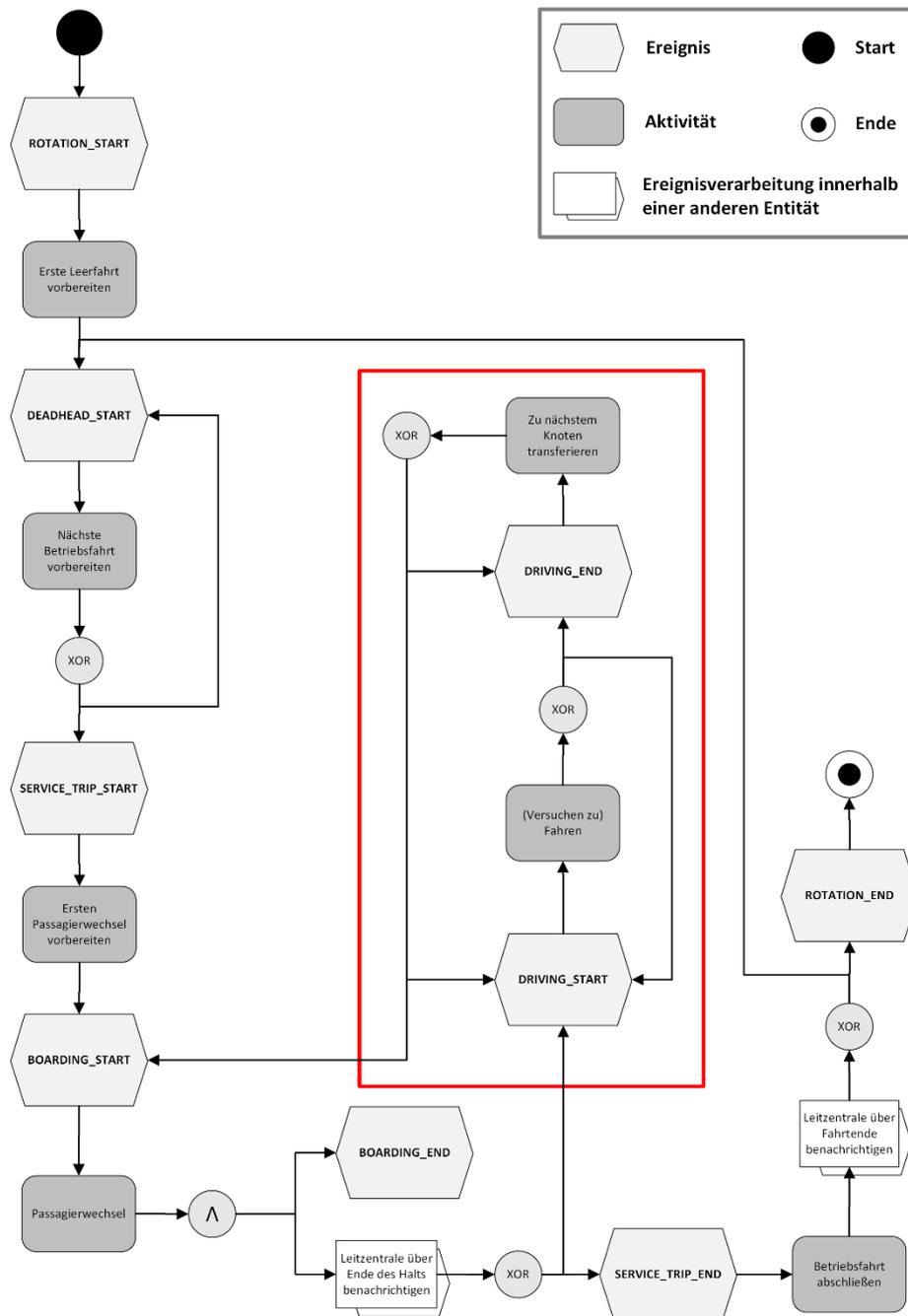


Abbildung 5.4: Ereignisprozesskette für Bus-Entitäten

Der Ereignistyp DEADHEAD\_START bildet den Beginn einer Leerfahrt ab. Dies sind Fahrten die in der Regel zwischen zwei Betriebsfahrten sowie zu Beginn und Ende des Dienstes stattfinden. Auf ihnen werden keine Passagiere befördert, sondern sie dienen lediglich dem Standortwechsel eines Fahrzeugs, z.B. um zu Dienstbeginn vom Depot zum ersten Haltepunkt der ersten Betriebsfahrt zu gelangen. Da die dem Simulationsmodell zugrunde liegende Datenbasis keine belastbaren Aussagen über Leerfahrten zulässt, werden die zugehörigen Aktivitäten, wie bei den Modellen von Joisten, Lückemeyer und Lückerkath, stark vereinfacht modelliert: Fahrzeug-Entitäten führen im Modell keine tatsächliche Leerfahrt durch, sondern unternehmen lediglich den Versuch Zugriff auf den ersten Haltepunkt-Knoten der nächsten Betriebsfahrt zu erhalten ohne dabei einen physischen Ort im simulierten Netzwerk zu belegen. Hat der Haltepunkt-Knoten zu diesem Zeitpunkt bereits seine maximale Kapazität erreicht, wird ermittelt wann die nächste Fahrzeug-Entität den Knoten verlassen wird und zu diesem Zeitpunkt ein neuer Versuch unternommen, d.h. ein neues Ereignis von Typ DEADHEAD\_START terminiert. Diese Form der Modellierung vernachlässigt allerdings im allgemeinen Effekte die aufgrund von Leerfahrten auftreten



**Abbildung 5.5: Ereignisprozesskette für Bahn-Entitäten. Rot umrandet: Unterschiedliche Abfolge im Vergleich zu Bus-Entitäten**

können, wie z.B. Verspätungen die von Fahrzeugen auf Leerfahrten zwischen zwei Haltepunkten eingefahren werden. Dem wird entgegengewirkt, indem bei der Erzeugung des Umlaufplans durch das Fahrzeugverwaltungsmodul (siehe Abschnitt 5.1.4.1) nur Leerfahrten zwischen Haltepunkt-Knoten der gleichen Haltestelle zugelassen werden. Aufgrund der geographischen Nähe zwischen den korrespondierenden Haltepunkten kann dabei angenommen werden, dass während des Regelbetriebs keine signifikanten Verspätungen auf diesen Fahrten auftreten. Von dieser Regelung ausgenommen ist die erste Leerfahrt einer Fahrzeug-Entität, die in der Regel vom Depot zum ersten Haltepunkt-Knoten der ersten Betriebsfahrt führt. Für diese Leerfahrt kann aber wiederum angenommen werden, dass das Depot rechtzeitig genug vor Beginn der ersten Betriebsfahrt verlassen wird, so dass sich eine auf der Leerfahrt auftretende Verspätung nicht signifikant auf den Beginn der Betriebsfahrt auswirkt.

Ist eine detaillierte Abbildung von Leerfahrten notwendig, müssen Modifikationen der Ereignisverarbeitungsroutinen der entsprechenden Entitäten erfolgen. Alle dazu notwendigen Bausteine sind bereits

im Simulationsmodell vorhanden, werden derzeit aber nicht oder nicht im notwendigen Ausmaß verwendet (siehe dazu auch Abschnitt 5.1.3).

Die Ereignistypen `SERVICE_TRIP_START` und `SERVICE_TRIP_END` bilden Beginn und Ende von Betriebsfahrten ab. Dies sind durch den Fahrplan festgelegte Fahrten zur Beförderung von Passagieren, die der durch den Linienverlauf vorgegebenen Reihenfolge von Haltepunkten und Verbindungen folgen. Während die durch `SERVICE_TRIP_START` ausgelösten Aktivitäten vornehmlich die Vorbereitungen für den ersten Passagierwechsel umfassen, dienen die `SERVICE_TRIP_END` folgenden Aktivitäten vor allem der Benachrichtigung relevanter Verwaltungsmodule, um z.B. kenntlich zu machen, dass die betroffene Fahrzeug-Entität zur Bedienung ihrer nächsten Fahrt bereit ist (siehe dazu auch Abschnitt 5.1.4).

Die Ereignistypen `BOARDING_START` und `BOARDING_END` behandeln Beginn und Ende des Passagierwechsels an Haltepunkten. Die auf `BOARDING_START` folgende Aktivität dient der Kalkulation der fahrzeug-, haltepunkt- und tageszeitspezifischen notwendigen Zeitdauer für den reinen Passagierwechsel (siehe dazu auch Abschnitt 5.1.5) und mündet in einem Ereignis vom Typ `BOARDING_END`, das sowohl die auslösende Fahrzeug-Entität als auch relevante Verwaltungsmodule als Zielentitäten besitzt. Im Rahmen der auf `BOARDING_END` folgenden Aktivität aktualisiert die betroffene Fahrzeug-Entität dabei lediglich ihren Status und löst kein Folgeereignis aus (vgl. Abbildung 5.4 und Abbildung 5.5). Stattdessen wird das nächste für die Entität relevante Ereignis nach der Verarbeitung von `BOARDING_END` durch die Verwaltungsmodule ausgelöst. Sie entscheiden dabei sowohl über die anzuwendende Haltestrategie (z.B. sofortige Weiterfahrt oder Verlängerung des Halts zur Sicherung von Umsteigeverbindungen), als auch über die (eventuell frühzeitige) Beendigung der aktuellen Fahrt (siehe dazu auch Abschnitte 5.1.4.3 und 7.2.1).

Ist der Passagierwechsel beendet und durch die Verwaltungsmodule die endgültige Abfahrtszeit festgelegt, wird – falls die Fahrt nicht (frühzeitig) beendet wird – die Fahraktivität angestoßen. Sie wird durch die Ereignistypen `DRIVING_START` und `DRIVING_END` eingerahmt und unterscheidet sich in einigen Bereichen für Bus- und Bahn-Entitäten: Zum einen führen Busse im Simulationsmodell ihre Fahraktivitäten auf Verbindungen aus, während Bahnen die einzelnen Segmente einer Verbindung befahren. Zum anderen müssen Bahn-Entitäten vor Gleisübergängen zwischenliegende Weichen reservieren. Zuletzt unterscheiden sich die Fahraktivitäten, wie bereits in Abschnitt 5.1.1 ausgeführt, auch anhand der durch die jeweiligen Knoten forcierten Randbedingungen bezüglich der Fahrzeugreihenfolge und der maximalen Fahrzeugkapazität des jeweiligen Knotens. Die sich daraus ergebende abweichende Verkettung zwischen Ereignissen und Aktivitäten für Bahn-Entitäten ist im umrandeten Bereich von Abbildung 5.5 zu sehen. Unmittelbar einsichtig ist, dass nach Abschluss der Fahraktivität auf einem Segment-Knoten, eine weitere Fahraktivität auf einem nachfolgenden Segment-Knoten notwendig sein kann (siehe Pfeil zwischen Aktivität *Zu nächstem Knoten transferieren* und Ereignis `DRIVING_START`). Weniger offensichtlich ist die Verbindung zwischen der Aktivität (*Versuchen zu Fahren*) und dem Ereignis `DRIVING_START`. Sie bildet einen aufgrund mangelnder Kapazität fehlgeschlagenen Versuch ab von einem Haltepunkt-Knoten auf einen nachgelagerten Gleis-Knoten zu transferieren. In diesem Fall muss die Fahrzeug-Entität auf dem Haltepunkt-Knoten verweilen, den Zeitpunkt ermitteln zu dem die nächste Entität den Zielknoten verlässt und zu diesem Zeitpunkt erneut versuchen die Fahraktivität anzustoßen.

Unabhängig davon ob sich Entitäten auf Verbindungs- oder Segment-Knoten fortbewegen wird der Fahrvorgang, den Grenzen der verfügbaren Datenbasis Rechnung tragend, stark abstrahiert simuliert: Findet sich ausreichend Platz, so verweilt die Fahrzeug-Entität für eine von diversen Faktoren wie z.B. der Tageszeit, dem Ort und den bereits auf dem Knoten befindlichen Entitäten abhängigen, Zeitdauer auf dem Knoten des Graphen (zur genauen Berechnung siehe Abschnitt 5.1.5.1). Mit Ablauf dieser Zeitdauer hat die Fahrzeug-Entität ihren Zielknoten erreicht und versucht Zugriff auf diesen zu erhalten. Besitzt der Zielknoten keine ausreichende Kapazität mehr, kann die Entität nicht auf ihn transferieren. In diesem Fall wird ermittelt wann die nächste der bereits auf dem Knoten befindlichen Entitäten diesen verlassen wird und zu diesem Zeitpunkt erneut versucht Zugriff zu erhalten, indem ein neues Ereignis vom Typ `DRIVING_END` terminiert wird (siehe Pfeil zwischen Aktivität *Zu nächstem Knoten transferieren* und Ereignis `DRIVING_END` in Abbildung 5.4 und Abbildung 5.5).

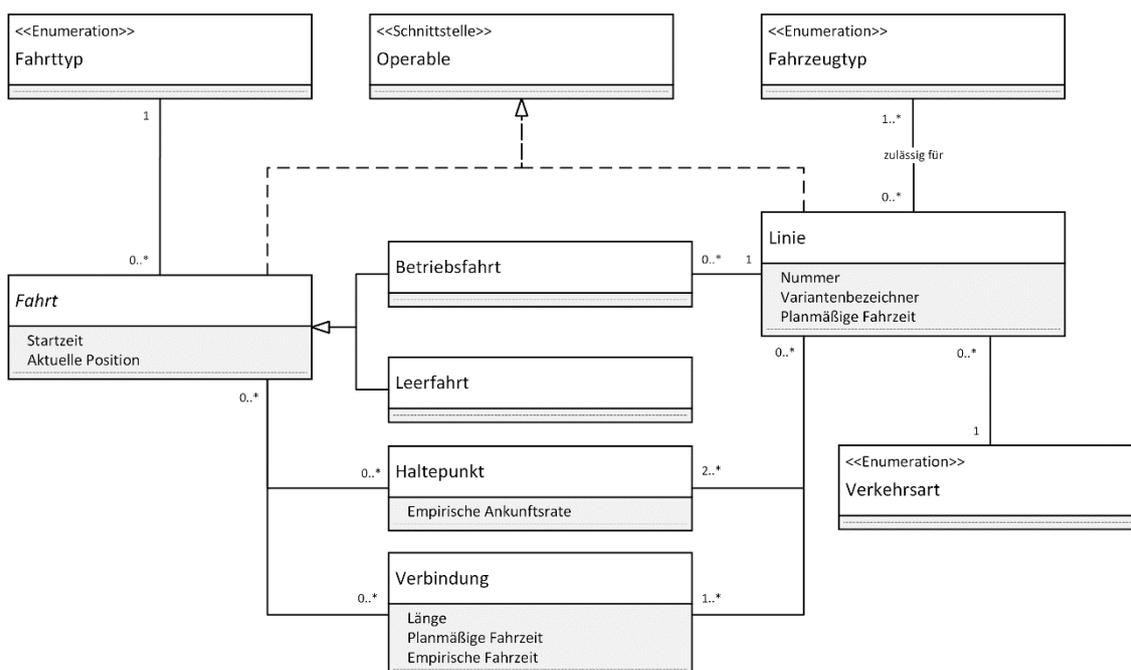


Abbildung 5.6: Verkürztes UML-Klassendiagramm für Fahrten und Linien

Ist eine umfangreichere Datenbasis gegeben und wird eine genauere Abbildung des Fahrverhaltens gewünscht, kann die Fahraktivität beispielsweise in eine beliebige Anzahl kleinerer, miteinander verknüpfter Aktivitäten und Ereignisse aufgeteilt werden. Ein Beispiel für eine solche Aufteilung wurde, wie bereits in Abschnitt 4.3 besprochen, durch den Autor in [60] für das Stadtbahnnetz der Stadt Köln vorgestellt.

### 5.1.3. Linien und Fahrten

Neben den bisher vorgestellten physischen Komponenten müssen zur Modellierung des Öffentlichen Personennahverkehrs auch logische Komponenten wie Linien, Fahrten und Fahr- bzw. Umlaufpläne berücksichtigt werden.

*Linien* bestehen aus einer geordneten Menge von Haltepunkten, die den im Regelbetrieb einzuhaltenen Linienvorlauf angibt. Im Simulationsmodell wird dies durch einen Verweis auf eine Menge entsprechender Knoten des Modell-Graphen abgebildet. Um eine aufwändige dynamische Wegfindung während des Simulationslaufs zu vermeiden, werden Linien für das Simulationsmodell zusätzlich um eine geordnete Menge der zwischen den anzufahrenden Haltepunkt-Knoten zu verwendenden Verbindungs-Knoten ergänzt. Darüber hinaus kann jede Linie einer spezifischen Verkehrsart (z.B. Bus oder Bahn) zugeordnet werden und darf nur von für sie zulässigen Fahrzeugtypen bedient werden. Beispielsweise darf eine Niederflurbahn keine Linie bedienen deren Haltepunkte für Hochflurbahnen ausgelegt sind. Entsprechend verwalten Linien im Modell Verweise auf ihre Verkehrsart und die für sie zulässigen Fahrzeugtypen. Linien mit verwandtem Linienvorlauf (z.B. Hin- und Rückrichtung) werden häufig unter einer gemeinsamen Liniennummer gruppiert (z.B. Linie 136 Richtung Neumarkt und Linie 136 Richtung Hohenlind). Im Modell erhält jede Linie entsprechend eine Liniennummer. Einzelne Linien einer Gruppe werden im Modell hingegen über einen Variantenbezeichner identifiziert. Wenn nicht anders gekennzeichnet setzt sich dieser Variantenbezeichner aus der Liniennummer gefolgt von einem Bindestrich, einem Indikator für den Verkehrstyp und einem numerischen Wert zusammen. Beispielsweise gehört Linie *1-B01* also zur Gruppe von Linien mit der Liniennummer 1, stellt eine Linie des Bussystems dar und ist innerhalb der Gruppe die erste Variante.

*Fahrten* kombinieren geordnete Mengen von Haltepunkten und Verbindungen mit einer Startzeit und werden in Betriebs- und Leerfahrten unterschieden. Im Modell verwalten Fahrten dazu, ähnlich wie Linien, Verweise auf Mengen von Haltepunkt- und Verbindungs-Knoten. *Betriebsfahrten* verweisen im Modell zusätzlich auf die durch sie bediente Linie und ihre Haltepunkt- und Verbindungsmengen müssen im Regelbetrieb den Haltepunkt- und Verbindungsmengen dieser Linie entsprechen. *Leerfahrten*

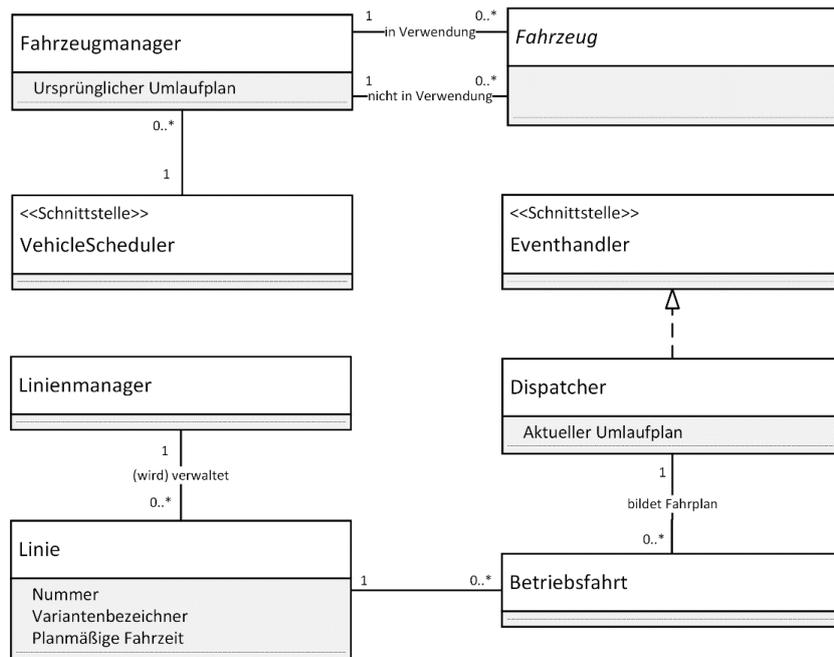


Abbildung 5.7: Verkürztes UML-Klassendiagramm für Verwaltungsmodul

folgen hingegen keinem vorgegeben Linienverlauf und verweisen im Modell entsprechend nicht auf eine Linie. Für alle Fahrten gilt, dass die Modellierung mit Hilfe eigenständiger Haltepunkt- und Verbindungsmengen – anstatt der Modellierung durch den Verweis auf eine Linie – eine dynamische Modifikation des Verlaufs einzelner Fahrten erheblich vereinfacht (für eine genauere Erläuterung siehe Kapitel 7). Abbildung 5.6 stellt die beschriebenen Sachverhalte anhand eines verkürzten UML-Klassendiagramms dar.

#### 5.1.4. Verwaltungsmodul

Die bisher vorgestellten Modellkomponenten sind ausreichend für die Abbildung elementarer Funktionen von ÖPNV-Systemen, sie vernachlässigen aber alle zur Funktionstüchtigkeit von Verkehrssystemen beitragenden übergeordneten Verwaltungstätigkeiten. So muss der Fahrplan um einen Umlaufplan ergänzt werden, der (Betriebs-)Fahrten zu Gruppen kombiniert, die innerhalb eines Betriebstags durch einzelne Fahrzeuge ausgeführt werden können. Erst auf Basis dieses Umlaufplans kann überhaupt entschieden werden welche Fahrzeuge aus dem Pool während eines Betriebstages eingesetzt werden müssen. Diese und andere Verwaltungstätigkeiten werden in drei Verwaltungsmodulen im Simulationsmodell gekapselt. So wirken sich Änderungen an der Modellierung einzelner Verwaltungstätigkeiten nicht signifikant auf die Modellierung anderer Bereiche des Simulationsmodells aus. Abbildung 5.7 zeigt das sich ergebende verkürzte UML-Klassendiagramm für die Verwaltungsmodul.

Im Folgenden werden lediglich die zur Modellierung des Regelbetriebs notwendigen Bestandteile der unterschiedlichen Module beschrieben. Die zur Einbindung des Störfallmanagements notwendigen Erweiterungen werden hingegen in Kapitel 7 beschrieben.

##### 5.1.4.1. Fahrzeugmanager

Zur Administration der Fahrzeugflotte wird der *Fahrzeugmanager* verwendet. Er erlaubt anderen Komponenten des Simulationsmodells über definierte Schnittstellen Zugriff auf den Fahrzeugpool, verwaltet welche Fahrzeuge aktuell (nicht) im Einsatz sind und ist für die Erzeugung und Verwaltung des initialen Umlaufplans zuständig.

Wird kein Umlaufplan durch die Anwenderin vorgegeben, verwendet der Fahrzeugmanager einen Umlaufplangenerator zur Erzeugung eines (künstlichen) Umlaufplans. Zur Interaktion zwischen Fahrzeugmanager und Umlaufplangenerator werden vorgegebene Schnittstellen verwendet (siehe Abbildung 5.7), sodass je nach Situation ein passender Umlaufplangenerator verwendet werden kann. Zum Zeitpunkt der Anfertigung dieser Arbeit liegt dem Simulationsmodell ein erster sehr simpler Generator bei, der Umlaufpläne mit den folgenden Eigenschaften erzeugen kann:

- i. Fahrzeuge tätigen nur Fahrten die für ihren Typ zulässig sind
- ii. Aufeinander folgende Fahrten müssen zu Linien mit identischer Liniennummer gehören
- iii. Start- und Endhaltepunkte aufeinanderfolgender Fahrten müssen Teil der gleichen Haltestelle sein
- iv. Die den Fahrzeugen zugeordneten Fahrten folgen einer zulässigen zeitlichen Ordnung, d.h. die zuletzt zugewiesene Fahrt kann beendet werden bevor die nächste Fahrt startet und
- v. Zwischen zwei einem Fahrzeug zugeordneten Fahrten wird eine durch die Anwenderin definierbare Mindestwendezeit eingehalten.

Wie Algorithmus 5.1 in Pseudocode zeigt, durchläuft das Verfahren dazu die gegebene Menge der (Betriebs-)Fahrten (Zeile 03) und sucht für jede Fahrt in der Menge der Fahrzeuge mit zulässigem Fahrzeugtyp (Zeile 05) entweder nach einem Fahrzeug dessen zuletzt zugewiesene Fahrt (Zeile 06) die Bedingungen ii. bis v. erfüllt (Zeilen 09 bis 11) oder, falls kein solches Fahrzeug existiert, nach einem Fahrzeug, das noch keine Fahrt zugewiesen bekommen hat (Zeilen 07 und 08). Sind mehrere Fahrzeuge zur Durchführung einer Fahrt geeignet, wird das Fahrzeug gewählt, dessen zuletzt zugewiesene Fahrt die frühestmögliche geplante Fertigstellungszeit aufweist (Zeilen 13 und 14). Ist ein gegebener Fahrzeugpool unter diesen Bedingungen nicht zur Abdeckung aller Fahrten geeignet oder werden keine Angaben zum Fahrzeugpool gemacht, erzeugt das Verfahren (künstliche) Fahrzeug-Entitäten und fügt sie dem Fahrzeugpool hinzu (Zeilen 18 bis 22).

---

**Input:** Menge aller Betriebsfahrten  $B$ , (Unbegrenzte) Menge aller Fahrzeuge  $F$ , Mindestwendezeit  $t_w$

**Output:** Aufteilung von  $B$  in disjunkte Teilmengen und Zuordnung zu Fahrzeugen aus  $F$

```

01 procedure ErstelleUmlaufplan( $B$ ,  $F$ )
02 begin
03   for alle  $b \in B$  do
04     Fahrzeug  $f_{opt} := 0$ 
05     for alle  $f \in F$  mit  $Typ(f) \cap Typ(b) \neq \emptyset$  do           /* i. */
06       Betriebsfahrt  $bf := Last(Fahrten(f))$ 
07       if  $bf = 0$  then
08          $f_{opt} := f$ 
09       else if  $Liniennummer(bf) = Liniennummer(b)$  and           /* ii. */
10          $Haltestelle(Ende(bf)) = Haltestelle(Start(b))$  and     /* iii. */
11          $Startzeit(bf) + Fahrtzeit(bf) + t_w \leq Startzeit(b)$  then /* iv. und v. */
12           Betriebsfahrt  $b_{opt} := Last(Fahrten(f_{opt}))$ 
13           if  $Startzeit(bf) + Fahrtzeit(bf) < Startzeit(b_{opt}) + Fahrtzeit(b_{opt})$  then
14              $f_{opt} := f$ 
15           fi
16       fi
17     od
18     if  $f_{opt} = 0$  then
19       Fahrzeug  $f_{neu} := erzeugeFahrzeug(Typ(b))$ 
20        $F := F \cup f_{neu}$ 
21        $f_{opt} := f_{neu}$ 
22     fi
23      $Fahrten(f_{opt}) := Fahrten(f_{opt}) \cup b$ 
24   od
25 end

```

---

*Algorithmus 5.1: Simple Heuristik zur Erzeugung eines Umlaufplans*

### 5.1.4.2. Linienmanager

Der *Linienmanager* administriert die im Rahmen eines Fahrplans bedienten Linien und assoziiert deren Hin- und Rückrichtungen miteinander. Darüber hinaus bietet er einheitliche Schnittstellen zum Zugriff auf einzelne Linien sowie Mengen von Linien. Dies ermöglicht bspw. den Zugriff auf alle Linien, die einen spezifischen Haltepunkt bedienen oder eine bestimmte Verbindung befahren.

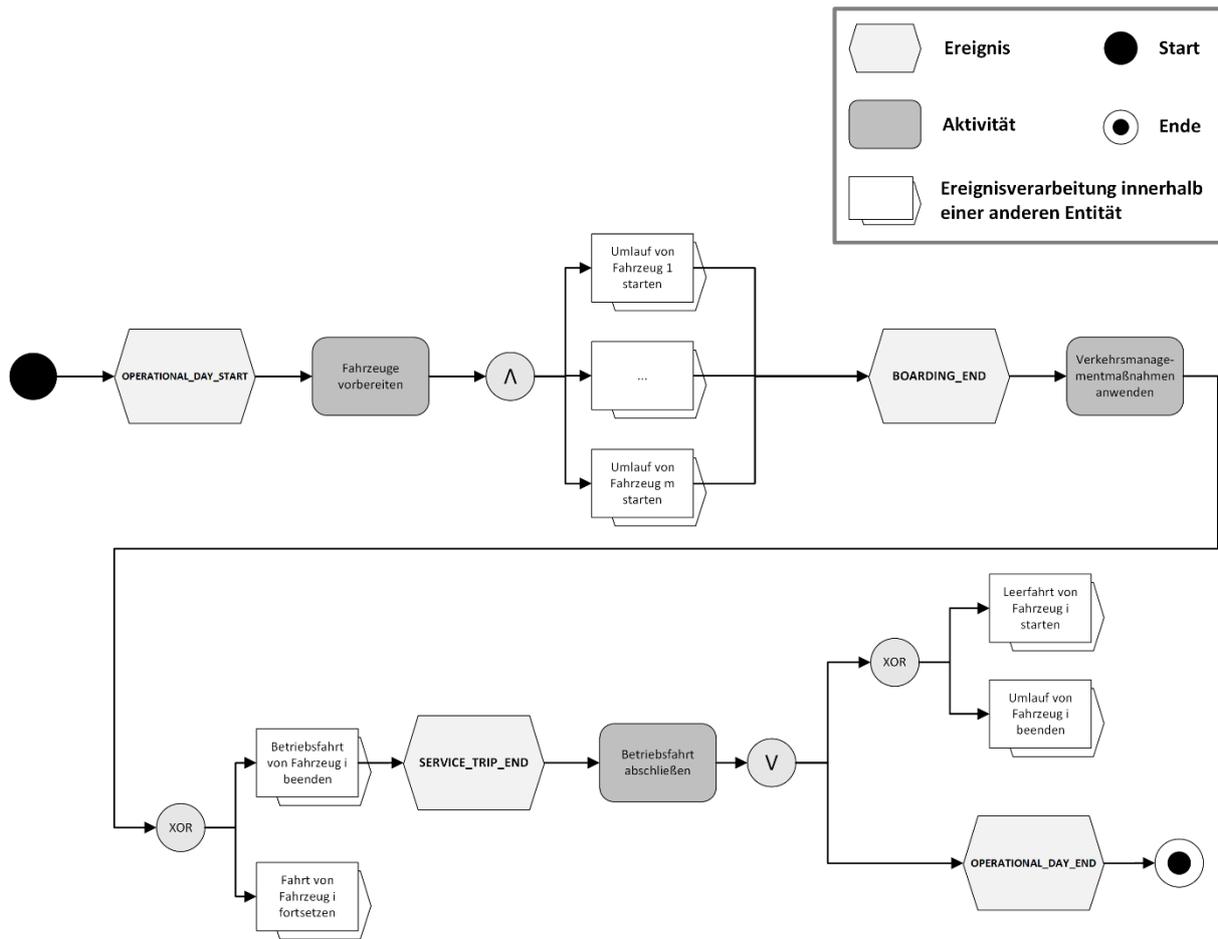


Abbildung 5.8: Ereignisprozesskette für Dispatcher-Entitäten

### 5.1.4.3. Dispatcher

Das wichtigste und umfangreichste Verwaltungsmodul ist der *Dispatcher*. Er bildet das zentrale Entscheidungsmodul und kann als Modell der (Entscheidungs-)Prozesse der Leitzentrale verstanden werden. Er hält alle für den Betriebsablauf notwendigen Daten, wie die zu einem spezifischen Zeitpunkt aktuellen Fahr- und Umlaufpläne. Neben der Verwaltung des Regelbetriebs umfasst der Dispatcher auch die für das Verkehrsmanagement notwendige Simulationslogik (siehe Kapitel 7).

Zur Abbildung des im Regelbetrieb stattfindenden Verhaltens wird zwischen vier verschiedenen Ereignistypen unterschieden (siehe Tabelle 5.2). Abbildung 5.8 zeigt die zwischen den Ereignistypen und Aktivitäten bestehenden Beziehungen.

Ereignistyp	Beschreibung
<b>OPERATIONAL_DAY_START</b>	Der Betriebstag beginnt
<b>OPERATIONAL_DAY_END</b>	Der Betriebstag endet
<b>BOARDING_END</b>	Ein Fahrzeug hat den Passagierwechsel abgeschlossen
<b>SERVICE_TRIP_END</b>	Ein Fahrzeug hat eine Betriebsfahrt abgeschlossen

Tabelle 5.2: Simulationsereignistypen zur Abbildung des Verkehrsmanagement im Regelbetrieb

Der Ereignistyp `OPERATIONAL_DAY_START` modelliert den Beginn des Betriebstages. Als Folge dieses Ereignistyps weist der Dispatcher den Fahrzeug-Entitäten des Fahrzeugpools, auf Basis des Umlaufplans, die erste von ihnen zu bedienende Fahrt zu. An jede zugewiesene Fahrzeug-Entität wird dabei ein Folgeereignis vom Typ `ROTATION_START` gesendet.

Das Ende des Betriebstages wird durch den Ereignistyp `OPERATIONAL_DAY_END` modelliert. Er signalisiert, dass alle Betriebsfahrten durchgeführt wurden und alle Fahrzeug-Entitäten ins Depot zurückgekehrt sind.

Wie bereits in Abschnitt 5.1.2 beschrieben, senden Fahrzeug-Entitäten im Verlauf des simulierten Betriebstages Ereignisse vom Typ `BOARDING_END`, die das Ende des reinen Ein-/Ausstiegsvorgang signalisieren, unter anderem an den Dispatcher. Dieser bestimmt daraufhin Kennzahlen, wie z.B. die im bisherigen Betriebsablauf durch die auslösende Fahrzeug-Entität eingefahrene Verzögerung und fällt auf deren Grundlage weitergehende Entscheidungen zu Verkehrsmanagementmaßnahmen. Dazu kann der Dispatcher auf unterschiedliche Strategien und Methoden zurückgreifen, die in Kapitel 7 detailliert beschrieben werden. Für die Abbildung des Regelbetriebs ist lediglich die Maßnahme *ortsbedingte Abfahrt nach Plan* zur Bestimmung des Abfahrtszeitpunkts eines Fahrzeugs relevant. Im Rahmen dieser Strategie werden ausgewählte Haltepunkte von Verkehrsplanern als Kontrollpunkte festgelegt, an denen Fahrzeuge im Regelbetrieb stets so lange halten bis ihr durch den Fahrplan definierter geplanter Abfahrtszeitpunkt erreicht wurde (siehe z.B. [30]). An allen anderen Haltepunkten des Netzes fahren Fahrzeuge hingegen stets ab sobald der Passagierwechsel abgeschlossen wurde, unabhängig davon ob die geplante Abfahrtszeit bereits erreicht wurde oder nicht. Die *ortsbedingte Abfahrt nach Plan* ermöglicht dabei sowohl eine aufwandsarme Sicherung von Umsteigeverbindungen zwischen unterschiedlichen Verkehrsmodi (bspw. zwischen Nah- und Regionalverkehr) als auch eine Reduzierung von ungeplanten Kolonnenfahrten (siehe dazu Abschnitt 5.1.5.2). Erreicht den Dispatcher ein Ereignis vom Typ `BOARDING_END` wird geprüft, ob Verkehrsmanagementmaßnahmen anzuwenden sind oder nicht. Abhängig vom Ergebnis dieser Prüfung wird dann unter anderem die zur Ein-/Ausstiegszeit hinzuzufügende Wartezeit bestimmt. Diese Wartezeit wird der betroffenen Fahrzeug-Entität mitgeteilt, indem ihr ein mit dem Ende der Wartezeit korrespondierendes Folgeereignis gesendet wird. Dieses Folgeereignis kann entweder vom Typ `SERVICE_TRIP_END` oder `DRIVING_START` sein. Ersteres ist der Fall, wenn sich die Fahrzeug-Entität am letzten Haltepunkt-Knoten ihrer aktuellen Fahrt befindet. Letzteres wird gesendet, um der Entität mitzuteilen, dass sie sich zum nächsten im Linienverlauf angegebenen Haltepunkt-Knoten bewegen soll.

Der letzte für den Regelbetrieb relevante Ereignistyp ist `SERVICE_TRIP_END`. Er bildet die Beendigung einer Betriebsfahrt und die darauf folgende Signalisierung der Leitstelle ab. Ereignisse dieses Typs werden von Fahrzeug-Entitäten an den Dispatcher gesendet, nachdem sie vorher bereits ein Ereignis von gleichem Typ, aber ausgelöst durch den Dispatcher, verarbeitet haben. Als Folge dieses Ereignisses weist der Dispatcher der Fahrzeug-Entität die laut des aktuellen Umlaufplans nächste durchzuführende Betriebsfahrt zu und veranlasst die vorher notwendige Leerfahrt. Hierbei wird darauf geachtet, dass zwischen dem Ende der einen Betriebsfahrt und dem Start der nächsten, die durch die Anwenderin vorgegebene Mindestwendezeit eingehalten wird. War die beendete Betriebsfahrt die für den simulierten Betriebstag letzte geplante Fahrt der Fahrzeug-Entität, weist der Dispatcher sie an, ihren Umlauf zu beenden. Sind mit dem Ende der Betriebsfahrt alle an diesem Betriebstag durchzuführenden Fahrten erledigt, so beendet der Dispatcher den Betriebstag, durch die Terminierung eines Ereignisses von Typ `OPERATIONAL_DAY_END`.

### 5.1.5. Randomisierung

Zur Abbildung dynamischer Effekte wie z.B. verkehrsabhängiger Schwankungen der Fahrzeiten, enthält das Simulationsmodell randomisierte Elemente. Dies sind die Fahrzeiten auf Verbindungs- bzw. Segment-Knoten, sowie – indirekt – die Dauer des Passagierwechsels an Haltepunkt-Knoten. Im Idealfall erfolgt die Randomisierung aufgrund statistischer Verteilungen, deren Typen sowie orts- und tageszeit-spezifische Parameter anhand empirischer Erhebungen bestimmt wurden. Wie in Abschnitt 5.1.1 bereits beschrieben, ermöglicht das Simulationsmodell die Einbindung eben solcher Verteilungsparameter für individuelle Verbindungs-, Segment- und Haltepunkt-Knoten. Die jeweils zu verwendende Verteilungsfunktion ist hingegen durch das Simulationsmodell vorgegeben, wird aber nur über festgelegte Schnittstellen angesprochen. Eine Anpassung der Verteilungsfunktionen ist entsprechend ohne Auswirkungen auf andere Modellkomponenten möglich.

#### 5.1.5.1. Fahrzeiten

Für Fahrzeiten wird, der Argumentation von Andersson et al. in [3] folgend, im Simulationsmodell eine Lognormalverteilung verwendet. Tritt nach erfolgreichem Transfer einer Fahrzeug-Entität auf einen Verbindungs- bzw. Segment-Knoten ein Ereignis von Typ `DRIVING_START` ein, wird mit Hilfe der

im jeweiligen Knoten hinterlegten Verteilungsparameter eine Zufallszahl erzeugt, die die minimale Dauer der Fahraktivität angibt. Für Knoten mit Überholmöglichkeiten – d.h. Verbindungen und Segmente des Bussystems – entspricht diese Dauer der tatsächlich benötigten Fahrzeit. Für Knoten ohne Überholmöglichkeiten – d.h. Gleise des Bahnsystems – muss zusätzlich noch überprüft werden, ob durch die berechnete Fahrzeit die festgelegte Fahrzeugreihenfolge verletzt wird. Dazu wird der durch die minimale Fahrzeit terminierte potentielle Eintrittszeitpunkt des Folgeereignisses mit dem des bereits terminierten Ereignisses der unmittelbar vorausfahrenden Fahrzeug-Entität verglichen. Wird dabei eine unzulässige zeitliche Ordnung zwischen den Ereignissen festgestellt, wird das neu zu terminierende Ereignis um die zeitliche Differenz zwischen den Eintrittszeitpunkten verzögert, die Fahrzeit also verlängert.

Stehen, wie im Falle der im Rahmen dieser Arbeit untersuchten Verkehrsnetze, keine empirischen Daten für Verteilungsparameter zur Verfügung, sondern nur die im veröffentlichten Fahrplan angegebenen geplanten Fahrzeiten, müssen Mittelwert und Standardabweichung auf Basis dieser Zeiten grob approximiert werden. Dazu soll die im Fahrplan veröffentlichte Fahrzeit  $t(c_i)$  für eine Verbindung  $c_i$  zunächst genauer betrachtet werden: Sie gibt die geplante Zeitdauer zwischen den jeweiligen Abfahrten an den verbundenen Haltepunkten an und lässt sich folglich in einen geplanten Fahrzeitanteil  $t^d(c_i)$  und einen geplanten Haltezeitanteil  $t^b(c_i)$  am Zielhaltepunkt unterteilen. Für jeden dieser Zeitanteile kann wiederum angenommen werden, dass er auf Basis von beobachteter durchschnittlicher Fahrzeit  $\mu_{c_i}^d$  bzw. Haltezeit  $\mu_{c_i}^b$  sowie operativer Überlegungen der Planer bestimmt wird. Diese operativen Überlegungen sind in der Regel nicht bekannt. Es kann jedoch angenommen werden, dass sie zumindest die beobachteten Standardabweichungen  $\sigma_{c_i}^d$  bzw.  $\sigma_{c_i}^b$  umfassen. Mit einer Aufteilung der operativen Überlegungen in Standardabweichungen sowie unbekannte Terme  $\epsilon_{c_i}^d$  bzw.  $\epsilon_{c_i}^b$  ergibt sich damit der in Gleichung (5.1) dargestellte Zusammenhang.

$$t(c_i) = t^d(c_i) + t^b(c_i) = (\mu_{c_i}^d + \sigma_{c_i}^d + \epsilon_{c_i}^d) + (\mu_{c_i}^b + \sigma_{c_i}^b + \epsilon_{c_i}^b) \quad \forall c_i \in C \quad \mu_{c_i}^d, \mu_{c_i}^b, \sigma_{c_i}^d, \sigma_{c_i}^b, \epsilon_{c_i}^d, \epsilon_{c_i}^b \in \mathbb{R} \quad (5.1)$$

Darüber hinaus lässt sich begründet vermuten, dass die geplante Fahrzeit  $t(c_i)$  immer größer als die durchschnittliche beobachtete Fahrzeit  $\mu_{c_i}^d$  gewählt werden wird. Andernfalls würde es auf der Verbindung zu systematischen Verspätungen kommen. Ohne Kenntnis der einzelnen Bestandteile in Gleichung (5.1) lässt sich die durchschnittliche Fahrzeit damit nur extrem grob als auf einen gewissen Prozentsatz  $0 < \gamma < 1$  reduzierte geplante Fahrzeit approximieren:

$$\hat{\mu}_{c_i}^d = t(c_i) * \gamma \quad \forall c_i \in C, 0 < \gamma < 1 \quad (5.2)$$

Hierbei muss der Prozentsatz  $\gamma$  aufgrund von Erfahrungswissen der Anwenderin bestimmt werden.

Auf gleiche Weise wird die Approximation der Standardabweichung  $\hat{\sigma}_{c_i}^d$  berechnet. Dabei kann allerdings angenommen werden, dass sie einen geringeren Anteil an der geplanten Fahrzeit ausmacht. Entsprechend gilt für den ebenfalls durch die Anwenderin bestimmten Prozentsatz  $0 < \eta < 1$ , auf den die geplante Fahrzeit zur Approximation reduziert wird, zusätzlich  $\eta \ll \gamma$  und es ergibt sich:

$$\hat{\sigma}_{c_i}^d = t(c_i) * \eta \quad \forall c_i \in C, 0 < \eta < 1, \eta \ll \gamma \quad (5.3)$$

Es ist unmittelbar einsichtig, dass sich die auf einer Verbindung zu findenden Kreuzungen, Weichen und Signale ebenfalls auf die geplante Fahrzeit auswirken. Es erscheint dabei plausibel, dass sich diese Auswirkungen nicht per se in einer Erhöhung der durchschnittlichen Fahrzeit münden, sondern vielmehr eine erhöhte Streuung der Fahrzeit zur Folge haben. Beispielsweise wird sich für einen Bus während einer „grünen Welle“ eine wesentlich geringere Fahrzeit ergeben, als unter herkömmlichen Bedingungen. Analog kann sich die Fahrzeit im Fall einer „roten Welle“ wesentlich erhöhen.

Für eine sehr simple Abbildung dieses Sachverhalts und eine daraus folgende Verfeinerung der Approximation der Standardabweichung reicht es dabei schon aus die Zahl der Segmente  $|SG(c_i)|$  einer Verbindung  $c_i$  zu kennen. Wie bereits in Abschnitt 5.1.1 ausgeführt, stellen Segmente kleinste Streckenabschnitte dar, an deren Endpunkten sich stets für die Modellierung der Verbindung relevante Elemente

wie Haltepunkte, Signale, Kreuzungen oder Weichen befinden. Für aufeinander folgende Segmente einer gegebenen Menge  $SG(c_i)$ , mit  $|SG(c_i)| > 1$ , gilt insbesondere

- dass an ihrem Übergang kein für  $c_i$  relevanter Haltepunkt liegen kann und
- dass dort nur dann ein Haltepunkt einer anderen Verbindung liegen kann, falls auch ein für die Modellierung von  $c_i$  relevantes Element dort vorzufinden ist.

Andernfalls könnte Verbindung  $c_i$  an diesem Punkt in zwei Teilverbindungen  $c_i^-$  und  $c_i^+$  aufgeteilt werden ohne die Aussagekraft des Modells zu verringern.

Damit kann die Zahl der Kreuzungen, Signale und Weichen auf einer Verbindung über die Zahl der Segmente bestimmt werden und entspricht  $|SG(c_i)| - 1$ . Unter der Annahme dass jeder dieser Segmentübergänge Auswirkungen auf die Streuung der Fahrzeit hat, kann die Approximation der Standardabweichung wie in Gleichung (5.4) gezeigt verbessert werden. Dabei wird angenommen, dass jeder Segmentübergang den Anteil der Standardabweichung an der geplanten Fahrzeit um einen festen, von der Nutzerin bestimmten, Prozentsatz  $0 \leq \iota < 1, \iota \ll 1$  erhöht.

$$\hat{\sigma}_{c_i}^d = t(c_i) * (\eta + (|SG(c_i)| - 1) * \iota) \quad \begin{array}{l} \forall c_i \in C \\ 0 < \eta < 1, 0 \leq \iota < 1, \iota \ll 1 \\ \eta + (|SG(c_i)| - 1) * \iota \ll \gamma \end{array} \quad (5.4)$$

Die notwendigen Daten vorausgesetzt, können die Verteilungsparameter für die Fahrzeiten auf einzelnen Segmenten analog zu Gleichungen (5.2) und (5.3) approximiert werden.

### 5.1.5.2. Passagierwechsel

Die Randomisierung des Passagierwechsels erfolgt im Simulationsmodell indirekt über die Abbildung von *ungewollten Kolonnenfahrten* (engl. *Bunching*). Dabei handelt es sich um das Phänomen, dass zwei Fahrzeuge die einen gemeinsamen Haltepunkt planmäßig in gleichmäßigem Abstand bedienen sollen, (nahezu) zeitgleich dort ankommen und für den Rest ihres gemeinsamen Streckenabschnitts ungewollt eine Kolonne bilden (siehe z.B. [1], [22]). Wie Abkowitz und Tozzi in [1] schreiben ist Bunching ein Symptom von Zuverlässigkeitsproblemen im ÖPNV und wirkt sich negativ auf die durch Passagiere wahrgenommene Angebotsqualität aus. Es wird durch die ungewollte Verzögerung von Fahrzeugen ausgelöst. Wenn diese Verzögerung zum nächsten planmäßigen Halt des Fahrzeugs mitgetragen wird, steigen bei einer angenommenen regelmäßigen Passagierankunftsrate mehr Passagiere als ursprünglich geplant zu, wodurch sich die Dauer des Passagierwechsels erhöht. Dadurch wiederum wird das betroffene Fahrzeug daran gehindert seine Verzögerung zu verringern und der Prozess wiederholt sich am darauf folgenden Haltepunkt. Umgekehrt benutzen weniger Passagiere das auf das verspätete Fahrzeug folgende Fahrzeug, wodurch sich die Dauer des Passagierwechsels für dieses verringert. Als Folge dieses Prozesses wird das verspätete Fahrzeug von seinem Nachfolger eingeholt und es kommt zur Kolonnenbildung.

Bunching tritt vornehmlich in Verkehrssystemen mit hoher Taktfrequenz auf, in denen angenommen werden kann, dass Passagiere zufällig gleichverteilt an ihren Starthaltepunkten ankommen (siehe [1], [22], [71]). In Systemen mit niedriger Taktfrequenz stimmen Passagiere ihre Ankunft hingegen in der Regel auf die im Fahrplan veröffentlichte Ankunfts-/Abfahrtszeit ab, so dass Schwankungen in den Sicherheitsabständen weniger starke Auswirkungen auf die Dauer des Passagierwechsels haben.

Ein simples und leicht zu erweiterndes mathematisch-analytisches Modell zur Abbildung von Bunching wurde von Chapman und Michel in [22] vorgestellt. Ausgangspunkt des Modells ist die Betrachtung aufeinander folgender Fahrzeuge einer Linie  $l_i$  unter Annahme identischer und deterministischer Fahrzeiten zwischen allen Haltepunkten. Darüber hinaus nehmen die Autoren vereinfachend an, dass die Dauer des Passagierwechsels lediglich von den einsteigenden Passagieren bestimmt wird. Ist unter diesen Annahmen die von jedem Passagier zum Einsteigen benötigte Zeitdauer  $B$  sowie die Zahl  $N_{f,s}$  der an einem Haltepunkt  $s$  in ein Fahrzeug  $f$  einsteigender Passagiere bekannt, kann die Einstiegsdauer  $T_{f,s}$  für ein Fahrzeug  $f$  an Haltepunkt  $s$  wie in Gleichung (5.5) gezeigt bestimmt werden. Hierbei bezeichnet  $C^b$  eine konstante Mindestzeit, etwa zum Öffnen und Schließen der Fahrzeugtüren.

$$T_{f,s} = C^b + B * N_{f,s} \quad (5.5)$$

Ist, wie beim Modell von Chapman und Michel angenommen, auch die Passagierankunftsrate  $a$  bekannt, kann die Zahl der einsteigenden Passagiere  $N_{f,s}$  in Abhängigkeit von ihr und dem Fahrzeugabstand berechnet werden. Letzterer entspricht im störungsfreien Betrieb dem Takt  $T_{L(f)}$  der Linie  $L(f)$  von Fahrzeug  $f$ , so dass sich Gleichung (5.6) ergibt.

$$N_{f,s} = T_{L(f)} * a \quad (5.6)$$

Gleichung (5.5) lässt sich damit wie folgt umformen:

$$T_{f,s} = C^b + T_{L(f)} * a * B \quad (5.7)$$

Wird nicht der geplante Fahrzeugabstand  $T_{L(f)}$ , sondern der tatsächlich auftretende, schwankende Fahrzeugabstand verwendet, zeigt sich das gewünschte Verhalten, d.h. ein größerer Fahrzeugabstand resultiert in einem längeren Passagierwechsel während ein geringerer Abstand eine Verkürzung des Passagierwechsels zur Folge hat. Chapman und Michel zeigen dies allgemein für  $m \geq 3$  Fahrzeuge und  $n \geq 1$  Haltepunkte, indem sie das zweite Fahrzeug (indiziert mit  $m = 1$ ) um einen Prozentsatz  $d > 0$  verzögern, d.h.  $N_{1,1} = d * T_{L(1)} * a$ . Dabei stellen sie fest, dass in ihrem Modell Fahrzeuge mit ungeradem Index eine höhere Ein-/Ausstiegsdauer zugewiesen bekommen als Fahrzeuge mit geradem Index. Aufgrund der ursprünglichen Annahme deterministischer Fahrzeiten kommt es so zu Bunching.

Das Modell von Chapman und Michel kann mit leichten Modifikationen für das Simulationsmodell adaptiert werden. Dazu werden Mindesthaltezeit  $C$  und Einstiegsrate  $B$  als Attribute der Fahrzeugeigenschaften, also fahrzeugspezifisch, modelliert. Die Ankunftsrate  $a$  wird hingegen als Attribut von Haltepunkten abgebildet. Damit kann die Dauer des Passagierwechsels für Fahrzeug  $f$  an Haltepunkt  $s$  zum Simulationszeitpunkt  $t$  wie folgt berechnet werden:

$$T_{f,s}(t) = \begin{cases} C_f^b, & f \text{ ist erstes Fahrzeug an } s \\ C_f^b + (t - t^{dep}(f-1, s)) * a_s * B_f, & \text{sonst} \end{cases} \quad (5.8)$$

$t^{dep}(f-1, s)$  bezeichnet hier den Abfahrtszeitpunkt des Vorgängerfahrzeugs von  $f$  an  $s$ , der bei Eintritt eines Ereignisses des Typs BOARDING\_END im betroffenen Haltepunkt-Knoten abgelegt werden kann. Um ungewollte Simulationsartefakte zu vermeiden, wird bei der Berechnung der erste Halt eines Betriebstages an einem Haltepunkt-Knoten gesondert betrachtet. Zu diesem Zeitpunkt kann noch keine Vorgängerentität am Knoten gehalten haben, so dass  $t^{dep}(f-1, s) = 0$  gilt, und sich ohne gesonderte Betrachtung unrealistisch hohe Ein-/Ausstiegsdauern ergeben.

Das beschriebene Teilmodell ermöglicht die plausible Abbildung des Passagierwechsels auch bei Vorlage einer schmalen Datenbasis und ist sowohl für Bus- als auch Bahnsysteme geeignet. Dabei kommt es in der hier beschriebenen Version ohne eigene randomisierte Elemente aus. Schwankungen in der Dauer des Passagierwechsels folgen stattdessen gänzlich aufgrund von Schwankungen in den Fahrzeiten, wodurch die Zahl potentieller Fehlerquellen im Modell reduziert wird. Wird eine weitere Randomisierung des Passagierwechsels gewünscht, kann das Modell leicht um entsprechende Elemente erweitert werden. Beispielhaft sei hier das von Adamski in [2] vorgestellte Modell erwähnt, das eine Möglichkeit aufzeigt sowohl die Zahl der ein-/aussteigenden Passagiere als auch die jeweils von einzelnen Passagieren benötigte Zeitdauer als stochastische Zufallsgrößen zu modellieren. Liegen entsprechend umfangreiche Rohdaten vor, kann so ein realistischerer Passagierwechsel modelliert werden.

Eingeschränkt wird die Aussagekraft des Modells aufgrund der pauschalen Annahme, dass Haltezeiten aufeinander folgender Fahrzeuge voneinander abhängig sind. Dies gilt nur, sofern die betroffenen Fahrzeuge im Anschluss an den gemeinsamen Haltepunkt noch weitere Haltepunkte gemeinsam haben bzw. die von den Fahrzeugen bedienten Linien für die potentiellen Passagiere substituierbar sind. Andernfalls hat der zeitliche Abstand zwischen Ankunft und Abfahrt der Fahrzeuge keine Auswirkung auf die Zahl der zusteigenden Passagiere. Das Modell aus Gleichung (5.8) kann diesen Sachverhalt nicht darstellen und resultiert an Haltepunkten, nach denen sich Linienverläufe trennen, u.U. in künstlich stärkeren Bunching-Effekten. Da solche Haltepunkte i.d.R. aber nur einen geringen Anteil der Gesamt-Haltepunkt-Menge ausmachen, sollten sich die resultierenden Effekte nicht negativ auf die Plausibilität

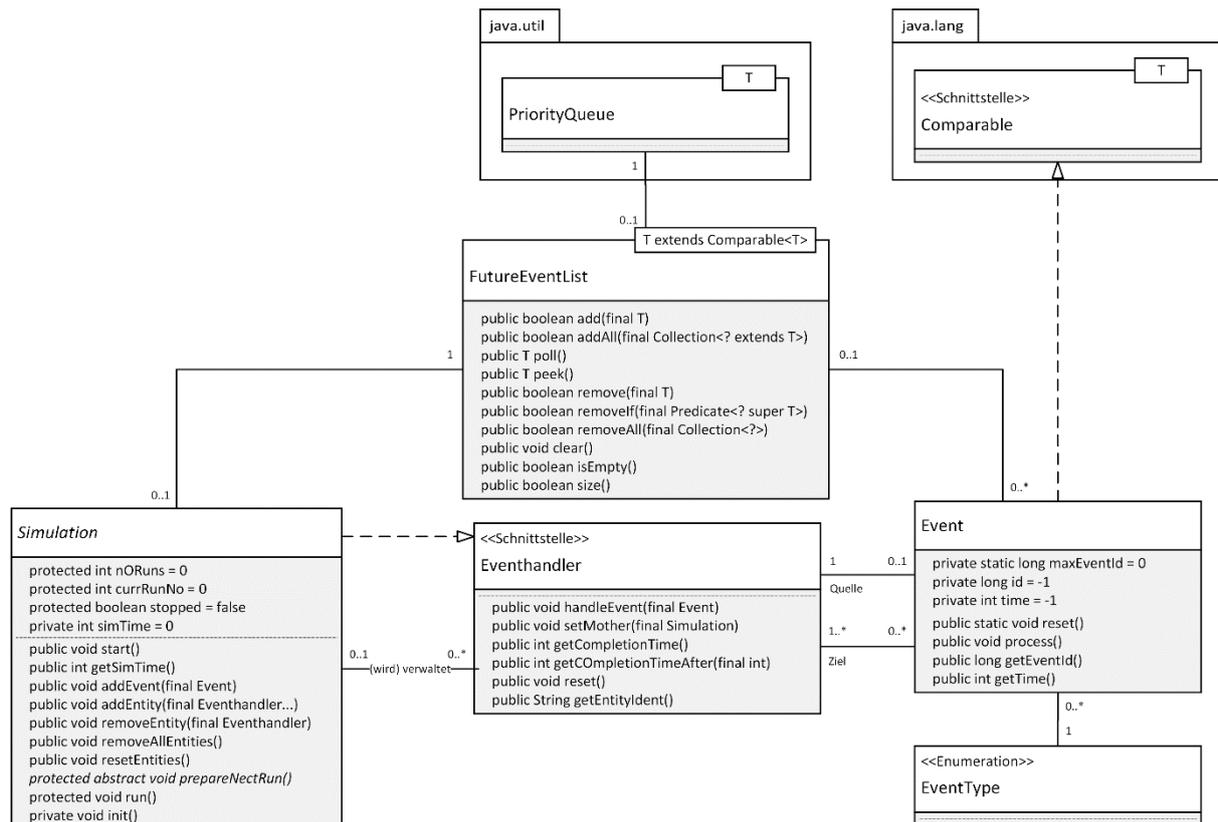


Abbildung 5.9: UML-Klassendiagramm des generischen Simulationsframeworks

des Gesamt-Modells auswirken. Darüber hinaus kann das Modell aber bei Bedarf zukünftig entsprechend erweitert werden, indem z.B. die letzte Abfahrtszeit, sowie die Ankunftsrate für Gruppen von für Passagieren substituierbaren Linien verwaltet werden.

## 5.2. Implementierung

Zur Umsetzung des Simulationsmodells wurde, wie beim Optimierungsmodell, eine Anwendung in Java 8 realisiert. Auch hier fand die Entwicklung unter *Microsoft Windows 7* (64 Bit), mit *Eclipse Mars* in der Version 4.5.1 und unter Verwendung derselben externen Bibliotheken statt (siehe Abschnitt 3.2).

Die Anwendung umfasst ca 7.000 Zeilen Quellcode verteilt auf 46 Klassen, 6 Enumerationstypen und 4 Schnittstellen.

### 5.2.1. Das generische Simulationsframework

Basis der Simulationsanwendung ist ein selbstentwickeltes, generisches, ereignisbasiertes Simulationsframework, das einzig die Standardbibliotheken von Java verwendet. Es ist nicht spezifisch für Verkehrssimulationen entwickelt, sondern kann für jede Art von ereignisbasierter Simulation verwendet werden. Wie das UML-Klassendiagramm in Abbildung 5.9 zeigt, teilt sich das Framework in die drei Klassen *Simulation*, *FutureEventList* und *Event*, die Schnittstelle *EventHandler*, sowie den Enumerationstyp *EventType* auf.

Zur Abbildung von ereignisverarbeitenden Komponenten wird die Schnittstelle *EventHandler* bereitgestellt. Sie definiert sechs Methoden, die von Entitäten des Simulationsmodells implementiert werden müssen wenn sie Quelle und/oder Ziel von Ereignissen sind:

- *public void handleEvent(final Event)*: Ein übergebenes Ereignis muss durch die Entität verarbeitet werden. Ob und wie das Ereignis tatsächlich verarbeitet wird, wird durch die jeweilige Entität geregelt.
- *public void setMother(final Simulation)*: Der Entität wird eine Simulations-Mutteranwendung zugewiesen. Über diesen Verweis erhält die Entität Zugriff auf allgemeine Simulationsmethoden, wie z.B. zum Hinzufügen von Ereignissen zur Future Event List.

- *public int getCompletionTime():* Die Entität liefert den Eintrittszeitpunkt ihres nächsten Ereignisses, d.h. den Fertigstellungszeitpunkt ihrer aktuellen Aktivität, zurück. Mit Hilfe dieser Funktion können Entitäten ihre Ereignisse und Aktivitäten miteinander koordinieren und die Zahl der Statusanfragen reduzieren.
- *public int getCompletionTimeAfter(final int):* Die Entität liefert den Eintrittszeitpunkt des nächsten Ereignisses zurück, das später als der übergebene Simulationszeitpunkt liegt. Andernfalls wird ein anwendungsspezifischer Sonderwert (z.B. null) zurückgeliefert. Diese Methode wird bspw. dazu genutzt um den frühestmöglichen Zeitpunkt zu bestimmen zu dem eine Fahrzeug-Entität erneut versuchen kann an einen bereits voll belegten Haltepunkt-Knoten heranzufahren.
- *public void reset():* Die (Neu-)Initialisierung der Entität wird angestoßen, d.h. alle für einen Simulationslauf relevanten Datenfelder sollen auf ihren initialen Stand zurückgesetzt werden.
- *public String getEntityIdent():* Eine Zeichenkette zur Identifizierung der Entität wird zurückgeliefert.

Die durch die Entitäten zu verarbeitenden Ereignisse werden durch die Klasse *Event* repräsentiert. Kernbestandteil dieser Klasse sind Datenfelder und Zugriffsmethoden für den Ereignistyp, den Eintrittszeitpunkt sowie die Quell- und Zielentität(en). Zusätzlich stellt sie die Methode *void process()* zur Ereignisverarbeitung bereit. Diese stößt für alle im Ereignis hinterlegten Zielentitäten wiederum die Methode *void handleEvent(final Event)* an, wobei das Ereignis selber den Übergabeparameter darstellt.

Zur Abbildung unterschiedlicher Ereignistypen wird der Enumerationstyp *EventType* verwendet. Jede Instanz der Klasse *Event* muss bei ihrer Erzeugung einen in *EventType* definierten Typ zugewiesen bekommen, um durch Entitäten der Simulation verarbeitet werden zu können. Standardmäßig enthält *EventType* dabei nur die beiden Ereignistypen *SIMULATION\_START* und *SIMULATION\_END*. Alle weiteren anwendungsspezifischen Ereignistypen müssen dem Enumerationstyp durch die Anwenderin hinzugefügt werden. *EventType* stellt damit die einzige Komponente des generischen Simulationsframeworks dar, die zwingend direkt durch die Nutzerin modifiziert werden muss.

Verwaltet werden Ereignisse durch die Klasse *FutureEventList*. Sie ist eine generische Wrapper-Klasse zur Bereitstellung standardisierter Zugriffsmethoden. Zum Zeitpunkt der Anfertigung dieser Arbeit erfolgt die interne Verwaltung der Ereignisse mit Hilfe der Prioritätsschlangenimplementierung der Java-Standardbibliothek.<sup>18</sup> Ein Austausch der Prioritätsschlange durch eine andere, für die konkrete Anwendung besser geeignete, Datenstruktur ist einfach möglich. Solange die durch *FutureEventList* definierten Schnittstellen dabei nicht verändert werden, erfolgt der Austausch der Datenstruktur dabei transparent für alle weiteren Bestandteile des Simulationsframeworks.

Um eine Ordnung auf den mit *FutureEventList* verwalteten Elementen zu erzeugen, müssen sie die Schnittstelle *Comparable* der Java-Standardbibliothek realisieren. Die so definierte Ordnung muss dabei eindeutig sein, da andernfalls eine willkürliche Auflösung von Konflikten droht. Auf Instanzen der Klasse *Event* wird eine solche eindeutige Ordnung definiert, indem zusätzlich zur Eintrittszeit *time* auch die für jede Instanz eindeutige ganzzahlige Variable *id* für Vergleiche herangezogen wird. Sie wird mit Hilfe des statischen Zählers *maxEventId* zum Zeitpunkt der Instanziierung automatisch zugewiesen und gibt die Erzeugungsreihenfolge aller *Event*-Instanzen wieder. Weisen zwei Ereignisse also die gleiche Eintrittszeit auf, wird *id* als zusätzliches Vergleichskriterium herangezogen. Das Ereignis mit der kleineren *id* – also das früher erzeugte – wird dann weiter vorne in die FEL einsortiert.

Wichtigster Bestandteil des Frameworks ist die abstrakte Klasse *Simulation*. Sie muss zur Erstellung einer eigenen Simulationsanwendung überschrieben werden und ist so konzipiert, dass beliebig viele Simulationsläufe mit identischen Parametern nacheinander ausgeführt werden können. Darüber hinaus verwaltet sie wichtige Datenfelder, wie beispielsweise die FEL und die aktuelle Simulationszeit sowie

---

<sup>18</sup> Die Implementierung der Java-Standardbibliothek basiert auf einem Prioritäts-Heap der für das Hinzufügen eines Elements sowie für das Zurückgeben (mit Löschen) des ersten Elements eine Laufzeit von  $O(\log(n))$  garantiert. Die Entfernung eines beliebigen Elements erfolgt mit einer Laufzeit von  $O(n)$ . Das Zurückliefern (ohne Löschen) des ersten Elements, sowie Abfragen zur Anzahl der gespeicherten Elemente erfolgen in konstanter Zeit (siehe auch [48]).

definierte standardisierte Zugriffsmethoden. Ihr Einstiegspunkt ist die Methode *void start()*, die die sequentielle Durchführung der Simulationsläufe veranlasst und von außen angestoßen werden muss. Vor jedem einzelnen Simulationslauf wird dabei zunächst die (Neu-)Initialisierung der allgemeinen Simulationsbestandteile über die private Methode *void init()* angestoßen. In ihr werden alle eventuell noch aus einem vorhergehenden Simulationslauf stammenden Ereignisse aus der FEL gelöscht, die Simulationszeit *simTime* und der Ereigniszähler *maxEventId* zurückgesetzt und schließlich ein neues Starterereignis vom Typ *SIMULATION\_START* in die FEL injiziert. Nach Abschluss der allgemeinen Initialisierung folgt die anwendungsspezifische (Neu-)Initialisierung über die abstrakte Methode *void prepareNextRun()*. Sie muss von der *Simulation* überschreibenden Klasse implementiert werden und alle nicht durch *init()* behandelten Datenfelder, d.h. die spezifisch für die Simulationsanwendung notwendigen Konstrukte wie z.B. den Graph des Streckennetzes, für den nächsten Simulationslauf vorbereiten. Nachdem beide Initialisierungsmethoden abgeschlossen sind, werden einzelne Simulationsläufe über die Methode *void run()* angestoßen. Wie Algorithmus 5.2 zeigt, folgt die Implementierung der internen Methode dem in Abbildung 5.1 dargestellten Schema: Zuerst wird das nächste Ereignis aus der FEL geholt (Zeile 03), anschließend wird die Simulationszeit auf den Eintrittszeitpunkt des Ereignisses gesetzt (Zeile 04) und schließlich wird die Ereignisverarbeitung der Zielentitäten angestoßen (Zeile 05). Zuletzt wird geprüft ob die FEL noch Ereignisse enthält und falls dem nicht so ist (und die Simulation nicht bereits anderweitig gestoppt wurde) wird ein Ereignis vom Typ *SIMULATION\_END* in die FEL injiziert, um die mit Simulationsende verbundenen Aktivitäten anzustoßen (Zeilen 06 bis 08). Letzteres bildet einen Sicherheitmechanismus um zu garantieren, dass in jedem Fall ein Ereignis vom Typ *SIMULATION\_END* terminiert wird und eine Nutzerin in ihrer eigenen Simulationsanwendung eine geordnete Beendigung von Simulationsläufen sicherstellt, d.h. zumindest bewusst die Variable *stopped* auf den Wert *true* setzt. Soll zur Beendigung der Simulationsanwendung nicht gewartet werden bis die FEL leer ist und das Simulationsframework ein Ereignis vom Typ *SIMULATION\_END* erzeugt, kann durch die Anwenderin in ihrer eigenen Simulationsanwendung jederzeit ein *SIMULATION\_END*-Ereignis erzeugt und in die FEL injiziert werden bzw. die Variable *stopped* den Wert *true* zugewiesen bekommen.

---

```

01 protected void run() {
02     while (!stopped) {
03         Event currentEvent = futureEventList.poll();
04         simTime = currentEvent.getTime();
05         currentEvent.process();
06         if (futureEventList.isEmpty() && !stopped) {
07             Event stop = new Event(simTime, EventType.SIMULATION_END, this, this);
08             addEvent(stop);
09         }
10     }
11 }
12 }

```

---

**Algorithmus 5.2:** Quellcode der Methode *void run()*

Zeile 07 in Algorithmus 5.2 zeigt eine weitere Besonderheit des Simulationsframeworks: Die Klasse *Simulation* – und damit alle von ihr ererbenden Simulationsanwendungen – implementiert selber ebenfalls die Schnittstelle *EventHandler*, kann also Quelle und Ziel von Ereignissen sein. Dies bedeutet zum einen, dass alle *Simulation* überschreibenden Klassen auch die durch *EventHandler* definierten Methoden implementieren müssen. Zum anderen ermöglicht diese Art der Implementierung eine sehr einfache Abbildung und Verarbeitung allgemeiner, d.h. nicht spezifischer Entitäten des abzubildenden Systems zuzuordnender, Ereignisse.

### 5.2.2. Die Anwendung zur Verkehrssimulation

Zur Implementierung der eigentlichen Simulationsanwendung werden die in Abschnitt 5.1 identifizierten Modellkomponenten in Java-Klassen, -Schnittstellen und -Enumerationstypen überführt. Entitäten und Ereignisse werden dabei unter Anwendung des generischen Frameworks erstellt. Darüber hinaus

umfasst die Simulationsanwendung sowohl eine Reihe von universellen als auch für das Untersuchungsgebiet spezifische Hilfsklassen, z.B. zur Erstellung von (Pseudo-)Zufallszahlen, zum Einlesen von Parametern aus einer XML-Datei oder zum Auslesen der Netz- und Fahrplandaten aus einer Datenbank.

Die Anwendung beginnt mit der Instanziierung der von *Simulation* ererbenden Klasse *DefuniakSprings*. Sie liest zunächst mit Hilfe der Klasse *PropertiesManager* die durch die Nutzerin zu spezifizierenden Anwendungsparameter, wie z.B. die zur Randomisierung der Fahrzeiten zu verwendenden Werte  $\gamma$ ,  $\eta$  und  $\iota$  ein. Anschließend werden die Fahrzeug-, Netz-, Fahr- und Umlaufplandaten durch die als Singleton implementierte Klasse *Dataconnector* aus der angegebenen Datenbank ausgelesen und Instanzen der zugehörigen Klassen erstellt. Nach Abschluss dieser Tätigkeiten wird durch den Aufruf der Methode *start()* die sequentielle Ausführung der Simulationsläufe veranlasst.

Vor, während und nach jedem Simulationslauf werden mit Hilfe der ebenfalls als Singleton implementierten Klasse *StatisticsProcessor* statistische Daten zu allen Fahrten, Varianten, Linien, Fahrzeugen und Knoten erfasst. Diese Daten werden dabei nicht in den sie erzeugenden Instanzen gespeichert, sondern in eigenständigen Datenstrukturen. So wird die Erfassung statistischer Daten von der eigentlichen Simulationslogik getrennt, Änderungen in einem Bereich wirken sich folglich nicht auf die Funktionsweise des anderen aus. Nach Abschluss aller Simulationsläufe werden die für einzelne Läufe erfassten Daten automatisch über alle durchgeführten Läufe aggregiert und eine Reihe relevanter Kennzahlen berechnet. Sowohl die Daten einzelner Läufe als auch die über alle Läufe aggregierten Daten werden anschließend in CSV-Dateien abgespeichert, sodass ein einfacher Import in Tabellenkalkulationsprogramme ermöglicht wird.

### 5.3. Validierung

Die in den vorangegangenen Abschnitten beschriebene Simulationsanwendung wird einer funktions- und theoriebezogenen Validierung unterzogen. Eine ergebnisbezogene Validierung kann aufgrund fehlender empirischer Vergleichsdaten (zunächst) nicht erfolgen. (Eine Übersicht über die verschiedenen Validierungsarten findet sich in [57], S. 206ff.)

Um sicherzustellen, dass die entwickelte Simulationsanwendung zur Bewertung von Fahrplänen für Nahverkehrssysteme geeignet ist, werden mit der in Kapitel 3 beschriebenen Optimierungsanwendung zwei 12-Stunden-Fahrpläne – von sieben Uhr morgens bis sieben Uhr abends – ohne Vorgabe verkehrsplanerischer Nebenbedingungen erzeugt. Die generierten Fahrpläne unterliegen dabei identischen Parametern und unterscheiden sich lediglich anhand des Zielfunktionswerts. Der eine Fahrplan entspricht dem durch die Optimierungsanwendung ermittelten Optimum, während der andere dem schlechtesten noch gültigen Fahrplan entspricht. Er wird erzeugt, indem in Zielfunktion (3.7) die Maximums-Anweisung durch eine Minimums-Anweisung ersetzt wird. Die unterschiedliche Güte der Fahrpläne sollte sich in den durch die Simulation erhobenen Kennzahlen widerspiegeln.

Für jeden der erzeugten Fahrpläne werden 100 Simulationsläufe durchgeführt und (Verspätungs-) Kennzahlen ermittelt. Simuliert wird der gesamte Zeitraum des Fahrplans, Kennzahlen werden aber nur im Simulationszeitraum von acht Uhr morgens bis sechs Uhr abends erhoben, um Verzerrungen aufgrund der Ein- und Ausschwingphase der Simulation zu vermeiden.

Um sicherzustellen, dass die Simulationsanwendung sowohl zur Abbildung von Bus- als auch Stadtbahnssystemen geeignet ist, wird dieser Prozess auf zwei Modelle von Verkehrsnetzen mit unterschiedlichen Charakteristiken angewendet: Das künstliche Busnetzwerk UCL 2015 (siehe Abbildung 3.2) und das Kölner Stadtbahnnetz aus dem Jahr 2001 (siehe Abbildung 3.3 bis Abbildung 3.6).

Zur Überprüfung der Leistungsfähigkeit der Simulationsanwendung werden außerdem noch einige Laufzeitvergleiche durchgeführt, für die neben den beiden bereits erwähnten Netzen auch das Kölner Busnetz aus dem Jahr 2001 (siehe ebenfalls Abbildung 3.3 bis Abbildung 3.6) herangezogen wird. Das Kölner Stadtbahn- bzw. Busnetz wird im Folgenden auch als KVB-T 2001 bzw. KVB-B 2001 bezeichnet. Tabelle 5.9 fasst die relevanten Charakteristiken der drei Netze zusammen.

Netz	L	S	C	SG
<b>UCL 2015</b>	8	40	50	0
<b>KVB-T 2001</b>	30	528	528	584
<b>KVB-B 2001</b>	68	1242	2482	2482

Tabelle 5.3: Kennzahlen der simulierten Streckennetze

Da weder für das künstliche Netz, noch für die beiden Kölner Netze empirischen Daten vorliegen, werden notwendige Simulationsparameter, soweit möglich, aufgrund stichpunktartiger Beobachtungen des Kölner Nahverkehrs festgelegt. Wo eine solche Beobachtung nicht möglich ist, werden Parameter aufgrund von Plausibilitätsüberlegungen und/oder stichpunktartiger Sensitivitätsanalysen einzelner Simulationstestläufe ermittelt. Dabei wird nicht auf die Erzwingung von (größeren) Störungen abgezielt. Ziel ist vielmehr ein möglichst reibungsarmer Betrieb, sowohl unter dem Fahrplan mit hoher Serviceregularität als auch unter dem Fahrplan mit niedriger Serviceregularität. So können auftretende Simulationsartefakte sowie durch die Fahrplanstruktur begründete Unterschiede in den erfassten Kennzahlen einfacher identifiziert werden. Letztere sollten dabei vergleichsweise moderat ausfallen, aber dennoch sichtbar sein. Darüber hinaus sollten sich trotz der notgedrungenen Oberflächlichkeit der Untersuchungen einige allgemeine Aussagen über die untersuchten Verkehrsnetze und die Zweckmäßigkeit der simulierten Fahrpläne tätigen lassen. Konkrete Handlungsempfehlungen für Verkehrsplaner lassen sich jedoch nicht abgeben.

Für die minimale Haltezeit  $C_f^b$  wird die beobachtete Zeitdauer verwendet, die die Bus- und Bahnfahrzeuge der *Kölner Verkehrs-Betriebe AG (KVB)* zum Öffnen und Schließen der Fahrzeugtüren benötigen; sie beträgt 12 Sekunden. Die Einstiegsrate  $B_f$  wird ebenfalls aufgrund eigener Beobachtungen ermittelt. Ohne Beeinträchtigungen aufgrund anderer Passagiere oder anderweitiger (körperlicher) Einschränkungen, wurde eine Zeit von drei Sekunden pro Person gemessen. Auf Basis dieser Beobachtungen und der in [28] sowie [98–100] zu findenden Fahrzeuginformationen, wurden zwei generische Fahrzeugmodelle erzeugt – jeweils eines für Busfahrzeuge und eines für Bahnfahrzeuge – die für die Simulationsläufe verwendet werden. Fahrplanunabhängige Passagierankunftsrate  $a_s$  für die einzelnen Netze werden festgelegt, indem Gleichung (5.7) nach  $a_s$  umgeformt wird:

$$a_s = \frac{T_{f,s} - C_f^b}{B_f * T_{L(f)}} \quad (5.9)$$

Für den Abstand aufeinanderfolgender Fahrzeuge  $T_{L(f)}$  wird die obere Schranke für den Sicherheitsabstand  $\bar{\delta}(s)$  an Haltepunkt  $s$  eingesetzt. Für  $C_f^b$  und  $B_f$  werden die beobachteten Werte verwendet. Die planmäßige Haltezeit  $T_{f,s}$  muss hingegen aufgrund operativer Überlegungen durch die Nutzerin festgesetzt werden.

Alle Experimente werden auf einem Intel Core i7-740QM Quad Core mit 1,73 GHz und 6 GB RAM ausgeführt.

### 5.3.1. UCL 2015

Netzwerk UCL 2015 wird, wie bereits in Abschnitt 3.4.1.2, aufgrund seiner realistischen Struktur bei gleichzeitig überschaubarer Größe für eine initiale Überprüfung der Plausibilität verwendet. Um moderate Verspätungseffekte aufgrund von Einstiegsprozessen zu erzielen, wird die planmäßige Haltezeit  $T_{f,s}$  für alle Haltepunkte auf 20 Sekunden gesetzt. Ausnahmen bilden die Endhaltepunkte 1012, 1081, 1161 und 1172. Da an Endhaltepunkten kein Zustieg von Passagieren erfolgt, wird für sie eine verringerte planmäßige Haltezeit von 15 Sekunden angenommen. Tabelle 5.4 zeigt die relevanten Parameter der einzelnen Haltepunkte und die sich daraus ergebenden Ankunftsrate. Darüber hinaus werden die jeweiligen Starthaltepunkte als Kontrollpunkte ausgewählt, an denen eine pünktliche Abfahrt von Fahrzeugen erzwungen wird; ihre Einträge in Tabelle 5.4 sind unterstrichen. Nicht aufgelistet ist die Haltepunktkapazität; sie entspricht für alle Haltepunkte zwei Fahrzeugen, um auch Überholvorgänge an Haltepunkten zu ermöglichen.

$s$	$\bar{\delta}(s)$ [Sek.]	$a_s$ [Pers./Sek.]	$T_{f,s}$ [Sek.]	$s$	$\bar{\delta}(s)$ [Sek.]	$a_s$ [Pers./Sek.]	$T_{f,s}$ [Sek.]
<u>1011</u>	600	0,0044	20	<b>1102</b>	180	0,0148	20
<b>1012</b>	600	0,0017	15	<b>1111</b>	180	0,0148	20
<b>1021</b>	600	0,0044	20	<b>1112</b>	180	0,0148	20
<b>1022</b>	600	0,0044	20	<b>1121</b>	180	0,0148	20
<u>1031</u>	180	0,0148	20	<b>1122</b>	180	0,0148	20
<u>1032</u>	180	0,0148	20	<b>1131</b>	180	0,0148	20
<b>1041</b>	300	0,0089	20	<b>1132</b>	180	0,0148	20
<b>1042</b>	300	0,0089	20	<b>1141</b>	300	0,0089	20
<b>1043</b>	300	0,0089	20	<b>1142</b>	180	0,0148	20
<b>1051</b>	300	0,0089	20	<b>1143</b>	180	0,0148	20
<b>1052</b>	300	0,0089	20	<b>1151</b>	300	0,0089	20
<b>1061</b>	180	0,0148	20	<b>1152</b>	300	0,0089	20
<b>1062</b>	180	0,0148	20	<b>1161</b>	300	0,0017	15
<b>1071</b>	600	0,0044	20	<b>1162</b>	300	0,0089	20
<b>1072</b>	600	0,0044	20	<u>1171</u>	300	0,0089	20
<b>1081</b>	600	0,0017	15	<b>1172</b>	300	0,0017	15
<u>1082</u>	600	0,0044	20	<b>1181</b>	300	0,0089	20
<b>1091</b>	180	0,0148	20	<b>1182</b>	300	0,0089	20
<b>1092</b>	180	0,0148	20	<b>1191</b>	300	0,0089	20
<b>1101</b>	180	0,0148	20	<b>1192</b>	300	0,0089	20

*Tabelle 5.4: Ankunftsdaten mit relevanten Berechnungsparametern, UCL 2015. Als Kontrollpunkte ausgewählte Haltepunkte sind unterstrichen*

Zur Festlegung der Fahrzeitparameter  $\gamma$  und  $\eta$  werden die Verbindungen des Netzwerks genauer betrachtet: Sie besitzen im Durchschnitt eine geplante Fahrzeit von ca. 120 Sekunden. Wie in Abschnitt 5.1.5.1 bereits erläutert, muss zur Bestimmung einer angemessenen durchschnittlichen Fahrzeit die geplante Haltezeit herausgerechnet werden. Mit der auf durchschnittlich 20 Sekunden festgelegten Haltezeit für Netzwerk UCL 2015 ergibt sich damit ein Wert von  $\gamma = 1 - \left(\frac{20}{120}\right) \approx 0,83$ . Zur Abbildung moderater Schwankungen der Fahrzeit wird außerdem ein Wert von  $\eta = 0,17$  gewählt. Damit entspricht die Standardabweichung auf einer Verbindung mit einer geplanten Fahrzeit von 120 Sekunden ca. 20 Sekunden. Da Netzwerk UCL 2015 keine Segmente enthält, wird  $\iota = 0$  gesetzt.

Die zu simulierenden Fahrpläne weisen, wie in Abschnitt 3.4.1.2, einen Takt von zehn Minuten für Linien 1-B01, 1-B02, 3-B01, 3-B02, 4-B01 und 4-B02, sowie einen Takt von 20 Minuten für Linien 2-B01 und 2-B02 auf. Der Optimalfahrplan UCL2015<sup>+</sup> hat einen Zielfunktionswert von 839 und einen netzwerkweiten Sicherheitsabstand von drei Minuten. Der schlechteste noch gültige Fahrplan UCL2015<sup>-</sup> erreicht hingegen nur einen Zielfunktionswert von 324 und einen netzwerkweiten Sicherheitsabstand von einer Minute. Die den einzelnen Linien unter den beiden Fahrplänen zugewiesenen Startzeiten  $\lambda_i$  sind in Tabelle 5.5 zu sehen.

Fahrplan	Linie							
	1-B01	1-B02	2-B01	2-B01	3-B01	3-B02	4-B01	4-B02
UCL2015 <sup>+</sup>	4	9	10	0	4	9	9	4
UCL2015 <sup>-</sup>	9	9	19	19	9	9	8	8

*Tabelle 5.5: Simulierte Fahrpläne für UCL 2015*

**Vergleich allgemeiner Merkmale.** Die durchschnittliche Verspätung pro verspäteter Abfahrt liegt unter Fahrplan UCL2015<sup>-</sup> bei 11,9 Sekunden, unter Fahrplan UCL2015<sup>+</sup> kann sie um 28,6 Prozent auf

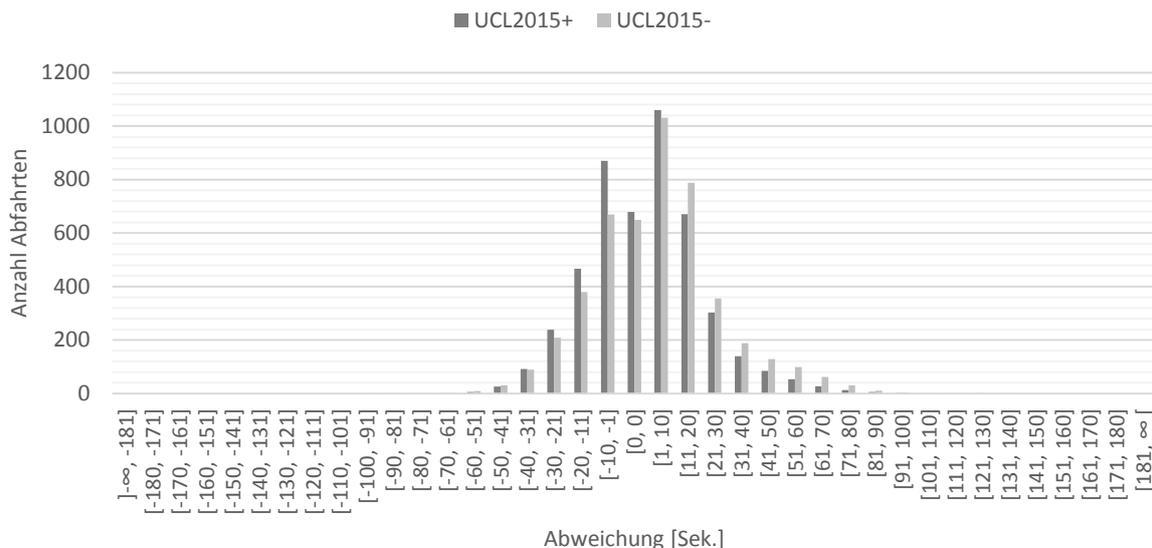


Abbildung 5.10: Häufigkeitsverteilung der Abfahrtszeitabweichungen, UCL 2015

8,5 Sekunden reduziert werden. Ein gegenteiliger Effekt kann bei frühzeitigen Abfahrten beobachtet werden. Unter Fahrplan UCL2015<sup>-</sup> finden frühzeitige Abfahrten im Durchschnitt 3,9 Sekunden zu früh statt, während diese Abweichung unter UCL2015<sup>+</sup> um 20,5 Prozent auf 4,7 Sekunden erhöht wird. Unter beiden Fahrplänen scheinen Abfahrten also eher zu spät als zu früh stattzufinden. Darüber hinaus scheint der Gewinn bei der durchschnittlichen Verspätung unter Fahrplan UCL2015<sup>+</sup> durch eine erhöhte Frühzeitigkeit erkauft zu werden.

Die Häufigkeitsverteilung der Abfahrtszeitabweichungen bestätigt diesen Effekt (siehe Abbildung 5.10). Sie misst wie viele im Messzeitraum stattfindende Abfahrten eine spezifische Abfahrtszeitabweichung aufweisen und zeigt für beide Fahrpläne eine höhere Zahl verspäteter als frühzeitiger Abfahrten. Außerdem zeigt sich, wie erwartet, dass Fahrplan UCL2015<sup>-</sup> im Vergleich zu Fahrplan UCL2015<sup>+</sup> eine erhöhte Zahl verspäteter Abfahrten aufweist, während unter Fahrplan UCL2015<sup>+</sup> eine erhöhte Zahl frühzeitiger Abfahrten zu beobachten ist. Insgesamt finden unter Fahrplan UCL2015<sup>-</sup> 2.699 von 4.740 Abfahrten (ca. 56,9%) verspätet statt, während unter Fahrplan UCL2015<sup>+</sup> nur 2.361 Abfahrten (ca. 49,8%) verspätet sind. Im Gegensatz dazu finden unter Fahrplan UCL2015<sup>-</sup> 1.391 (ca. 29,4%) aller Abfahrten zu früh statt, während unter Fahrplan UCL2015<sup>+</sup> 1.700 Abfahrten (ca. 35,9%) zu früh sind.

Ebenfalls wie erwartet, fällt das Gros der Abweichungen sehr moderat aus. Unter Fahrplan UCL2015<sup>+</sup> weisen 2.034 Abfahrten (ca. 42,9%) lediglich eine Verspätung von maximal 30 Sekunden auf und auch unter Fahrplan UCL2015<sup>-</sup> bildet diese Kategorie mit 2.175 Abfahrten (ca. 45,9%) den Großteil der verspäteten Abfahrten. Unter den zu früh stattfindenden Abfahrten findet sich ein ähnliches Muster: 1.575 (ca. 33,2%) aller Abfahrten finden unter Fahrplan UCL2015<sup>+</sup> bis zu 30 Sekunden zu früh statt, während unter Fahrplan UCL2015<sup>-</sup> 1.258 (ca. 26,5%) aller Abfahrten in diese Kategorie fallen.

Noch deutlicher wird dieser Effekt, wenn man die Zahl der Abfahrten mit einer Abweichung von mehr als 60 Sekunden betrachtet. Unter beiden Fahrplänen finden sich nahezu keine Abfahrten die mehr als 60 Sekunden zu früh stattfinden. Fahrplan UCL2015<sup>+</sup> weist im Durchschnitt eine

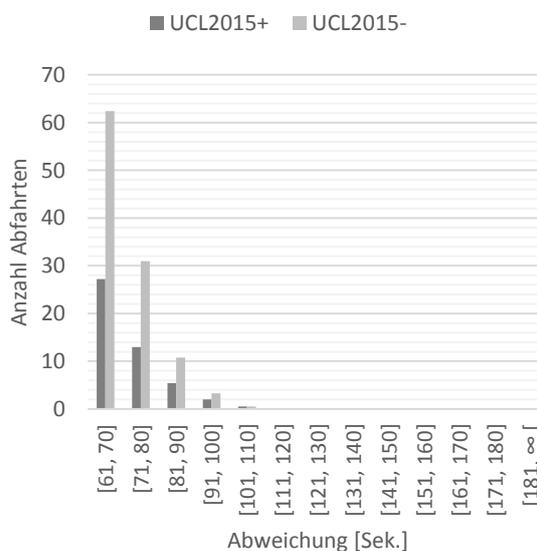


Abbildung 5.11: Häufigkeitsverteilung der Abfahrten mit einer Verspätung von mehr als 60 Sekunden, UCL 2015

einzelne frühzeitige Abfahrt mit einer Abweichung von mehr als 60 Sekunden auf, während sich unter Fahrplan UCL2015<sup>-</sup> durchschnittlich drei Abfahrten finden, die in diese Kategorie fallen. Für die Zahl der Abfahrten mit einer Verspätung von mehr als 60 Sekunden zeigt sich ein ähnliches Muster, wenn auch mit erhöhten Zahlen (siehe Abbildung 5.11): Unter Fahrplan UCL2015<sup>-</sup> weisen gerade einmal 108 (ca. 2,3%) aller Abfahrten eine Verspätung von mehr als 60 Sekunden auf; unter Fahrplan UCL2015<sup>+</sup> sind es sogar nur 48 (ca. 1,0%).

Die Zahl der exakt pünktlichen Abfahrten liegt mit 679 unter Fahrplan UCL2015<sup>+</sup> resp. 650 unter Fahrplan UCL2015<sup>-</sup> sehr nah beisammen und entspricht ca. 14,3 bzw. 13,7 Prozent. Dies ist zu erwarten, da für beide Fahrpläne die gleichen Kontrollpunkte verwendet werden, an denen immer zum im Fahrplan angegebenen Zeitpunkt abgefahren wird.

Es lässt sich also festhalten, dass unter beiden Fahrplänen, wie erhofft, nur sehr geringe Abweichungen der Abfahrtszeiten auftreten. Außerdem finden Abfahrten allgemein eher zu spät als zu früh statt. Unter Fahrplan UCL2015<sup>-</sup> ist dieses Phänomen dabei stärker ausgeprägt als unter Fahrplan UCL2015<sup>+</sup>. Auch dies ist zu erwarten, da die gewählten Ankunftsdaten auf Basis der theoretisch optimalen Sicherheitsabstände  $\bar{\delta}(s)$  bestimmt wurden. Während Fahrplan UCL2015<sup>+</sup> häufig an diese Sicherheitsabstände herankommt, weist Fahrplan UCL2015<sup>-</sup> sehr unregelmäßige Sicherheitsabstände auf. In Folge dessen weichen die Haltezeiten unter diesem Fahrplan stärker von den Planwerten ab und ziehen häufiger eine Verzögerung der Fahrzeuge nach sich.

Linie	Frühzeitigkeit [Sek.]		Verspätung [Sek.]	
	UCL2015 <sup>+</sup>	UCL2015 <sup>-</sup>	UCL2015 <sup>+</sup>	UCL2015 <sup>-</sup>
1-B01	1,9	3,7	8,4	6,1
1-B02	1,6	1,1	14,7	11,8
2-B01	4,6	11,2	4,4	2,2
2-B02	12,2	8,7	0,8	5,6
3-B01	5,9	3,3	11,4	21,1
3-B02	5,4	6,3	4,4	5,6
4-B01	6,2	1,1	7,6	17,3
4-B02	2,0	2,2	8,4	9,6
∅	5,0	4,7	7,5	9,9

Tabelle 5.6: Durchschnittliche Frühzeitigkeit und Verspätung pro Linie, UCL 2015

**Vergleich von Linien.** Für eine detaillierte Untersuchung des Simulationsverhaltens werden die einzelnen Linien des Netzwerks genauer betrachtet. Tabelle 5.6 zeigt die durchschnittliche Abfahrtszeitabweichung, aufgeteilt in Frühzeitigkeit und Verspätung. Sie werden berechnet, indem zunächst für jede Fahrt die durchschnittliche Frühzeitigkeit/Verspätung über ihre Abfahrten berechnet wird. Anschließend wird der Durchschnittswert einer Linie über die entsprechenden Durchschnitte aller ihrer Fahrten gebildet. Die resultierenden Werte bestätigen die bereits anhand der allgemeinen Merkmale identifizierten Muster: Abfahrten finden unabhängig vom Fahrplan eher zu spät als zu früh statt und unter Fahrplan UCL2015<sup>-</sup> ist dieser Effekt stärker als unter Fahrplan UCL2015<sup>+</sup>. Außerdem fallen die durchschnittlichen Abfahrtszeitabweichungen mit maximal 21,5 Sekunden, wie erhofft, moderat aus.

Die bisher untersuchten Kennzahlen lassen vermuten, dass Fahrplan UCL2015<sup>+</sup> unter den gewählten Simulationsparametern eine bessere Systemperformanz gewährleistet. Allerdings profitieren nicht alle Linien von ihm. Unter Fahrplan UCL2015<sup>+</sup> kann die durchschnittliche Verspätung der Linien 2-B02 bis 4-B02 im Vergleich zu Fahrplan UCL2015<sup>-</sup> zwischen 12,5 Prozent (4-B02) und 85,7 Prozent (2-B02) reduziert werden, während sie für Linien 1-B01 bis 2-B01 zwischen 24,6 Prozent (1-B02) und 100 Prozent (2-B01) erhöht wird. Darüber hinaus kann die reduzierte durchschnittliche Verspätung von keinem der beiden Fahrpläne durchgehend in einen Gewinn an Pünktlichkeit umgewandelt werden, sondern wird z.T. durch eine Steigerung der Frühzeitigkeit erzielt. Dies wird besonders bei den Linien 2-B01/B02, 3-B01 und 4-B01 deutlich, deren Frühzeitigkeit im Vergleich zum jeweils besseren Fahrplan um 143,5 Prozent, 40,2 Prozent, 78,8 Prozent und 463,6 Prozent steigt.

Da sich – wie bspw. Schüle et al in [80] schreiben – zu frühe und zu späte Abfahrten gleichermaßen negativ auf die wahrgenommene Angebotsqualität auswirken, indem sie eventuelle Umsteigeverbindungen in gleicher Weise gefährden, reichen die bisher betrachteten Kennzahlen alleine nur bedingt zur Bewertung der Güte der Fahrpläne – und der Plausibilität der Simulationsanwendung – aus. Stattdessen muss eine detaillierte Untersuchung des Systemverhaltens erfolgen.

Hierzu werden zunächst die Häufigkeitsverteilungen der Abfahrtszeitabweichungen für einzelne Linien herangezogen (siehe Abbildung 5.12 bis Abbildung 5.15). Sie bestätigen für die Mehrheit der Linien die bisherigen Beobachtungen und zeigen häufig unter beiden Fahrplänen ähnliche Verteilungsmuster: Die unter Fahrplan UCL2015<sup>+</sup> stärker verspäteten Linien 1-B01/B02 und 2-B01 besitzen im Vergleich zu Fahrplan UCL2015<sup>-</sup> einen größeren Anteil verspäteter Abfahrten und – mit Ausnahme von Linie 1-B02 – einen geringeren Anteil frühzeitiger Abfahrten. Letztere besitzt unter Fahrplan UCL2015<sup>+</sup> neben dem erhöhten Anteil leicht verspäteter Abfahrten auch einen erhöhten Anteil von Abfahrten, die mindestens 20 Sekunden zu früh stattfinden. Diese Streuung lässt darauf schließen, dass Fahrten dieser Linie unter Fahrplan UCL2015<sup>+</sup> – bspw. aufgrund von Effekten ungewollter Kolonnenfahrten – stärkeren Schwankungen unterliegen. Ein ähnlicher Effekt lässt sich für Linien 3-B02 und 4-B02 unter Fahrplan UCL2015<sup>-</sup> beobachten.

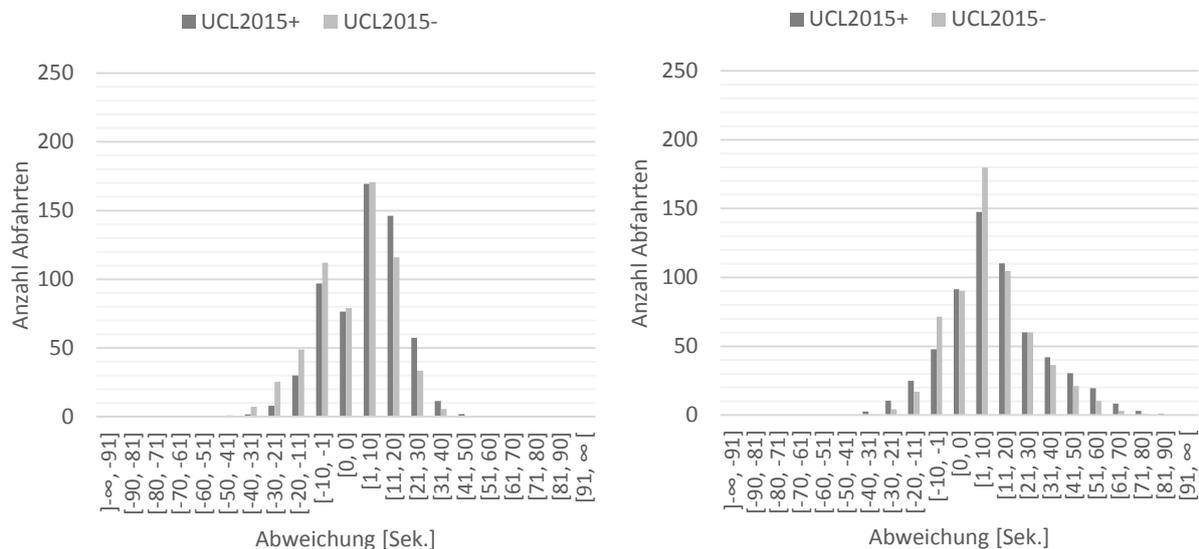


Abbildung 5.12: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 1. Links: 1-B01, rechts: 1-B02

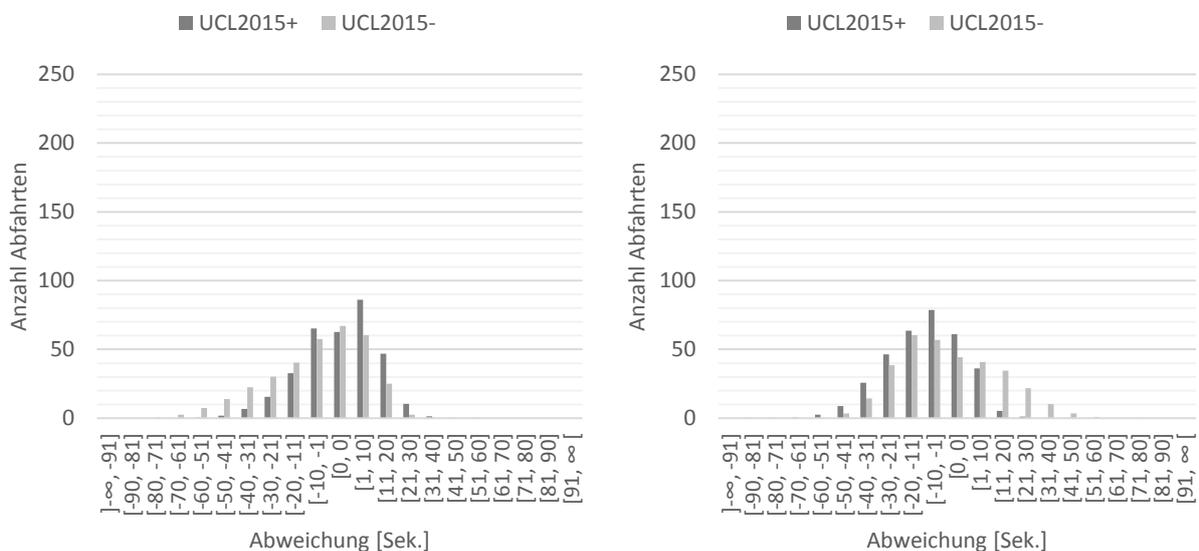


Abbildung 5.13: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 2. Links: 2-B01, rechts: 2-B02

Deutlichere Abweichungen in den Häufigkeitsverteilungen der Abfahrtszeitabweichungen zwischen den beiden Fahrplänen zeigen sich bei Linien 2-B02, 3-B01 und 4-B01, wie die starken Unterschiede in Frühzeitigkeit und Verspätung in Tabelle 5.6 bereits vermuten ließen. Es liegt nahe, dass der höhere Anteil verspäteter Abfahrten auf diesen Linien unter Fahrplan UCL2015<sup>-</sup> aufgrund der schlechteren Serviceregularität und den in Folge dessen verlängerten Haltezeiten entsteht. Unter Fahrplan UCL2015<sup>+</sup> fällt die Zahl verspäteter Abfahrten hingegen wesentlich geringer aus, wobei dies bei Linien 3-B01 und 4-B01 auf einen erhöhten Anteil (leicht) frühzeitiger Abfahrten zurückzuführen ist.

Fahrplanunabhängig sind außerdem die Häufigkeitsverteilungen der Linien 2-B01 und 2-B02 auffällig (siehe Abbildung 5.13). Sie weisen aufgrund ihres geringeren Taktes eine geringere Gesamtzahl an Abfahrten auf, weswegen die Kurven ihre Häufigkeitsverteilungen flacher ausfallen. Dennoch besitzen sie, abhängig vom Fahrplan, die höchste durchschnittliche Abfahrtszeitabweichung bei frühzeitigen Abfahrten (vgl. Tabelle 5.6). Grund dafür ist, dass die beiden Linien größere Anteile leicht erhöhter Abfahrtszeitabweichungen von mehr als 30 Sekunden aufweisen. Unter Fahrplan UCL2015<sup>+</sup> sind 38 von 330 Abfahrten (ca. 11,4%) von Linie 2-B02 mehr als 30 Sekunden zu früh, wohingegen unter Fahrplan UCL2015<sup>-</sup> nur 18 Abfahrten (ca. 5,5%) eine erhöhte Abweichung aufweisen. Unter diesem besitzt hin-

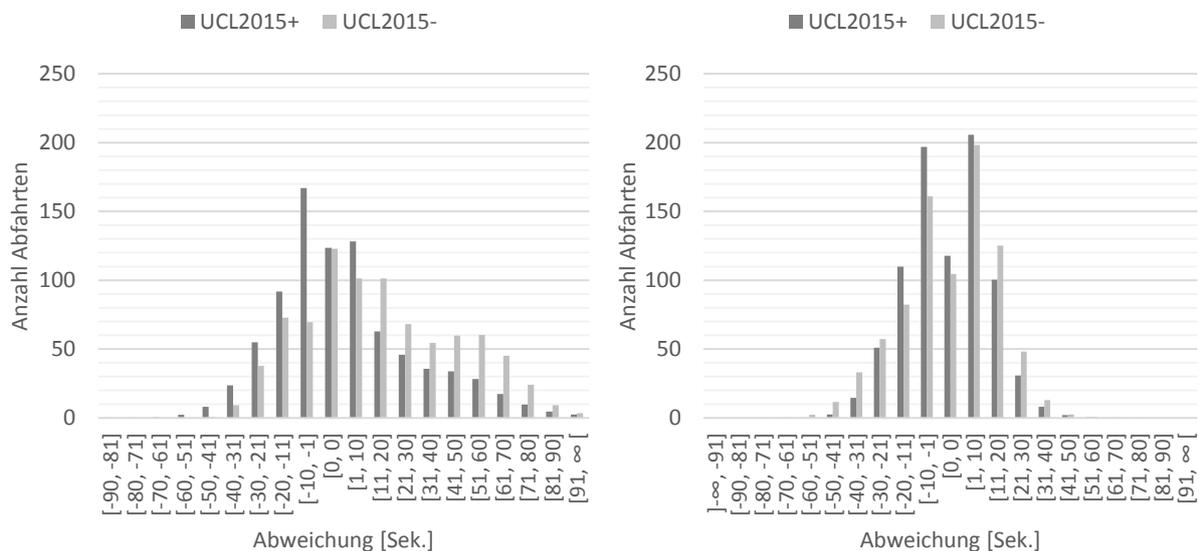


Abbildung 5.14: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 3. Links: 3-B01, rechts: 3-B02

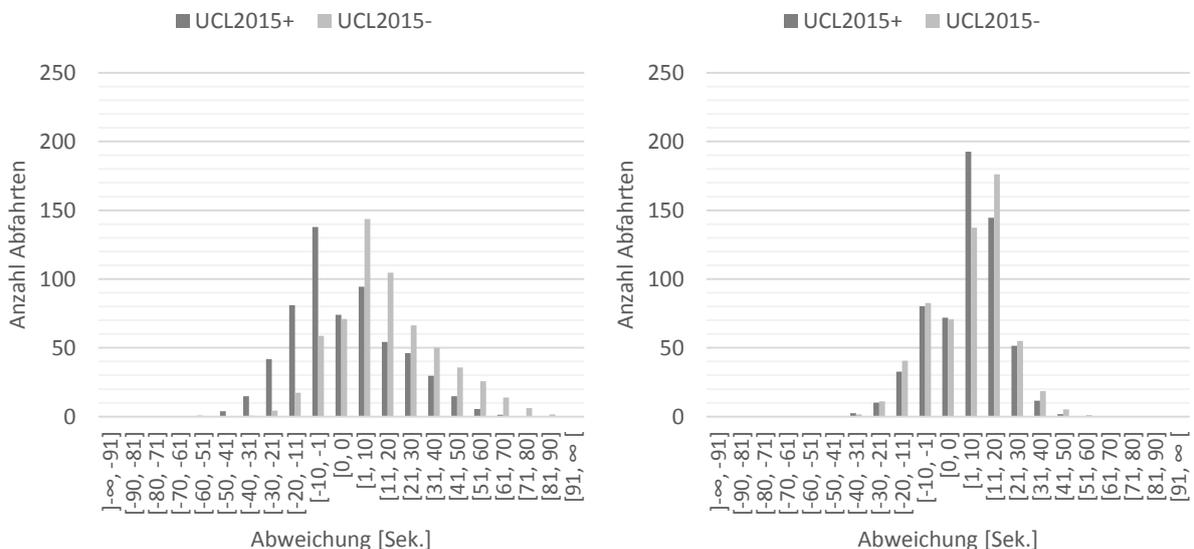


Abbildung 5.15: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Liniengruppe 4. Links: 4-B01, rechts: 4-B02

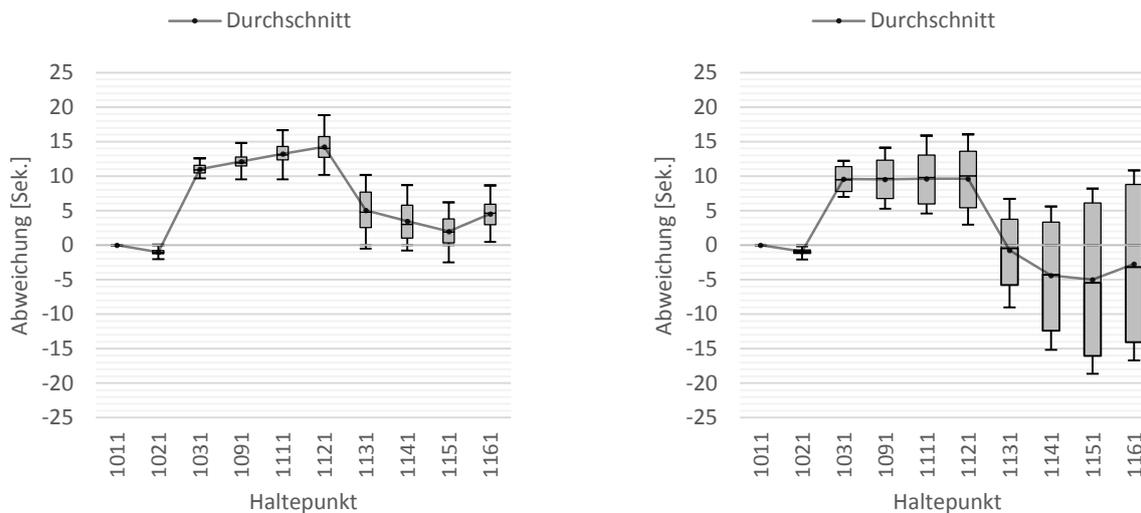


Abbildung 5.16: Abweichungsverlauf, Linie 1-B01. Links: UCL2015+, rechts: UCL2015-

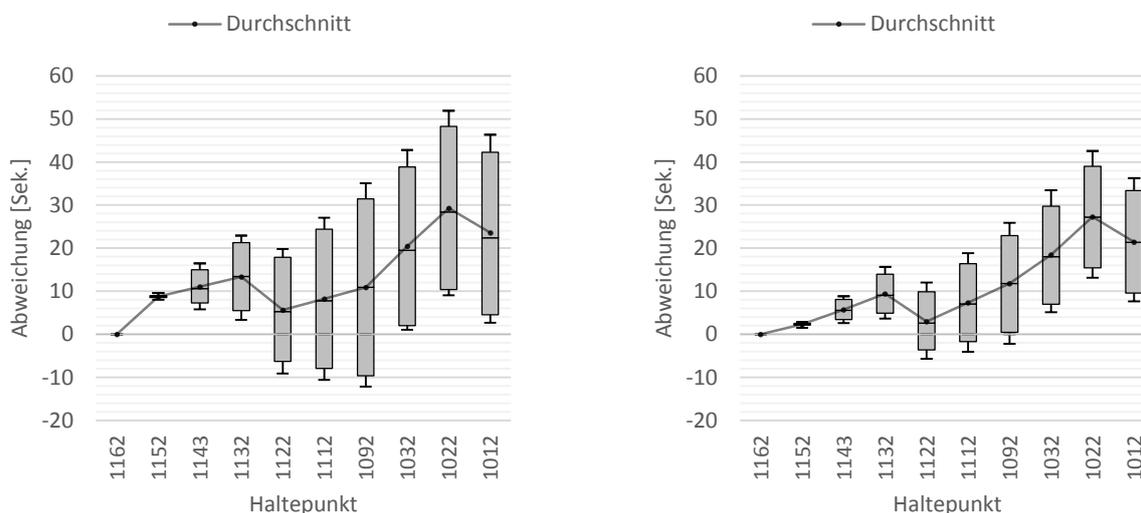


Abbildung 5.17: Abweichungsverlauf, Linie 1-B02. Links: UCL2015+, rechts: UCL2015-

gegen Linie 2-B01 mit 47 Abfahrten (ca. 14,2%) einen größeren Anteil leicht erhöhter frühzeitiger Abweichungen, während unter Fahrplan UCL2015+ nur 9 Abfahrten (ca. 2,7%) von Linie 2-B01 mehr als 30 Sekunden zu früh stattfinden.

Um präzisere Aussagen über potentielle Problemstellen und die Plausibilität des Simulationsmodells zu treffen, werden die Abweichungsverläufe der einzelnen Linien betrachtet. Dazu werden über den Simulationsverlauf die Abfahrtszeitabweichungen der einzelnen Fahrten an ihren jeweiligen Haltepunkten erfasst und über die durchgeführten Simulationsläufe gemittelt. Anschließend werden für die einzelnen Linien Durchschnitt, Median, 25%- und 75%-Quantil sowie Minimum und Maximum der Abfahrtszeitabweichung an jedem Haltepunkt berechnet. Die sich daraus ergebenden Boxplot-Diagramme werden in Abbildung 5.16 bis Abbildung 5.23 dargestellt.

Der Abweichungsverlauf für Linie 1-B01 weist unter beiden Fahrplänen ein ähnliches Muster auf (siehe Abbildung 5.16). Den Höchstwert ihrer durchschnittlichen Verspätung erreicht Linie 1-B01 unter beiden Fahrplänen an Haltepunkt 1121, drei Haltepunkte nachdem sie zu den Linien 2-B02 und 3-B01 hinstößt. Unter Fahrplan UCL2015+ besitzt sie dort eine Verspätung von 14,2 Sekunden, während sie unter Fahrplan UCL2015- eine geringere Verspätung von 9,6 Sekunden einfährt. Der Grund für die höhere maximale Verspätung unter Fahrplan UCL2015+ ist dabei der unter Fahrplan UCL2015+ nicht mit dem (theoretischen) Optimum übereinstimmende, größere Abstand zwischen Fahrzeugen der Linie 1-

B01 und ihren jeweiligen Vorgängerfahrzeugen zwischen den Haltepunkten 1031 und 1131. Unter Fahrplan UCL2015<sup>+</sup> beträgt dieser Abstand vier Minuten, während er unter Fahrplan UCL2015<sup>-</sup> mit drei Minuten dem zur Festlegung der Passagierankunftsdaten verwendeten Optimalwert entspricht. In der Folge unterliegen Fahrzeuge der Linie 1-B01 unter Fahrplan UCL2015<sup>+</sup> zwischen den Haltepunkten 1031 und 1131 sehr leichten, systematischen Bunching-Effekten, wodurch sie leicht höhere Haltezeiten von durchschnittlich ca. 22 Sekunden zugewiesen bekommen. Unter Fahrplan UCL2015<sup>-</sup> betragen die Haltezeiten für Linie 1-B01 an den Haltepunkten hingegen durchschnittlich 20 Sekunden. Da die planmäßigen Fahrzeiten zwischen den Haltepunkten 1031 und 1121 jeweils zwei Minuten – und damit dem zur Bestimmung der durchschnittlichen Fahrzeit verwendeten Planwert – entsprechen, bleibt die Verspätung der Linie 1-B01 unter Fahrplan UCL2015<sup>-</sup> auf diesem Streckenabschnitt konstant, während sie unter Fahrplan UCL2015<sup>+</sup> pro Haltepunkt um ca. 2 Sekunden steigt.

Als Zwischenfazit lässt sich an dieser Stelle festhalten, dass das Simulationsmodell das durch die Fahr- und Haltezeitparameter festgelegte Verhalten korrekt abbildet. Aufgrund der Wahl der Parameter sollten Fahrzeuge, die auf einem Streckenabschnitt sowohl der durchschnittlichen (geplanten) Fahrzeit des Gesamtsystems als auch der durchschnittlichen Haltezeit unterliegen, auf diesem Streckenabschnitt keine signifikante Abfahrtszeitabweichung einfahren. Dies ist für Linie 1-B01 und Fahrplan UCL2015<sup>-</sup> offensichtlich der Fall.

Das umgedrehte Phänomen lässt sich zwischen den Haltepunkten 1141 und 1161 beobachten: Hier weisen Fahrzeuge von Linie 1-B01 unter Fahrplan UCL2015<sup>-</sup> nur einen Abstand von einer Minute zu ihren Vorgängerfahrzeugen der Linie 4-B01 auf, während sie unter Fahrplan UCL2015<sup>+</sup> in einem optimalen planmäßigen Abstand von fünf Minuten abfahren. In der Folge unterliegen Fahrzeuge von Linie 1-B01 unter Fahrplan UCL2015<sup>-</sup> wesentlich kürzere Haltezeiten, wodurch sie auf diesem Streckenabschnitt durchschnittlich bis zu 5 Sekunden zu früh abfahren. Dass auch unter Fahrplan UCL2015<sup>+</sup> trotz des optimalen Fahrzeugabstands eine extrem leichte Reduktion der Verspätung beobachtet werden kann lässt sich dadurch erklären, dass Fahrzeuge der Linie 4-B01 ab Haltepunkt 1041 einer verstärkten Streuung der Abfahrtszeitabweichung unterliegen (vgl. Abbildung 5.22), sodass es zu leichten Bunching-Effekten zwischen den Fahrzeugen der beiden Linien kommt.

Zwei weitere interessante Phänomene sind zwischen den Haltepunkten 1021 und 1031 sowie 1121 und 1131 zu beobachten: Hier zeigt sich ein systematischer Anstieg resp. Abfall der Verspätung um ca. 10 Sekunden. Grund dafür ist, dass die planmäßige Fahrzeit dort mit 60 resp. 180 Sekunden für die durchschnittliche Haltezeit etwas zu hoch bzw. zu gering ist, sodass sich Fahrzeuge ohne Interaktion mit Fahrzeugen anderer Linien systematisch leicht Verspäten bzw. dazu in der Lage sind ihre eingefahrene Verspätung zu reduzieren.

Abbildung 5.16 lässt noch eine weitere Beobachtung zu: Die Spannweite zwischen dem 25%- und 75%-Quantil nimmt für beide Fahrpläne zu, sobald Linie 1-B01 an Haltepunkt 1031 zu den Linien 2-B02 und 3-B01 stößt. Unter Fahrplan UCL2015<sup>-</sup> wächst sie dabei stärker als unter Fahrplan UCL2015<sup>+</sup> und nimmt auch noch zu, nachdem sich Linie 1-B01 an Haltepunkt 1131 von Linien 2-B02 und 3-B01 trennt. Ein gewisser Zuwachs in der Spannweite der Abfahrtszeitabweichungen ist dabei zu erwarten, da Linie 1-B01 durch das Hinzustoßen der anderen beiden Linien einer größeren Dynamik in den Haltezeiten unterliegt. Dieser Effekt wird durch die unterschiedliche Taktung von Linie 2-B02 allerdings noch verstärkt: Da Linie 2-B02 nur alle 20 Minuten fährt, bekommen Fahrzeuge der Nachfolgerlinie regelmäßig längere Haltezeiten zugewiesen, nämlich genau dann, wenn gerade kein Fahrzeug von Linie 2-B02 fährt. Unter Fahrplan UCL2015<sup>+</sup> ist dies für Fahrzeuge der Linie 3-B01 der Fall. Fährt ein Fahrzeug der Linie 2-B02, so beträgt der planmäßige Sicherheitsabstand zwischen ihm und seinem Nachfolgerfahrzeug der Linie 3-B01 drei Minuten. Fährt kein Fahrzeug der Linie 2-B02, verdoppelt sich der planmäßige Abstand zu sechs Minuten auf das vorwegfahrende Fahrzeug der Linie 1-B01. In der Folge vergrößert sich die Haltezeit für das Fahrzeug der Linie 3-B01 systematisch, wodurch sich wiederum der Abstand zum nachfolgenden Fahrzeug der Linie 1-B01 verkürzt und dieses eine kürzere Haltezeit zugewiesen bekommt (siehe hierzu auch Abbildung 5.20). Unter Fahrplan UCL2015<sup>-</sup> tritt der gleiche Effekt in einer komplexeren Konstellation auf: Zunächst fahren Fahrzeuge der Linie 3-B01 planmäßig alle zehn Minuten mit einem Abstand von sechs Minuten hinter Fahrzeugen der Linie 1-B01 an Halte-

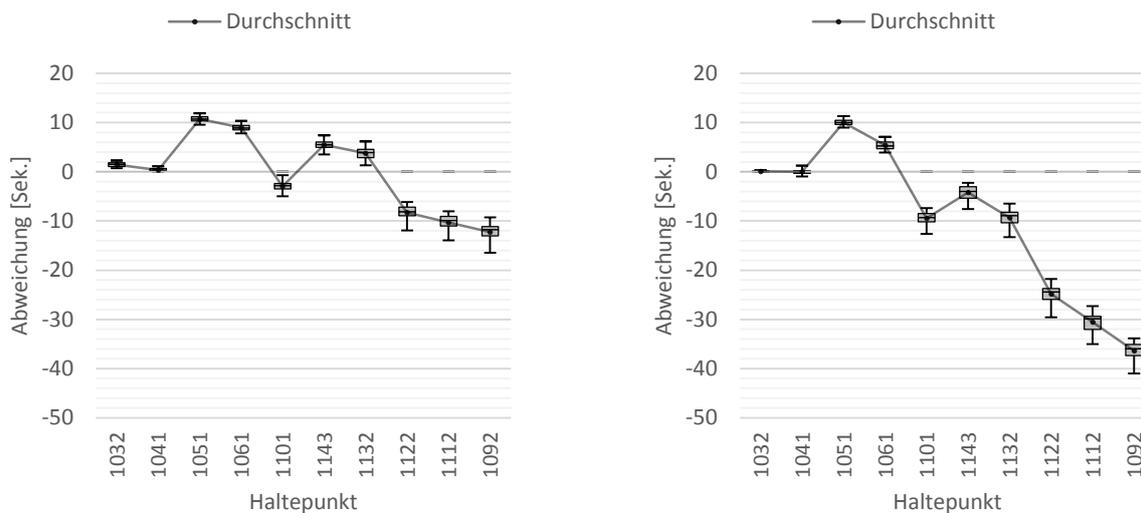


Abbildung 5.18: Abweichungsverlauf, Linie 2-B01. Links: UCL2015<sup>+</sup>, rechts: UCL2015<sup>-</sup>

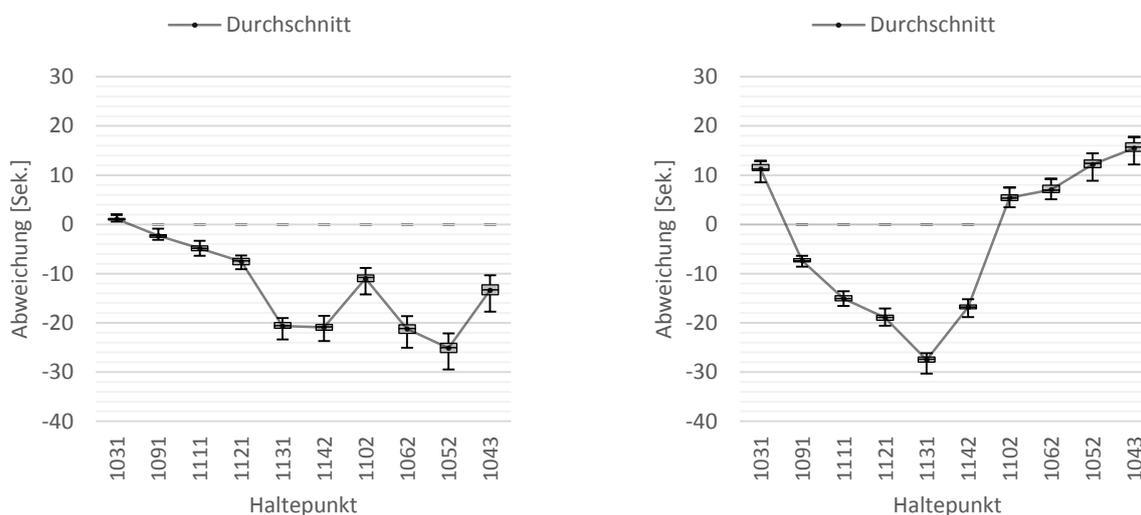


Abbildung 5.19: Abweichungsverlauf, Linie 2-B02. Links: UCL2015<sup>+</sup>, rechts: UCL2015<sup>-</sup>

punkt 1031 ab; sie unterliegen unter Fahrplan UCL2015<sup>-</sup> entsprechend systematisch einer längeren Haltezeit. Zusätzlich fährt alle 20 Minuten ein Fahrzeug der Linie 2-B02 planmäßig eine Minute nach einem Fahrzeug der Linie 3-B01 ab, sie unterliegen entsprechend systematisch einer verkürzten Haltezeit. Schließlich fahren Fahrzeuge der Linie 1-B01 drei Minuten nach Fahrzeugen der Linie 2-B02 bzw. vier Minuten nach Fahrzeugen der Linie 3-B01 ab. Als Resultat unterliegen die Abfahrtszeiten von Fahrzeugen der Linie 1-B01 stärkeren Schwankungen auf dem gemeinsamen Streckenabschnitt, während Fahrzeuge der Linie 3-B01 dort stärkere Verspätungen aufweisen (vgl. Abbildung 5.20) und Fahrzeuge der Linie 2-B02 häufig zu früh abfahren (vgl. Abbildung 5.19). Ein ähnlicher Effekt tritt unter Fahrplan UCL2015<sup>-</sup> auch zwischen den Linien 1-B01 und 4-B01 auf ihrem gemeinsamen Streckenabschnitt ab Haltepunkt 1141 auf, wodurch sich die im Gegensatz zu Fahrplan UCL2015<sup>+</sup> weiter wachsende Spannweite zwischen den Quantilen erklärt.

Der Abweichungsverlauf von Linie 1-B02 zeigt unter beiden Fahrplänen ebenfalls ähnliche Tendenzen (siehe Abbildung 5.17): Zwischen den Haltepunkten 1132 und 1122 kann aufgrund der großzügig geplanten Fahrzeit von 180 Sekunden systematisch Zeit gutgemacht werden, während die geplante Fahrzeit von 60 Sekunden zwischen den Haltepunkten 1032 und 1022 für die geplante Haltezeit von 20 Sekunden zu knapp bemessen ist. Zudem nimmt auch hier die Spannweite zwischen dem 25%- und 75%-Quantil ab Haltepunkt 1143 aufgrund des 20-Minuten-Taktes von Linie 2-B01 zu. Unter beiden

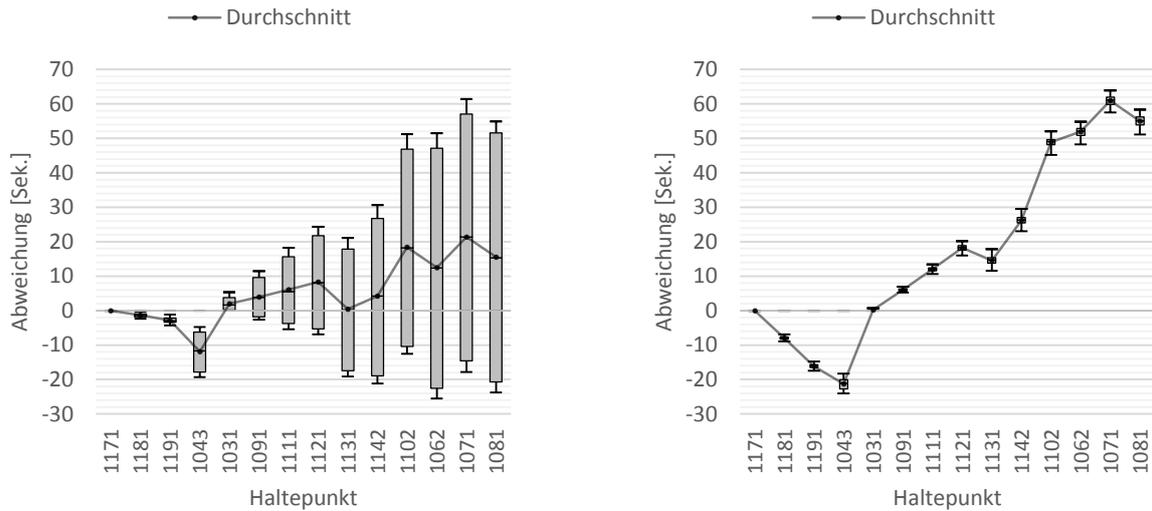


Abbildung 5.20: Abweichungsverlauf, Linie 3-B01. Links: UCL2015+, rechts: UCL2015-

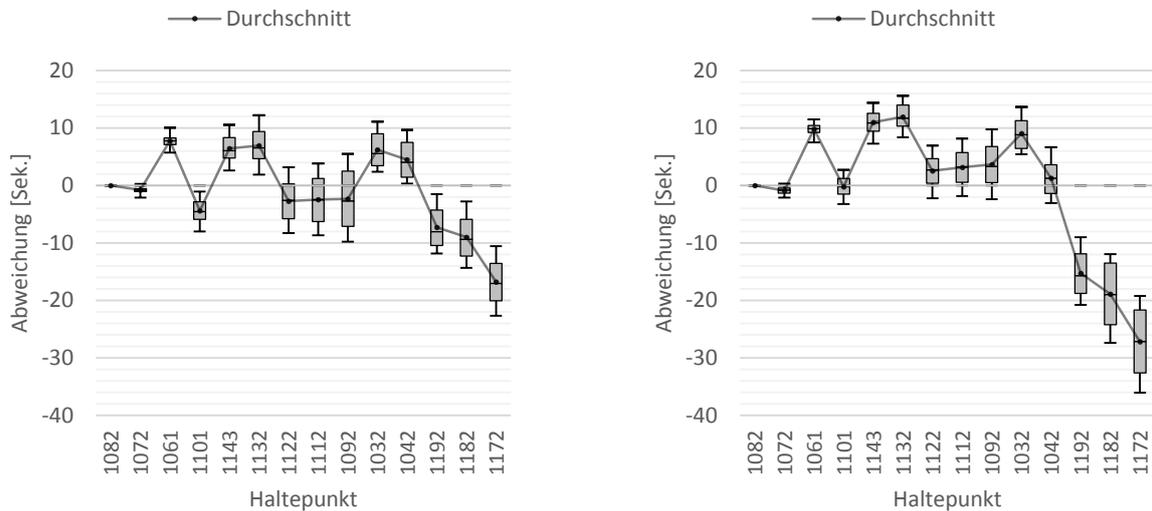


Abbildung 5.21: Abweichungsverlauf, Linie 3-B02. Links: UCL2015+, rechts: UCL2015-

Fahrplänen verkehren Fahrzeuge von Linie 1-B02 dort planmäßig sieben Minuten nach Fahrzeugen der Linie 3-B02 und drei bzw. vier Minuten nach Fahrzeugen der Linie 2-B01. Die unter Fahrplan UCL2015+ zu beobachtende größere Spannweite der Abfahrtszeitabweichungen muss sich dabei durch die unter diesem Fahrplan bereits an Haltepunkt 1152 eingefahrene leichte Verspätung erklären. Hier liegt auch der größte Unterschied zwischen den Abweichungsverläufen unter den beiden Fahrplänen. Da Fahrzeuge von Linie 1-B02 unter Fahrplan UCL2015+ an Haltepunkt 1152 fünf Minuten nach Fahrzeugen von Linie 4-B02 abfahren, erhalten Sie in der Regel die in Tabelle 5.4 angegebene geplante Haltezeit von 20 Sekunden. Aufgrund der mit 60 Sekunden bemessenen geplanten Fahrzeit zwischen Haltepunkt 1162 und Haltepunkt 1152 ergibt sich damit auch hier eine leichte systematische Verspätung. Unter Fahrplan UCL2015- fahren Fahrzeuge von Linie 1-B02 hingegen eine Minute nach Fahrzeugen der Linie 4-B02 ab und erhalten folglich eine wesentlich kürzere Haltezeit, sodass sich die knappe Fahrzeit nicht negativ auswirkt.

Die Abweichungsverläufe der Linien 2-B01 und 2-B02 (siehe Abbildung 5.18 und Abbildung 5.19) weisen unter beiden Fahrplänen sehr geringe Spannweiten zwischen den 25%- und 75%-Quantilen auf. Dies ist zu erwarten, da ihre Haltezeiten im Gegensatz zu denen der anderen Linien nicht durch unregelmäßige Abfahrten anderer Linien beeinflusst werden. Linie 2-B01 zeigt, wie Linie 1-B02, unter beiden Fahrplänen ähnliche Tendenzen im Abweichungsverlauf, unterliegt unter Fahrplan UCL2015- aber

leicht stärkeren Bunching-Effekten sobald sie an Haltepunkt 1061 mit Linie 3-B02 zusammenfährt. Diese Effekte verstärken sich sukzessive auf dem restlichen gemeinsamen Linienverlauf bis Haltepunkt 1092, an dem Linie 2-B01 unter Fahrplan UCL2015<sup>+</sup> im Durchschnitt 12,2 Sekunden und unter Fahrplan UCL2015<sup>-</sup> 36,4 Sekunden zu früh abfährt. Linie 2-B02 zeigt unter Fahrplan UCL2015<sup>-</sup> das gleiche Phänomen zwischen ihrem Starthaltepunkt 1031 und Haltepunkt 1131. Ab dort ändert sich ihr Abweichungsverlauf unter Fahrplan UCL2015<sup>-</sup> jedoch deutlich: Während Sie unter Fahrplan UCL2015<sup>+</sup> den Trend der frühzeitigen Abfahrten fortsetzt und lediglich aufgrund der knappen geplanten Fahrzeiten zwischen den Haltepunkten 1142 und 1102 bzw. 1052 und 1043 Teile des herausgefahrenen Puffers einbüßt, verliert sie unter Fahrplan UCL2015<sup>-</sup> nach Haltepunkt 1131 ihren gesamten Vorsprung und beginnt sich zu verspäten. Grund dafür ist, dass Fahrzeuge von Linie 2-B02 die eine Minute vorausfahrenden Fahrzeuge von Linie 3-B01 zwischen Haltepunkt 1031 und Haltepunkt 1102 ein- und zum Teil kurzzeitig überholen. In Folge von Überholvorgängen erhöht sich die Haltezeit für Fahrzeuge von Linie 2-B02, da zu ihrer Berechnung anstatt des Abstands zu Fahrzeugen der Linie 3-B01 nun die wesentlich größeren Abstände zu Fahrzeugen der Linien 1-B01 (planmäßig sechs Minuten ab Haltepunkt 1031) bzw. 4-B02 (planmäßig acht Minuten ab Haltepunkt 1142) verwendet werden. Der dadurch eingebüßte Vorsprung kann anschließend nicht wieder gutgemacht werden. Auffällig ist, dass sich solche Änderungen der Fahrzeugreihenfolge langfristig nicht positiv auf den Abweichungsverlauf der Linie 3-B01 auswirken (siehe Abbildung 5.20). Das liegt daran, dass die Reihenfolgeänderungen zwischen Fahrzeugen der Linien 2-B02 und 3-B01 häufig nur von kurzer Dauer sind und zum Teil bereits nach einem Haltepunkt ein erneuter Reihenfolgewechsel stattfindet. Während Fahrzeuge von Linie 2-B02 durch einen Überholvorgang zwar vor Fahrzeugen der Linie 3-B01 am nächsten Haltepunkt ankommen, büßen sie diesen Vorsprung durch ihre längere Haltezeit wieder ein. Fahrzeuge von Linie 3-B01 erhalten hingegen eine wesentlich kürzere Haltezeit, wodurch die ursprüngliche Fahrzeugreihenfolge wiederhergestellt werden kann.

Zu erwähnen ist an dieser Stelle noch, dass die an Haltepunkt 1031 zu beobachtenden Abweichungen von Linien 2-B01 und 2-B02 aufgrund der Zirkularität des Linienverlaufs sowohl die initialen als auch die am Ende einer Fahrt eingefahrenen Abweichungen umfassen.

Neben Linie 2-B02 weist Linie 3-B01 die größten Unterschiede im Abweichungsverlauf zwischen den beiden Fahrplänen auf (siehe Abbildung 5.20). Die zu beobachtenden Phänomene wurden bereits erläutert. Interessant ist die Abfahrtszeitabweichung an Haltepunkt 1031. Er ist als Starthaltepunkt von Linie 2-B02 ein Kontrollpunkt, an dem nicht vor der im Fahrplan angegebenen Abfahrtszeit losgefahren wird. Entsprechend wird an ihm keine signifikante Abweichung von der Haltezeit gemessen.

Der Abweichungsverlauf von Linie 3-B02 zeigt unter beiden Fahrplänen ebenfalls das gleiche Muster (siehe Abbildung 5.21). Unter Fahrplan UCL2015<sup>-</sup> fällt lediglich die maximale Frühzeitigkeit stärker aus. Sie wird unter beiden Fahrplänen am Endhaltepunkt 1172 erreicht. Unter Fahrplan UCL2015<sup>+</sup> wird dort eine durchschnittliche Frühzeitigkeit von 16,8 Sekunden gemessen, während unter Fahrplan UCL2015<sup>-</sup> ein Durchschnittswert von 27,2 Sekunden erreicht wird. Wie bei den anderen Linien nimmt die Spannweite zwischen den Quantilen auf den gemeinsamen Streckenabschnitten mit Linie 2-B01 aufgrund der unterschiedlichen Taktung zu.

Linie 4-B01 zeigt unter beiden Fahrplänen ebenfalls ein ähnliches Muster des Abweichungsverlaufs (siehe Abbildung 5.22). Unter Fahrplan UCL2015<sup>+</sup> fahren ihre Fahrzeuge mit einem Abstand von fünf Minuten zu Fahrzeugen der Linie 3-B01 vom Starthaltepunkt 1171 los und können so bis Haltepunkt 1041 von den großzügig geplanten Fahrzeiten profitieren. Im Gegensatz dazu fahren Fahrzeuge von Linie 4-B01 unter Fahrplan UCL2015<sup>-</sup> neun Minuten nach Fahrzeugen der Linie 3-B01 an Haltepunkt 1171 los, erhalten folglich erhöhte Haltezeiten und verspäten sich systematisch. Gegen Ende des Linienverlaufs, ab Haltepunkt 1141, tritt dieser Effekt erneut auf. Da Fahrzeuge von Linie 1-B01 dort, wie bereits beschrieben, eine Minute nach Fahrzeugen von Linie 4-B01 abfahren, erhalten sie stark verkürzte Haltezeiten (siehe Abbildung 5.16), während Fahrzeuge von Linie 4-B01 erhöhte Haltezeiten zugewiesen bekommen.

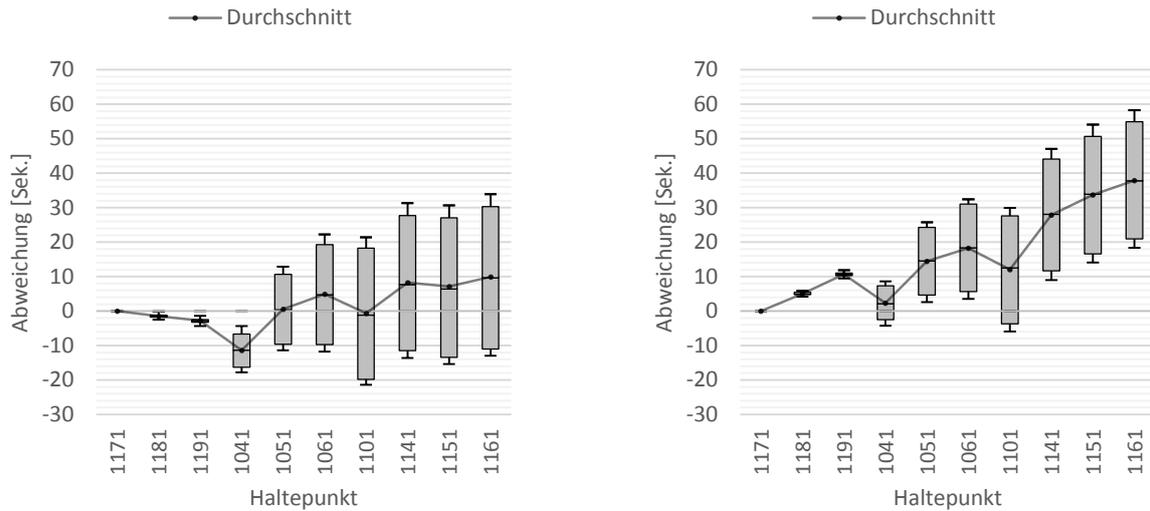


Abbildung 5.22: Abweichungsverlauf, Linie 4-B01. Links: UCL2015<sup>+</sup>, rechts: UCL2015<sup>-</sup>

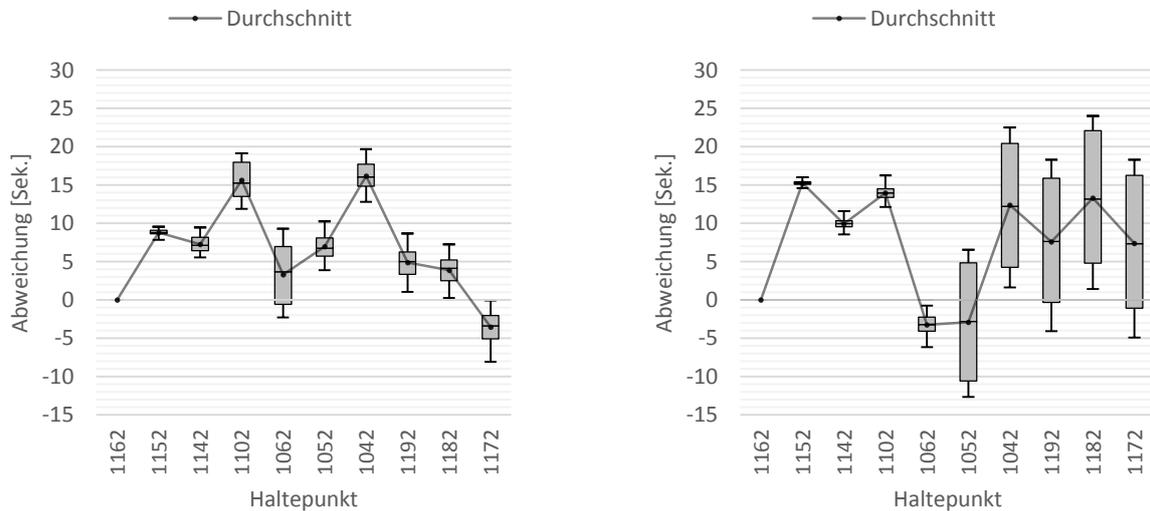


Abbildung 5.23: Abweichungsverlauf, Linie 4-B02. Links: UCL2015<sup>+</sup>, rechts: UCL2015<sup>-</sup>

Der Abweichungsverlauf von Linie 4-B02 zeigt keine neuen Phänomene (siehe Abbildung 5.23). Ähnlich wie Linie 4-B01 unterliegen ihre Fahrzeuge unter Fahrplan UCL2015<sup>+</sup> geringeren Haltezeiten am Starthaltepunkt 1162 als unter Fahrplan UCL2015<sup>-</sup>, da sie dort in einem Abstand von fünf anstatt neun Minuten zu Fahrzeugen der Linie 1-B02 abfahren. Der gleiche Effekt kann an Haltepunkt 1042 in Zusammenspiel mit Fahrzeugen der Linie 3-B02 beobachtet werden. Der unterschiedliche Abweichungsverlauf ab Haltepunkt 1062 erklärt sich durch die unregelmäßigen geplanten Sicherheitsabstände zwischen Fahrzeugen der Linien 2-B02 und 4-B02 unter Fahrplan UCL2015<sup>-</sup>.

Allgemein zeigt sich, dass die gemessenen Abweichungskennzahlen, wie erhofft, unabhängig vom Fahrplan weitestgehend moderat ausfallen. Die Abweichungsverläufe bestätigen dabei, wie auch bereits die Häufigkeitsverteilungen, dass Fahrplan UCL2015<sup>+</sup> im Vergleich zu Fahrplan UCL2015<sup>-</sup> unter den gewählten Simulationsparametern eine bessere Systemperformanz aufweist. Insbesondere Linien 1-B01, 2-B01, 3-B02, 4-B01 und 4-B02 scheinen unter Fahrplan UCL2015<sup>+</sup> bei Betrachtung der durchschnittlichen Abfahrtszeitabweichung sowie ihrer Streuung eine verbesserte Performanz aufzuweisen, während Linie 1-B02 von Fahrplan UCL2015<sup>-</sup> zu profitieren scheint. Linien 2-B02 und 3-B01 lassen hingegen keine eindeutige Aussage zu. Während letztere unter Fahrplan UCL2015<sup>+</sup> wesentlich geringere Durchschnittsabweichungen aufweist, ist die Streuung ihrer Abweichungen unter Fahrplan UCL2015<sup>-</sup> wesentlich geringer. Linie 2-B02 ist unter Fahrplan UCL2015<sup>+</sup> hingegen systematisch zu

früh, während sie unter Fahrplan UCL2015<sup>-</sup> zunächst zu früh und gegen Ende ihres Linienverlaufs erneut zu spät ist.

**Vergleich ausgewählter Haltepunkte.** Ein Großteil der im vorhergehenden Abschnitt identifizierten Phänomene entsteht in den Abschnitten des Netzwerks, in denen die Linien 2-B01 und 2-B02 verkehren. Beispielhaft sollen deswegen die Haltepunkte des Linienverlaufs von Linie 2-B02 sowie die beiden an ihn angrenzenden Haltepunkte 1071 und 1141 näher betrachtet werden. Tabelle 5.7 listet die an den Haltepunkten gemessene durchschnittliche Frühzeitigkeit und Verspätung auf. Die maximale durchschnittliche Frühzeitigkeit wird unter beiden Fahrplänen an Haltepunkt 1043 erreicht. Unter Fahrplan UCL2015<sup>+</sup> fällt sie 1,1 Sekunden (ca. 0,7%) geringer aus als unter Fahrplan UCL2015<sup>-</sup>. Die maximale durchschnittliche Verspätung wird hingegen unter beiden Fahrplänen an Haltepunkt 1071 erreicht. Unter Fahrplan UCL2015<sup>+</sup> kann sie dabei mit 31,4 Sekunden (ca. 51,6%) signifikant reduziert werden. Allgemein weist Fahrplan UCL2015<sup>+</sup>, wie bereits bei der durchschnittlichen Linienabweichung gesehen (vgl. Tabelle 5.6), vermehrt höhere durchschnittliche Abweichungen bei frühzeitigen Abfahrten auf, lediglich an Haltepunkten 1043, 1091, 1111 und 1121 kann sie im Vergleich zu Fahrplan UCL2015<sup>-</sup> leicht reduziert werden. Im Gegensatz dazu kann allerdings die durchschnittliche Verspätung unter Fahrplan UCL2015<sup>+</sup> an nahezu jedem Haltepunkt reduziert werden. Nur Haltepunkt 1111 weist unter ihm eine – allerdings nicht signifikant – höhere Verspätung auf. An Haltepunkt 1031 wird unter beiden Fahrplänen keine Abweichung bei frühzeitigen Abfahrten gemessen, da es sich als Starthaltepunkt von Linie 2-B02 um einen Kontrollpunkt handelt, an dem immer zum planmäßigen Abfahrtszeitpunkt abgefahren wird.

Haltepunkt	Frühzeitigkeit [Sek.]		Verspätung [Sek.]	
	UCL2015 <sup>+</sup>	UCL2015 <sup>-</sup>	UCL2015 <sup>+</sup>	UCL2015 <sup>-</sup>
<b>1031</b>	<b>0,0</b>	<b>0,0</b>	<b>4,5</b>	5,2
<b>1043</b>	<b>13,6</b>	14,7	<b>1,2</b>	5,7
<b>1052</b>	9,7	<b>5,2</b>	<b>5,9</b>	7,3
<b>1062</b>	10,1	<b>2,8</b>	<b>12,1</b>	23,7
<b>1071</b>	8,1	<b>0,0</b>	<b>29,5</b>	60,9
<b>1091</b>	<b>1,3</b>	1,7	<b>7,3</b>	6,5
<b>1102</b>	4,9	<b>0,6</b>	<b>16,4</b>	26,8
<b>1111</b>	<b>2,3</b>	3,4	9,1	<b>9,0</b>
<b>1121</b>	<b>3,2</b>	4,3	<b>10,8</b>	11,7
<b>1131</b>	8,9	<b>8,0</b>	<b>6,9</b>	8,0
<b>1141</b>	5,3	<b>4,5</b>	<b>11,1</b>	16,3
<b>1142</b>	8,2	<b>3,5</b>	<b>8,6</b>	14,6

*Tabelle 5.7: Durchschnittliche Frühzeitigkeit und Verspätung an ausgewählten Haltepunkten, UCL 2015*

Wie bereits die Abweichungskennzahlen für einzelne Linien, lassen auch die Abweichungskennzahlen für ausgesuchte Haltepunkte Fahrplan UCL2015<sup>+</sup> unter den gegebenen Bedingungen insgesamt als geeigneter erscheinen. Allerdings zeigt sich auch hier, dass eine Reduzierung der Verspätung häufig mit einer Erhöhung der Frühzeitigkeit einhergeht. In solchen Fällen ist eine eindeutige Empfehlung für einen der beiden Fahrpläne nur schwer möglich. Ein, insbesondere für Fahrpläne für Bussysteme, geeigneteres Qualitätskriterium wurde bereits in den vorhergehenden Abschnitten eingeführt: Bunching. Die mit diesem Phänomen beschriebenen ungewollten Kolonnenfahrten wirken sich negativ auf die durch Passagiere wahrgenommene Angebotsqualität aus, indem sie zu unregelmäßigen Haltezeiten und Fahrzeugauslastungen führen.

Die Auswirkungen von Bunching-Effekten konnten z.T. bereits bei der Untersuchung der Abweichungsverläufe der einzelnen Linien beobachtet werden. Die Streuung der Abfahrtszeitabweichung ist dabei allerdings nur ein indirekter Indikator für Bunching. Für eine direkte Messung werden im Folgenden die im Simulationsverlauf an den einzelnen Haltepunkten beobachteten Sicherheitsabstände zwischen aufeinander folgenden Abfahrten mit ihren im Fahrplan angegebenen Planwerten verglichen. Weisen die beobachteten Sicherheitsabstände dabei eine Reduktion gegenüber den geplanten Abständen

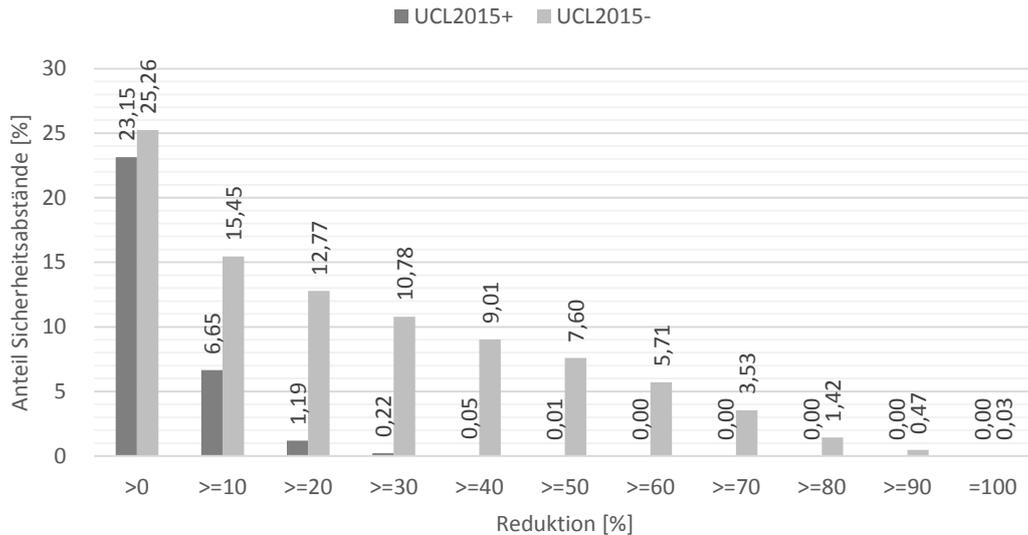


Abbildung 5.25: Durchschnittlicher, kumulierter Anteil reduzierter Sicherheitsabstände, UCL 2015

auf, so ist dies ein Indiz für Bunching. Um unterschiedliche Fahrpläne – mit unterschiedlich großen geplanten Sicherheitsabständen – miteinander vergleichen zu können wird die Differenz zwischen geplantem und beobachtetem Sicherheitsabstand mit dem geplanten Sicherheitsabstand ins Verhältnis gesetzt. Hierbei wird für den beobachteten Sicherheitsabstand der Absolutwert verwendet, da Änderungen der Fahrzeugabfolge – und eine daraus resultierende negative Differenz der beobachteten Abfahrtszeiten – keine Auswirkung auf das Auftreten ungewollter Kolonnenfahrten haben. Da Fahrpläne darüber hinaus auch eine unterschiedliche Zahl von Abfahrten innerhalb eines spezifischen Zeitfensters aufweisen können, wird zur Wahrung der Vergleichbarkeit die Zahl der beobachteten reduzierten Sicherheitsabstände mit der Gesamtzahl aller beobachteten Sicherheitsabstände ins Verhältnis gesetzt. Abbildung 5.25 zeigt den so berechneten durchschnittlichen, kumulierten Anteil reduzierter Sicherheitsabstände für Fahrplan UCL2015<sup>+</sup> und Fahrplan UCL2015<sup>-</sup>, aufgeteilt in elf Reduktionsklassen. Beide Fahrpläne zeigen mit 23,15 bzw. 25,26 Prozent einen ähnlich hohen Gesamtanteil reduzierter Sicherheitsabstände. Während unter Fahrplan UCL2015<sup>+</sup> der Großteil der reduzierten Sicherheitsabstände allerdings eine Reduktion von weniger als zehn Prozent aufweist, werden unter Fahrplan UCL2015<sup>-</sup> signifikant höhere Anteile größerer Reduktionen beobachtet. Dies lässt darauf schließen, dass die unter Fahrplan UCL2015<sup>-</sup> abrufbare Servicequalität aufgrund von Bunching-Effekten geringer ist als unter Fahrplan UCL2015<sup>+</sup>.

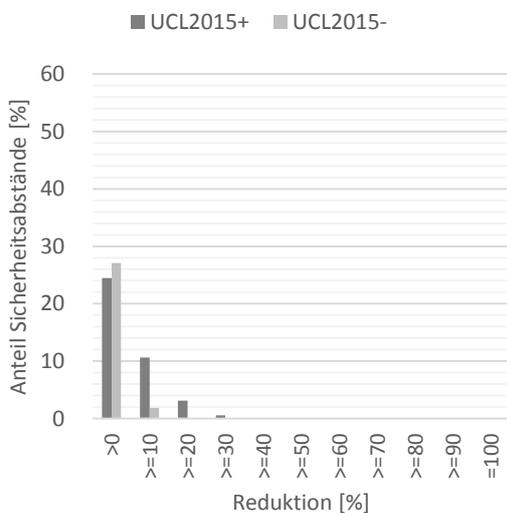


Abbildung 5.26: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1091, UCL 2015

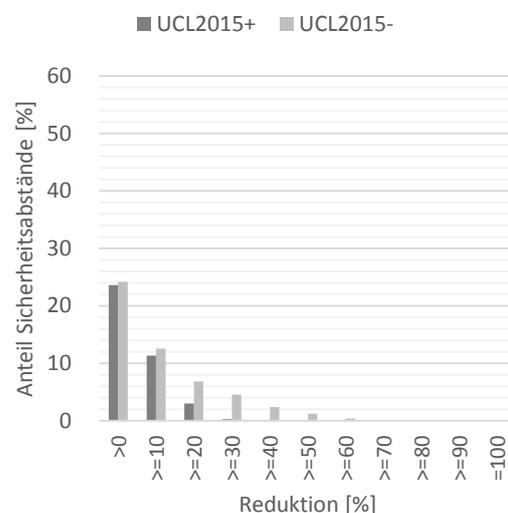


Abbildung 5.24: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1111, UCL 2015

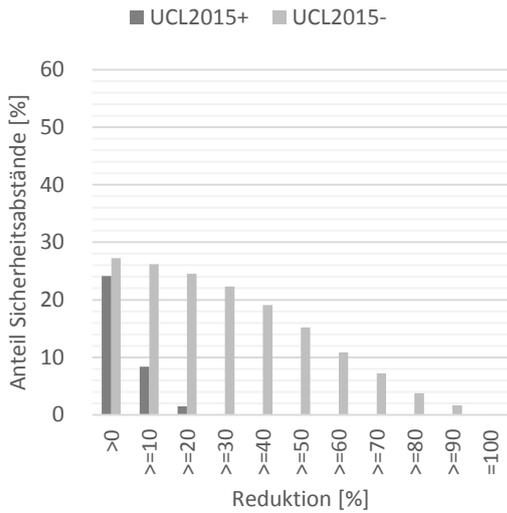


Abbildung 5.27: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1121, UCL 2015

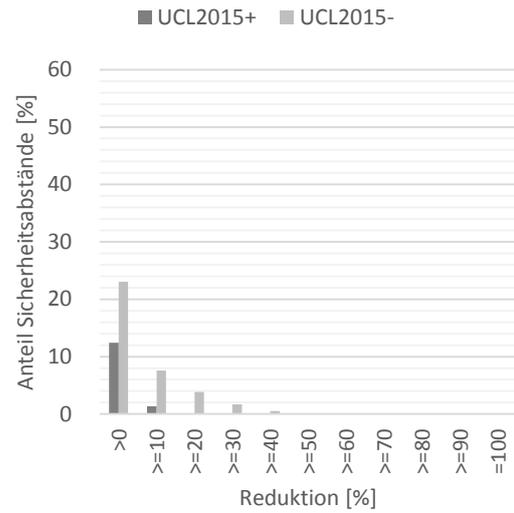


Abbildung 5.28: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1131, UCL 2015

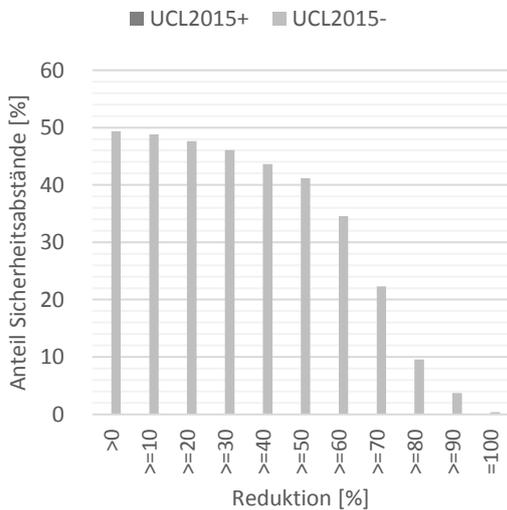


Abbildung 5.29: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1142, UCL 2015

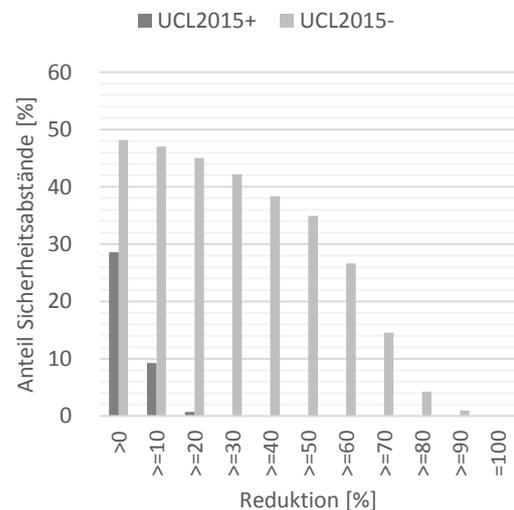


Abbildung 5.30: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1102, UCL 2015

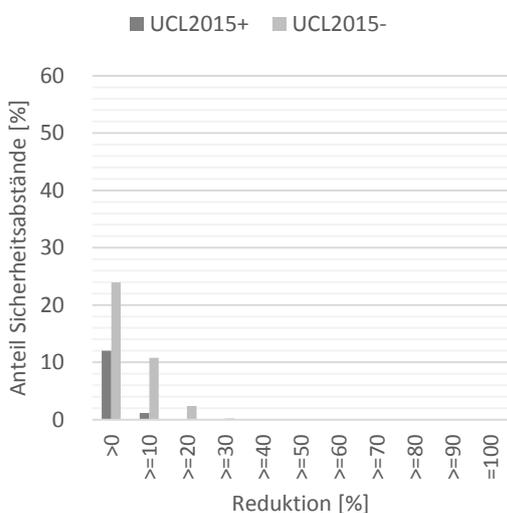


Abbildung 5.31: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1062, UCL 2015

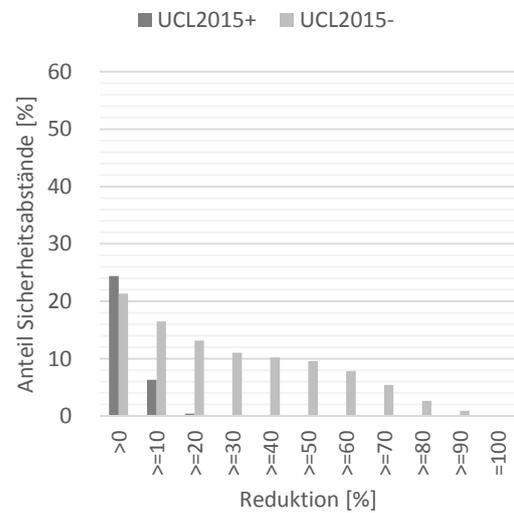


Abbildung 5.32: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1052, UCL 2015

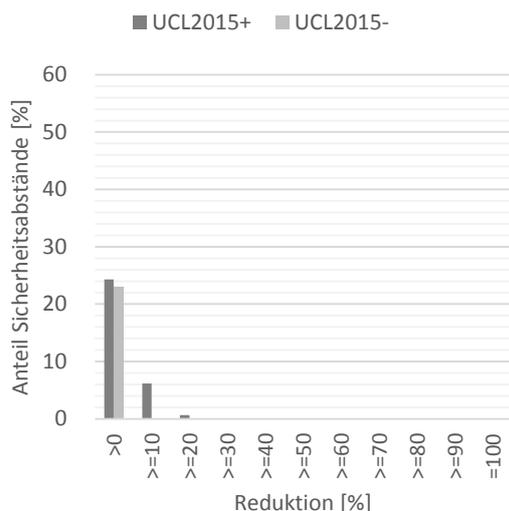


Abbildung 5.33: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1043, UCL 2015

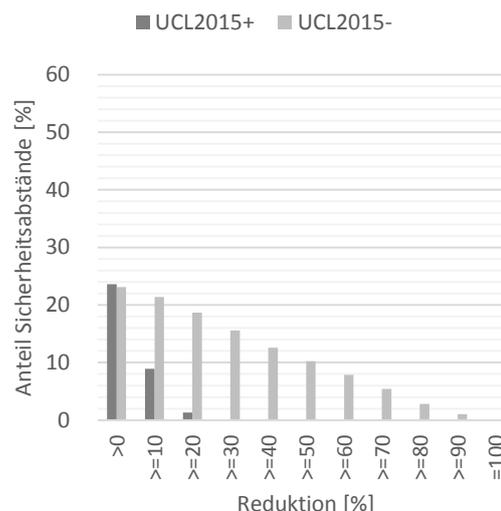


Abbildung 5.34: Kumulierter Anteil reduzierter Sicherheitsabstände, Haltepunkt 1031, UCL 2015

Um zu untersuchen wie sich Bunching-Effekte über den Verlauf einer Linie entwickeln, werden beispielhaft die von Linie 2-B02 bedienten Haltepunkte genauer betrachtet (siehe Abbildung 5.26 bis Abbildung 5.34), wobei aufgrund des zirkulären Linienverlaufs mit Haltepunkt 1091 begonnen und mit Haltepunkt 1031 geendet wird. Unter beiden Fahrplänen wird an Haltepunkt 1091 mit einem ähnlich hohen Gesamtanteil reduzierter Sicherheitsabstände begonnen. Wie aufgrund der Untersuchungen zum Abweichungsverlauf bereits vermutet, besitzt Fahrplan UCL2015<sup>+</sup> hier einen höheren Anteil leicht reduzierter Sicherheitsabstände. Dies ist dabei auf die erhöhte Streuung der Abfahrtszeitabweichungen von Linie 3-B01 unter Fahrplan UCL2015<sup>+</sup> zurückzuführen. Während der Anteil reduzierter Sicherheitsabstände unter Fahrplan UCL2015<sup>+</sup> aber über den weiteren Linienverlauf bis Haltepunkt 1131 konstant bleibt oder leicht sinkt, nimmt er unter Fahrplan UCL2015<sup>-</sup> deutlich zu. Insbesondere steigt dabei der Anteil stark reduzierter Sicherheitsabstände. Dies bestätigt die bereits bei den Abweichungsverläufen identifizierten Phänomene. An Haltepunkt 1131 sinkt der Anteil reduzierter Sicherheitsabstände kurzzeitig, nimmt bis Haltepunkt 1102 unter beiden Fahrplänen aber wieder zu, bevor er an Haltepunkt 1062 erneut sinkt. Es steht zu vermuten, dass auf diesem Abschnitt die bei den Untersuchungen der Abweichungsverläufe identifizierten kurzzeitigen Überholmanöver zwischen den Linien 2-B02 und 3-B01 stattfinden. Darüber hinaus stößt an Haltepunkt 1142 Linie 4-B02 zu Linien 2-B02 und 3-B01 hinzu. Zusammen mit der schlechteren Koordination zwischen den Linien zeigen sich hier unter Fahrplan UCL2015<sup>-</sup> deutliche Bunching-Effekte bei denen ca. jeder zweite Sicherheitsabstand reduziert wird. Unter Fahrplan UCL2015<sup>+</sup> sind solche Effekte hingegen nicht zu beobachten. Zwar nimmt auch hier der Anteil reduzierter Sicherheitsabstände an Haltepunkt 1102 (wieder) zu, aber in einem wesentlich geringeren Ausmaß. Gegen Ende des Linienverlaufs, zwischen den Haltepunkten 1052 und 1031, zeigen sich unter Fahrplan UCL2015<sup>-</sup> erneut Schwankungen im Anteil reduzierter Sicherheitsabstände, aufgrund der schlechteren Koordination zwischen den auf diesem Abschnitt wegfallenden bzw. neu hinzustoßenden Linien.

**Fazit.** Zusammenfassend lässt sich festhalten, dass Optimalfahrplan UCL2015<sup>+</sup> unter den gewählten Simulationsparametern insgesamt eine bessere Systemperformanz aufweist. Zwar fallen unter beiden verglichenen Fahrplänen die beobachteten Abfahrtszeitabweichungen ähnlich moderat aus, unter Fahrplan UCL2015<sup>+</sup> treten aber weniger verspätete Abfahrten und insbesondere weniger stark verspätete Abfahrten auf. Zu beachten ist, dass diese Reduktion nicht zwingend in einem Gewinn an Pünktlichkeit mündet, sondern aufgrund der geringen Zahl an Kontrollpunkten z.T. zu einer Steigerung der Frühzeitigkeit führt. Unabhängig davon zeigt unter Fahrplan UCL2015<sup>+</sup> aber ein Großteil der Linien des Netzes ein vorteilhafteres Verhalten mit weniger starken Streuungen und Extremwerten der Abfahrtszeitabweichungen sowie wesentlich schwächeren Bunching-Effekten. Da die wesentlichen Simulationsparameter

i.d.R. auf Basis theoretisch optimaler Werte bestimmt wurden (z.B. Ankunftsrate aufgrund von oberen Schranken von Sicherheitsabständen) und Fahrplan UCL2015<sup>+</sup> häufiger an diese Werte herankommt als Fahrplan UCL2015<sup>-</sup>, war dieses Resultat zu erwarten und zeigt, dass die Simulationsanwendung dazu herangezogen werden kann, die Güte eines Fahrplans für ein gegebenes Verkehrssystem zu untersuchen.

Allgemein zeigen die Untersuchungen, dass das Simulationsmodell zur Abbildung von Bussystemen geeignet ist: Die erhobenen Kennzahlen lassen auf plausibles Verhalten schließen und zeigen keine Anhaltspunkte für übermäßig unrealistische Verhaltensmuster. Insbesondere kann, wie erhofft, bereits mit den vergleichsweise simplen Fahr- und Haltezeitmodellen das auch in Realsystemen zu beobachtende Bunching und damit einhergehende dynamische Effekte abgebildet werden. Die Ergebnisse der Untersuchungen weisen aber auch auf potentielle Simulationsartefakte hin: Die einheitliche Bestimmung durchschnittlicher Fahrzeiten aufgrund geplanter Fahrzeiten für alle Verbindungen/Segmente ist nicht detailliert genug und resultiert in leichten systematischen Verzögerungen aufgrund von zu knapp geplanten Fahrzeiten bzw. zu lang geplanten Haltezeiten. Außerdem führt das Haltezeitmodell zu ungewollten Schwankungen in den Haltezeiten aufgrund von unterschiedlichen Takten auf gemeinsamen Streckenabschnitten. Alle diese Phänomene können dabei auf die sehr spärliche Datenbasis und die in Folge dessen notwendigerweise simplen Teilmodelle zurückgeführt werden. Erst mit detaillierteren empirischen Daten können die betroffenen Teilmodelle weiter ausgearbeitet und die Zahl der Simulationsartefakte reduziert werden.

### 5.3.2. Das Kölner Stadtbahnnetz aus dem Jahr 2001

Um zu überprüfen, ob das Simulationsmodell auch das Verhalten von Stadtbahnssystemen plausibel abbildet, wird es auf ein Modell des Kölner Stadtbahnnetzes aus dem Jahr 2001 angewendet. Aufgrund des signifikant größeren Umfangs im Vergleich zu Netzwerk UCL 2015 – es besteht aus ca. zehnmal mehr physischen Komponenten auf denen ca. viermal so viele Linien verkehren (siehe Tabelle 5.3) – ist eine vollumfängliche Besprechung aller Komponenten im Rahmen dieser Arbeit nicht möglich. Stattdessen werden beispielhaft einige Teilbereiche des simulierten Systems untersucht, die auffällige Unterschiede bei der Simulation der beiden Fahrpläne zeigen.

Die haltepunktabhängigen Ankunftsrate werden auch hier mangels empirischer Daten auf Basis der oberen Schranken für die Sicherheitsabstände gemäß Gleichung (5.9) bestimmt. Dabei werden für die planmäßigen Haltezeiten  $T_{f,s}$  erneut 20 Sekunden festgelegt, um moderate Verspätungseffekte zu erzielen. Im Unterschied zu Netz UCL 2015 werden alle Haltepunkte, wie bei Stadtbahnssystemen üblich, als Kontrollpunkte deklariert. Da Haltepunkte in Stadtbahnssystemen darüber hinaus in der Regel nur einem Fahrzeug Platz bieten, wird die maximale Kapazität aller Haltepunkt-Knoten auf ein Fahrzeug limitiert. So werden außerdem unzulässige Überholvorgänge an Haltepunkten vermieden.

Zur Bestimmung der Fahrzeitparameter  $\gamma$  und  $\eta$  werden ebenfalls wieder die vorhandenen Daten zu geplanten Fahrzeiten herangezogen. Da Bahnen im Simulationsmodell allerdings auf Segmenten und nicht auf Verbindungen verkehren, werden entsprechend die Fahrzeitdaten der Segmente verwendet. Die durchschnittliche Segment-Fahrzeit beträgt in Netz KVB-T 2001 ca. 80 Sekunden. Wird erneut die planmäßige Haltezeit von 20 Sekunden herausgerechnet, ergibt sich so  $\gamma = 1 - \left(\frac{20}{80}\right) = 0,75$ . Da Stadtbahnen in der Kölner Innenstadt in der Regel unterirdisch und in den äußeren Bezirken der Stadt häufig auf dedizierter oberirdischer Infrastruktur verkehren, sind sie weniger starken Schwankungen der Fahrzeit ausgesetzt und es wird  $\eta = 0,1$  gesetzt. Durch die Befahrung einzelner Segmente wird darüber hinaus der Parameter  $\iota$  nicht benötigt, es wird  $\iota = 0$  gesetzt.

Die mit der Optimierungsanwendung erzeugten Fahrpläne besitzen einen Einheitstakt von zehn Minuten. Der berechnete Optimalfahrplan KVBT2001<sup>+</sup> besitzt einen Zielfunktionswert von 11.712 und einen netzwerkweiten Sicherheitsabstand von zwei Minuten. Der schlechteste noch gültige Fahrplan KVBT2001<sup>-</sup> besitzt hingegen nur einen Zielfunktionswert von 7.257 und einen netzwerkweiten Sicherheitsabstand von einer Minute. Tabelle 5.8 zeigt die den einzelnen Linien zugewiesenen Startzeiten  $\lambda_i$ .

Fahrplan	Variante	Liniengruppe														
		1	3	4	5	6	7	8	9	12	13	15	16	17	18	19
KVBT2001 <sup>+</sup>	T01	3	9	5	7	7	0	7	9	9	9	1	5	9	7	1
	T02	2	5	2	4	9	9	9	9	4	8	9	2	2	3	8
KVBT2001 <sup>-</sup>	T01	9	9	8	9	9	9	9	9	9	9	9	9	6	8	6
	T02	8	9	9	8	9	9	9	9	9	8	9	8	9	7	

Tabelle 5.8: Simulierte Fahrpläne für KVB-T 2001

**Vergleich allgemeiner Merkmale.** Die durchschnittliche Verspätung pro verspäteter Abfahrt beträgt unter Fahrplan KVBT2001<sup>-</sup> 9,8 Sekunden. Sie kann unter Fahrplan KVBT2001<sup>+</sup> um 50 Prozent auf 4,9 Sekunden reduziert werden. Frühzeitige Abfahrten werden, entsprechend der gewählten Verkehrsmanagementstrategie, nicht verzeichnet.

Die in Abbildung 5.36 abgebildete Häufigkeitsverteilung der Abfahrtszeitabweichungen zeigt ein für Stadtbahnssysteme zu erwartendes Bild (vgl. dazu auch [94]): Fahrplan KVBT2001<sup>+</sup> – mit der durch die Optimierungsanwendung ausgewiesenen optimalen Serviceregularität – zeigt im Vergleich zu Fahrplan KVBT2001<sup>-</sup> eine höhere Zahl sehr leicht verspäteter Abfahrten und eine geringere Zahl von Abfahrten mit erhöhten Verspätungen von über 60 Sekunden. Von 50.520 beobachteten Abfahrten sind unter Fahrplan KVBT2001<sup>+</sup> insgesamt 24.152 (ca. 47,8%) verspätet, während unter Fahrplan KVBT2001<sup>-</sup> 25.095 (ca. 49,7%) eine Verspätung aufweisen. Aufgrund der gleichmäßigeren Verteilung der Sicherheitsabstände unter Fahrplan KVBT2001<sup>+</sup> kommt es seltener vor, dass sich Verspätungen auf Nachfolgerfahrzeuge übertragen. Bereits leicht verspätete Fahrzeuge werden dementsprechend seltener zusätzlich behindert, wodurch einmal vorhandene leichte Verspätungen seltener über die Minutenschwelle vergrößert werden. Wie Ullrich bereits in [94] (S. 138) festgestellt hat, wirkt sich eine erhöhte Serviceregularität erst bei Verspätungen ab 60 Sekunden signifikant aus, da durch das Optimierungsmodell für beide Fahrpläne ein minimaler Sicherheitsabstand von einer Minute garantiert wird. Abbildung 5.35 bestätigt dies: Während unter Fahrplan KVBT2001<sup>-</sup> 3.142 (ca. 6,2%) aller Abfahrten eine Verspätung von über einer Minute aufweisen, machen die 42 Abfahrten (ca. 0,1%) mit erhöhter Verspätung unter Fahrplan KVBT2001<sup>+</sup> einen zu vernachlässigenden Anteil aus.

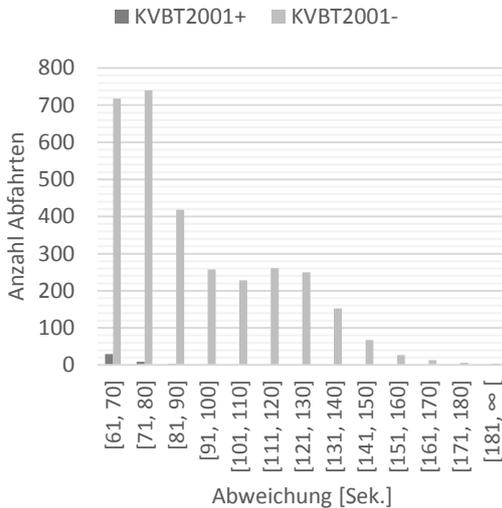


Abbildung 5.35: Häufigkeitsverteilung der größeren Abfahrtszeitabweichungen, KVB-T 2001

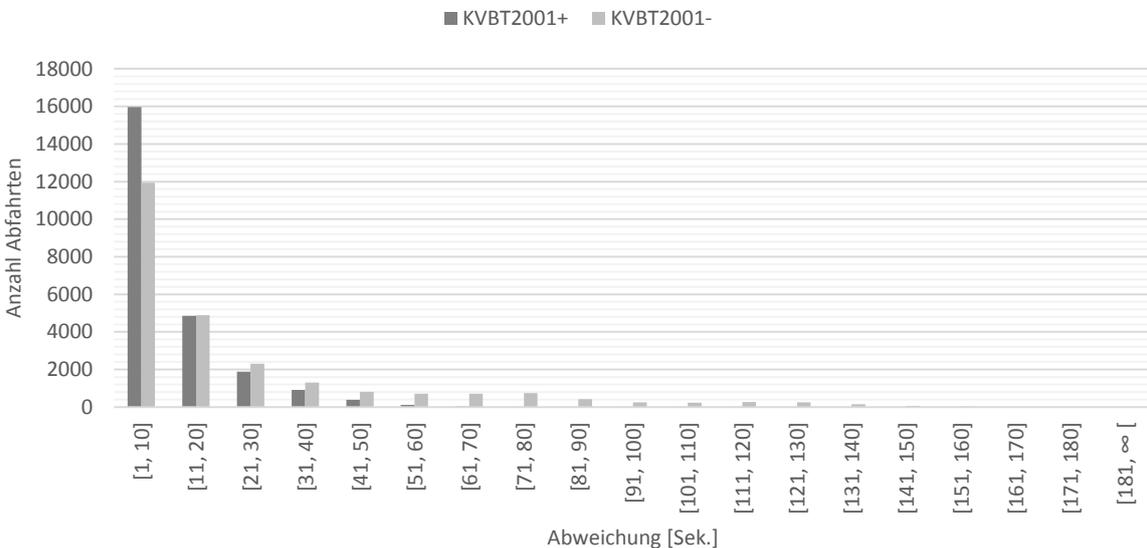


Abbildung 5.36: Häufigkeitsverteilung der Abfahrtszeitabweichungen, KVB-T 2001

Da in Bahnsystemen, im Gegensatz zu Bussystemen, keine frühzeitigen Abfahrten stattfinden, schlägt sich die geringere Zahl verspäteter Abfahrten unter Fahrplan KVBT2001<sup>+</sup> direkt in einer Erhöhung der Zahl pünktlicher Abfahrten und damit in einer verbesserten Systemperformanz und einer erhöhten wahrgenommenen Angebotsqualität nieder. Unter Fahrplan KVBT2001<sup>+</sup> finden entsprechend 26.368 (ca. 52,2%) aller Abfahrten pünktlich statt, während unter Fahrplan KVBT2001<sup>-</sup> 25.425 (ca. 50,6%) aller Abfahrten pünktlich sind.

Auffällig ist, dass sich unter Fahrplan KVBT2001<sup>-</sup> ein Wellenmuster mit lokalen Maxima bei der Häufigkeitsverteilung zeigt. Diese lokalen Maxima liegen dabei bei den Abweichungsklassen [71, 80] sowie [111, 120] und deuten auf systematische Probleme bei der Linienkoordination unter Fahrplan KVBT2001<sup>-</sup> hin.

Der Vergleich der allgemeinen Merkmale zeigt ansonsten keine Indikatoren dafür, dass die gewählten Modellparameter zu unrealistischem Simulationsverhalten führen. Darüber hinaus zeigt sich trotz scheinbar eher moderater Verspätungseffekte, wie erhofft, ein messbarer Unterschied zwischen der Güte der simulierten Fahrpläne, der Fahrplan KVBT2001<sup>+</sup> als geeigneter erscheinen lässt.

**Vergleich von Linien.** Tabelle 5.9 gibt einen Überblick über die durchschnittliche Verspätung der Stadtbahnlinien unter den beiden Fahrplänen. Unter Fahrplan KVBT2001<sup>+</sup> kann die durchschnittliche Linienverspätung im Vergleich zu Fahrplan KVBT2001<sup>-</sup> um ca. 60,1 Prozent von 13,3 Sekunden auf 5,2 Sekunden reduziert werden. Darüber hinaus weisen unter Fahrplan KVBT2001<sup>+</sup> nur Linien 3-T02, 6-T02, 15-T02 und 17-T01 eine durchschnittliche Verspätung von mehr als zehn Sekunden auf. Auch unter Fahrplan KVBT2001<sup>-</sup> bleibt die durchschnittliche Verspätung bei ca. zwei Dritteln der 30 Linien unter diesem Wert. Die drei unter Fahrplan KVBT2001<sup>+</sup> am stärksten verspäteten Linien belegen auch unter Fahrplan KVBT2001<sup>-</sup> diese Positionen, weisen dabei aber deutlich höhere durchschnittliche Verspätungen von (nahezu) über einer Minute auf.

Wie bereits bei den Untersuchungen zum Bussystem, profitieren auch hier nicht alle Linien vom Optimalfahrplan. So weisen Linien 1-T01, 3-T02, 7-T02, 9-T02, 12-T01, 16-T01 und 16-T02 unter ihm eine höhere Durchschnittsverspätung auf. Dabei kann aber lediglich die Verbesserung der Pünktlichkeit von Linie 3-T02 um 78,2 Prozent von 10,7 Sekunden auf 2,3 Sekunden unter Fahrplan KVBT2001<sup>-</sup> als signifikant bezeichnet werden.

Eine weitere interessante Beobachtung ist, dass sich von den Linien mit der größten durchschnittlichen Verspätung unter Fahrplan KVBT2001<sup>-</sup> Linien 6-T01 und 17-T01 sowie 6-T02, 15-T02 und 19-T02 weite Teile ihrer jeweiligen Streckenabschnitte miteinander teilen (vgl. Abbildung 3.3 bis Abbildung 3.6). Dies lässt auf eine mangelhafte Koordination dieser Linienbündel unter Fahrplan KVBT2001<sup>-</sup> schließen, weswegen sie im Folgenden genauer betrachtet werden.

Zunächst werden dazu wieder die Häufigkeitsverteilungen der Abfahrtszeitabweichungen herangezogen (siehe Abbildung 5.37 bis Abbildung 5.41). Für Linie 6-T01 zeigt sich dabei das bereits beim Vergleich der allgemeinen Merkmale beobachtete Verteilungsmuster: Unter Fahrplan KVBT2001<sup>+</sup> besitzt sie keine Abfahrten mit einer Verspätung von über 40 Sekunden, allerdings eine stark erhöhte Zahl leichter Verspätungen. Unter Fahrplan KVBT2001<sup>-</sup> zeigt sich hingegen eine größere Zahl erhöhter Verspätungen von über einer Minute und eine starke Verringerung der Zahl leichter Verspätungen. Konkret

Linie	Fahrplan	
	KVBT2001 <sup>+</sup>	KVBT2001 <sup>-</sup>
1-T01	7,2	5,6
1-T02	5,3	8,3
3-T01	2,5	5,1
3-T02	10,7	2,3
4-T01	6,7	25,9
4-T02	2,6	8,4
5-T01	2,6	9,1
5-T02	2,7	3,6
6-T01	3,9	36,8
6-T02	15,5	60,7
7-T01	1,6	1,7
7-T02	1,3	1,0
8-T01	1,2	1,7
8-T02	1,2	3,0
9-T01	3,3	6,5
9-T02	3,2	2,5
12-T01	4,7	2,6
12-T02	2,0	3,5
13-T01	4,6	5,6
13-T02	3,6	4,4
15-T01	8,7	10,3
15-T02	20,5	52,9
16-T01	5,1	4,9
16-T02	3,3	3,0
17-T01	17,3	71,8
17-T02	1,6	5,8
18-T01	2,5	3,8
18-T02	4,5	9,5
19-T01	1,8	11,5
19-T02	3,4	26,9
∅	5,2	13,3

Tabelle 5.9: Durchschnittliche Verspätung in Sekunden, pro Linie, KVB-T 2001

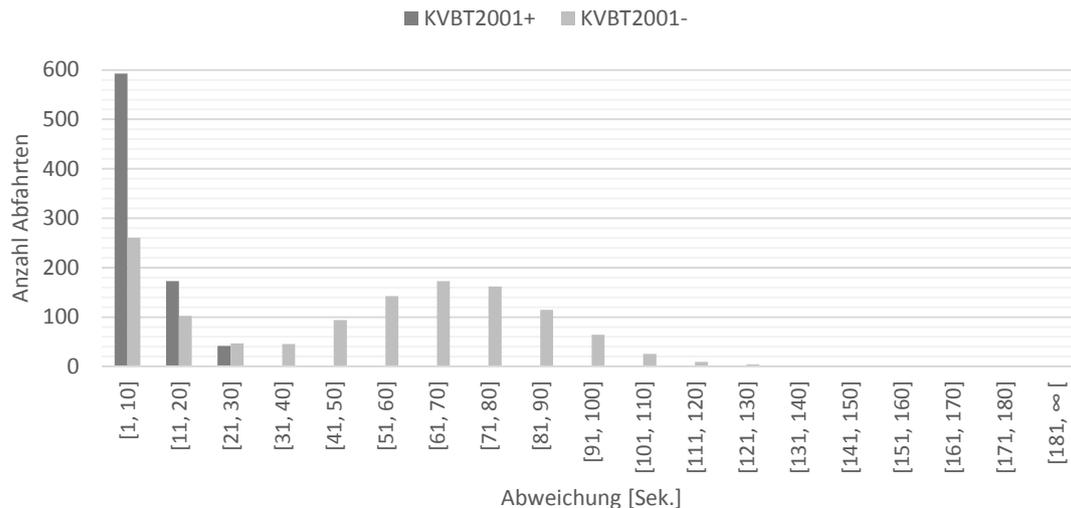


Abbildung 5.37: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 6-T01, KVB-T 2001

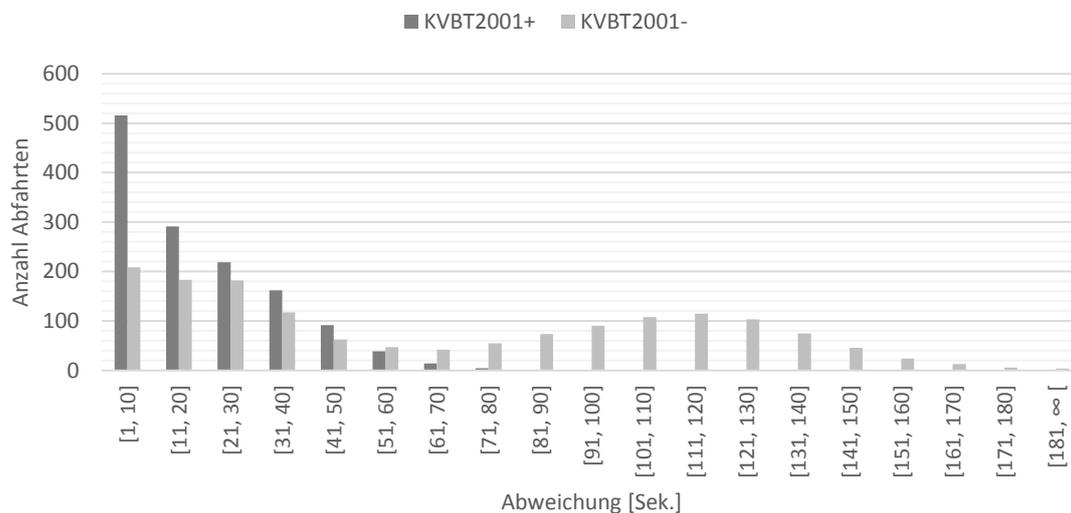


Abbildung 5.38: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 6-T02, KVB-T 2001

sind unter Fahrplan KVBT2001<sup>-</sup> von 1.680 Abfahrten von Linie 6-T01 insgesamt 1.246 (ca. 74,2%) verspätet und 381 (ca. 22,7%) weisen eine Verspätung von mehr als einer Minute auf. Unter Fahrplan KVBT2001<sup>+</sup> sind hingegen nur 809 (ca. 48,2%) aller Abfahrten verspätet und keine Abfahrt weist eine Verspätung von mehr als einer Minute auf. Unter Fahrplan KVBT2001<sup>-</sup> zeigt sich darüber hinaus ein lokales Maximum bei der Abweichungsklasse [61, 70], das auf ein systematisches Koordinierungsproblem schließen lässt.

Die in der entgegengesetzten Richtung verkehrende Linie 6-T02 weist unter beiden Fahrplänen eine der höchsten durchschnittlichen Verspätungen auf, was sich auch bei Betrachtung der Häufigkeitsverteilung der Abfahrtszeitabweichungen zeigt (siehe Abbildung 5.38). Unter Fahrplan KVBT2001<sup>+</sup> sind insgesamt 1.338 (ca. 79,6%) von 1.680 Abfahrten verspätet; unter Fahrplan KVBT2001<sup>-</sup> sind es hingegen 1.553 (ca. 92,4%). Während sich unter Fahrplan KVBT2001<sup>+</sup> die beobachteten Verspätungen allerdings im Bereich der leichten Verspätungen konzentrieren und nur sechs Abfahrten (ca. 0,4%) eine Verspätung von mehr als einer Minute aufweisen, zeigt sich unter Fahrplan KVBT2001<sup>-</sup> mit 711 Abfahrten (ca. 42,3%) ein hoher Anteil größerer Verspätungen. Auffällig ist auch hier die Existenz eines lokalen Maximums unter Fahrplan KVBT2001<sup>-</sup>. Anders als bei Linie 6-T01 liegt es aber bei Abweichungsklasse [111, 120].

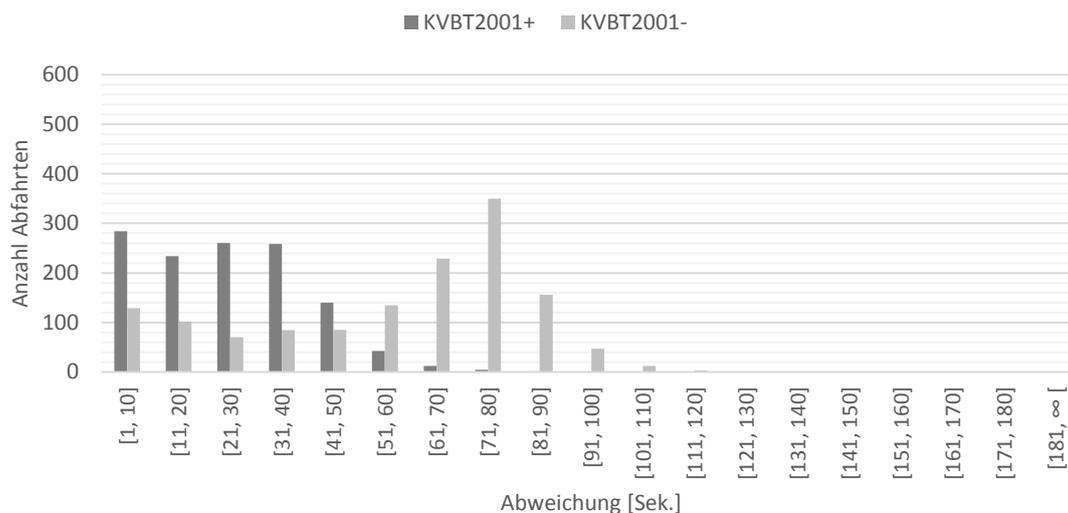


Abbildung 5.39: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 15-T02, KVB-T 2001

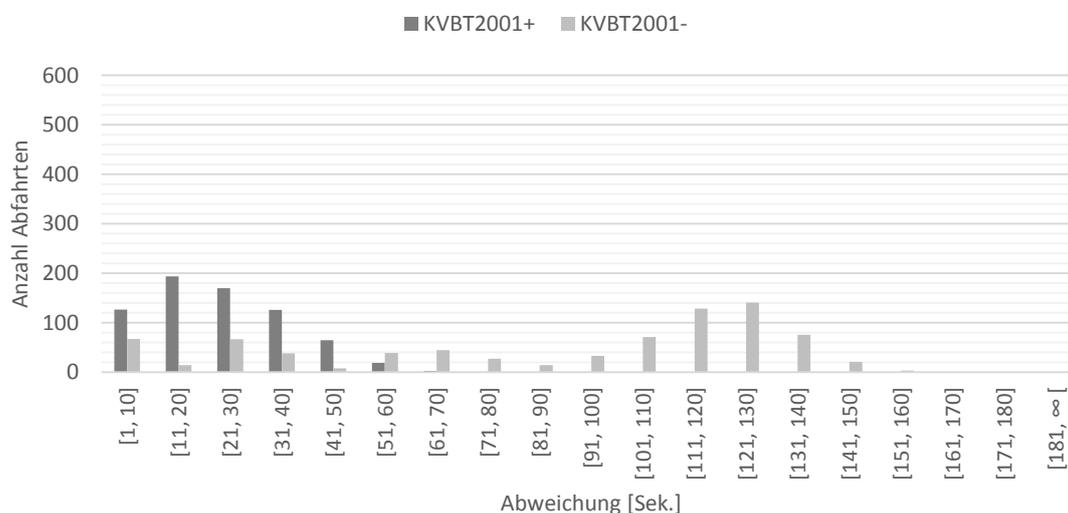


Abbildung 5.40: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 17-T01, KVB-T 2001

Die Häufigkeitsverteilung der Abfahrtszeitabweichung von Linie 15-T02 zeigt unter beiden Fahrplänen ein von der systemweit gemessenen Verteilung abweichendes Muster (siehe Abbildung 5.39). Unter Fahrplan KVBT2001<sup>+</sup> kann zwar auch hier beobachtet werden, dass der Anteil leichter Verspätungen erhöht und der Anteil erhöhter Verspätungen mit gerade einmal 19 von 1.500 Abfahrten (ca. 1,2%) sehr gering ist, allerdings findet sich ein lokales Maximum bei Abweichungsklasse [21, 30]. Es liegt also die Vermutung nahe, dass auch unter Fahrplan KVBT2001<sup>+</sup> ein – wenn auch kleines – systematisches Koordinierungsproblem besteht. Unter Fahrplan KVBT2001<sup>-</sup> zeigt sich ein ähnliches, allerdings wesentlich stärker ausgeprägtes, Verteilungsmuster wie bei Linien 6-T01/T02 mit einem globalen Maximum bei Abweichungsklasse [71, 80]. Wie zu erwarten, ist der Anteil erhöhter Verspätungen unter Fahrplan KVBT2001<sup>-</sup> mit 798 von 1.500 (ca. 53,2%) wesentlich höher als unter Fahrplan KVBT2001<sup>+</sup>.

Linie 17-T01 zeigt ähnliche Verteilungsmuster wie Linie 15-T02 (siehe Abbildung 5.40), wobei die jeweiligen Verteilungsspitzen allerdings bei den Abweichungsklassen [11, 20] resp. [121, 130] liegen. Trotz der unter Fahrplan KVBT2001<sup>-</sup> höheren durchschnittlichen Linienverspätung weist Linie 17-T01 dabei keinen höheren Gesamtanteil verspäteter Abfahrten als Linie 6-T02 auf. Mit 789 von 960 Abfahrten (ca. 82,2%) liegt er unter dem von Linie 6-T02. Allerdings unterliegt Linie 17-T01 mit 558 Abfahrten (ca. 58,1%) unter Fahrplan KVBT2001<sup>-</sup> einem noch höheren Anteil erhöhter Verspätungen als Linie 6-T02. Unter Fahrplan KVBT2001<sup>+</sup> sind hingegen nur drei Abfahrten (ca. 0,3%) mehr als einer Minute verspätet.

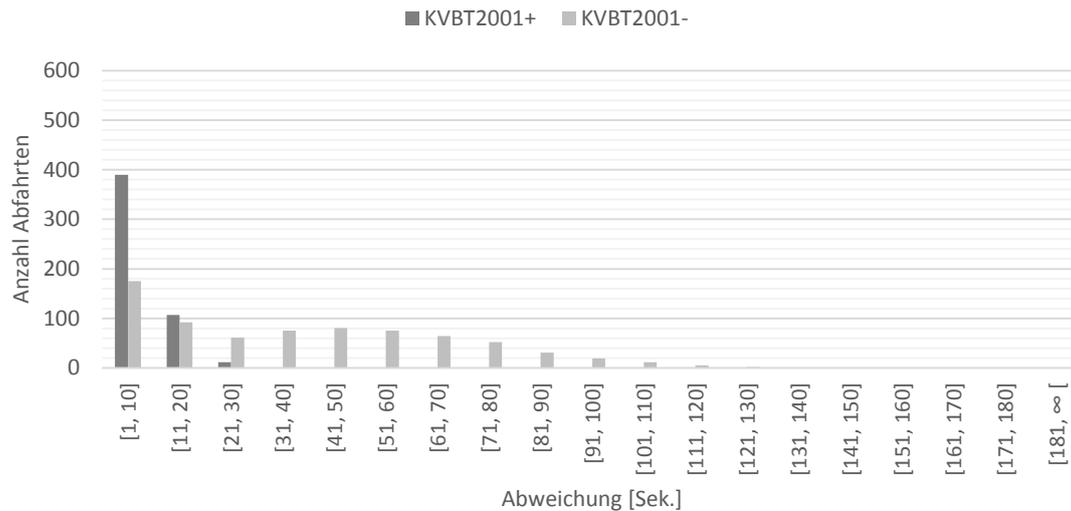


Abbildung 5.41: Häufigkeitsverteilung der Abfahrtszeitabweichungen, Linie 19-T02, KVB-T 2001

Die Häufigkeitsverteilung der Abfahrtszeitabweichungen von Linie 19-T02 zeigen keine neuen Phänomene. Mit 746 von 1.080 Abfahrten (ca. 69,1%) sind auch hier unter Fahrplan KVBT2001<sup>-</sup> insgesamt mehr Abfahrten verspätet als unter Fahrplan KVBT2001<sup>+</sup>, bei dem die Zahl der verspäteten Abfahrten bei 509 (ca. 47,1%) liegt. Darüber hinaus ist unter Fahrplan KVBT2001<sup>+</sup> keine Abfahrt mehr als 60 Sekunden verspätet, während unter Fahrplan KVBT2001<sup>-</sup> noch 187 (ca. 17,3%) eine Abweichung von mehr als einer Minute aufweisen. Unter Fahrplan KVBT2001<sup>-</sup> zeigt sich auch hier ein lokales Maximum, das bei Abweichungsklasse [41, 50] liegt und auf Koordinierungsprobleme hindeutet.

Die Häufigkeitsverteilungen bestärken prinzipiell die Vermutung, dass das Simulationsverhalten plausibel ist. Um allerdings auszuschließen, dass es sich bei den identifizierten Phänomenen nicht im Simulationsartefakte, sondern tatsächlich um fahrplanbedingte Koordinierungsprobleme zwischen verschiedenen Linien handelt und um die konkreten Stellen im Streckennetz zu identifizieren an denen diese Phänomene ihren Ursprung haben, werden erneut die Abweichungsverläufe der entsprechenden Linien betrachtet. Sie werden in Abbildung 5.42 bis Abbildung 5.51 dargestellt.

Fahrzeuge von Linie 6-T01 fahren von der Haltestelle *Marienburg, Südpark (MBG)* zur Haltestelle *Longerich, Friedhof (LGR)* und zeigen unter beiden Fahrplänen anfänglich nahezu identische Muster im Abweichungsverlauf (siehe Abbildung 5.42 und Abbildung 5.43). Der Anstieg in der Verspätung zwischen den Haltestellen MBG und *Goltsteinstr./Gürtel (GTG)* bzw. *Tacitusstr. (TAC)* und *Bonntor (BNT)* ergibt sich dabei, wie auch bereits beim Bussystem beobachtet, aufgrund der mit 60 Sekunden etwas zu knapp geplanten Fahrzeit. Ab der Haltestelle *Rudolfplatz (RDP)* ändert sich der Verlauf der Abfahrtszeitabweichungen unter Fahrplan KVBT2001<sup>-</sup> deutlich. Fahrzeuge von Linie 6-T01 sammeln unter ihm ab dort bis zur Haltestelle *Hansaring (HSR)* eine durchschnittliche Verspätung von 83,6 Sekunden an, während sie unter Fahrplan KVBT2001<sup>+</sup> keinen signifikanten Störungen ausgesetzt sind. Der Grund für den hohen Anstieg in Verspätung sind zu geringe Sicherheitsabstände zwischen Fahrzeugen von Linie 6-T01 und vorausfahrenden Fahrzeugen von Linie 17-T01. Letztere fahren ab der Haltestelle *Eifelstr. (EFS)* unter Fahrplan KVBT2001<sup>-</sup> sukzessive größere Verspätungen ein, deren Maximum sie mit 129,4 Sekunden an Haltestelle HSR erreichen (vgl. Abbildung 5.49). Da Nachfolgerfahrzeuge von Linie 6-T01 unter Fahrplan KVBT2001<sup>-</sup> planmäßig nur in einem Abstand von einer Minute zu Fahrzeugen von Linie 17-T01 verkehren, holen sie trotz der Festlegung von Kontrollpunkten, auf dem gemeinsamen Streckenabschnitt zwischen Haltestellen *Chlodwigplatz (CWP)* und HSR auf ihre Vorgängerfahrzeuge auf. Da in Stadtbahnsystemen, anders als in Bussystemen, aber keine Überholvorgänge möglich sind, überträgt sich ein Teil der Verspätung der Fahrzeuge von Linie 17-T01 in der Folge auf Fahrzeuge der Linie 6-T01. Erst nachdem sich ihre Linienverläufe nach Haltestelle HSR wieder trennen sind Fahrzeuge von Linie 6-T01 in der Lage einen Teil der eingefahrenen Verspätung aufzuholen. Unter Fahrplan KVBT2001<sup>+</sup> sind die Sicherheitsabstände auf dem – zum Teil ebenfalls durch die Linien 12-

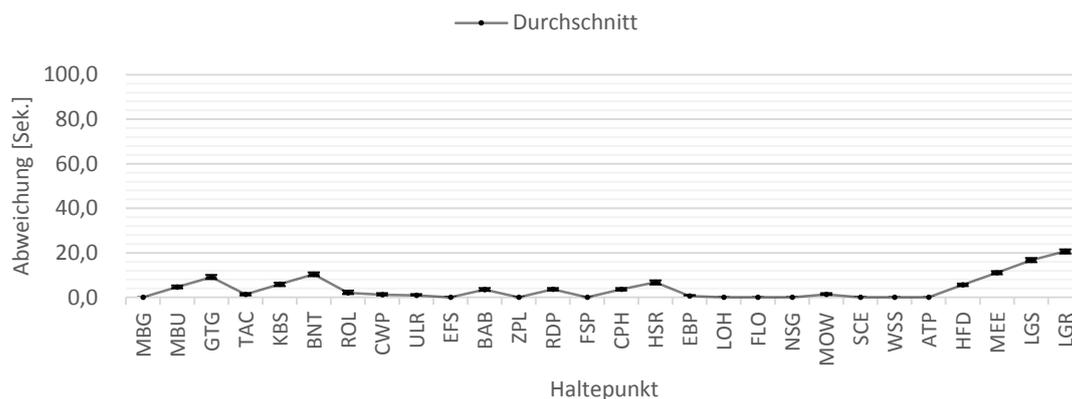


Abbildung 5.42: Abweichungsverlauf, Linie 6-T01, Fahrplan KVBT2001+

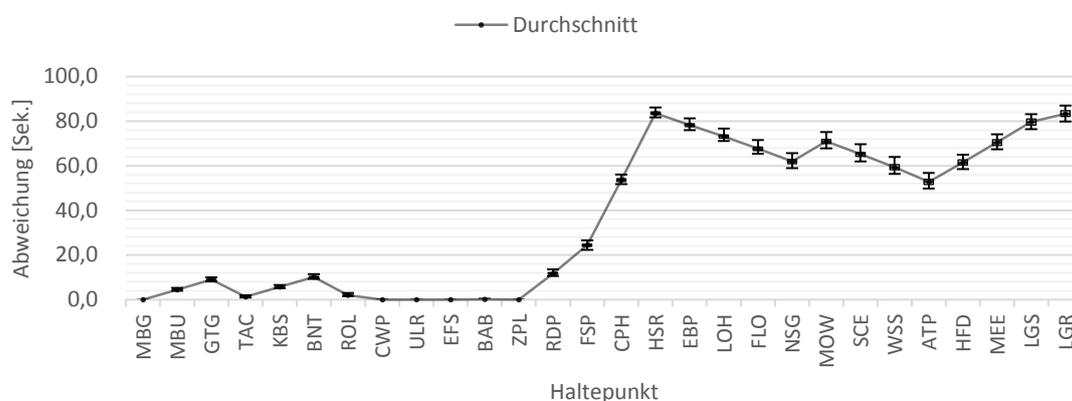


Abbildung 5.43: Abweichungsverlauf, Linie 6-T01, Fahrplan KVBT2001-

T01, 15-T01 und 19-T01 befahrenen – gemeinsamen Streckenabschnitten gleichmäßiger verteilt. Fahrzeuge von Linie 17-T01 fahren unter ihm z.B. vier Minuten nach Fahrzeugen der Linie 6-T01, während diese wiederum vier Minuten nach Fahrzeugen der Linie 15-T01 abfahren. Entsprechend müsste sich ein vorrausfahrende Fahrzeug unter Fahrplan KVBT2001+ zunächst um mindestens vier Minuten verspäten, bevor sich diese Verspätung auf ein Nachfolgerfahrzeug übertragen kann. Fahrplanunabhängig verlaufen die Abfahrtszeitabweichungen von Fahrzeugen der Linie 6-T01 ab der Haltestelle *Altonaer Platz (ATP)* wieder identisch. Da die unter Fahrplan KVBT2001- vorher eingefahrene Verspätung aber nur auf durchschnittlich ca. 52,8 Sekunden reduziert werden kann, ist der Verlauf unter ihm entsprechend verschoben.

Linie 6-T02 bedient die Gegenrichtung zu Linie 6-T01, ihre Fahrzeuge verkehren also zwischen den Haltestellen Longerich, Friedhof und Marienburg, Südpark (siehe Abbildung 5.44 und Abbildung 5.45). Auch der Verlauf ihrer Abfahrtszeitabweichungen zeigt zunächst aufgrund der kurzen geplanten Fahrzeiten zwischen den Haltestellen LGR und ATP ein ähnliches Muster, allerdings fällt die Steigung unter Fahrplan KVBT2001- stärker aus, da Fahrzeuge von Linie 6-T02 unter ihm ab Haltestelle *Longericher Str. (LGS)* acht Minuten nach Fahrzeugen der dort hinstoßenden Linie 18-T02 abfahren und folglich – wie ebenfalls bereits bei den Untersuchungen des Bussystems beobachtet – systematisch erhöhte Haltezeiten zugewiesen bekommen. Unter Fahrplan KVBT2001+ beträgt dieser Sicherheitsabstand hingegen vier Minuten, sodass sich unter ihm kürzere Haltezeiten für Fahrzeuge von Linie 6-T02 ergeben.

Wie bei Linie 6-T01 ändert sich der Verlauf der Abfahrtszeitabweichungen unter Fahrplan KVBT2001- signifikant sobald sich Linien 12-T02 und 18-T02 von Linie 6-T02 trennen und Linien 15-T02, 17-T02, sowie 19-T02 zu ihr hinzustoßen; dies geschieht an der Haltestelle *Ebertplatz (EBP)*. Unter Fahrplan KVBT2001- finden sich auch hier sehr unregelmäßige Sicherheitsabstände: Beginnend mit Fahrzeugen von Linie 6-T02 fahren Fahrzeuge von Linie 19-T02, 17-T02 und 15-T02 jeweils in einem Abstand von einer Minute nacheinander ab, während Fahrzeuge von Linie 6-T02 wiederum planmäßig

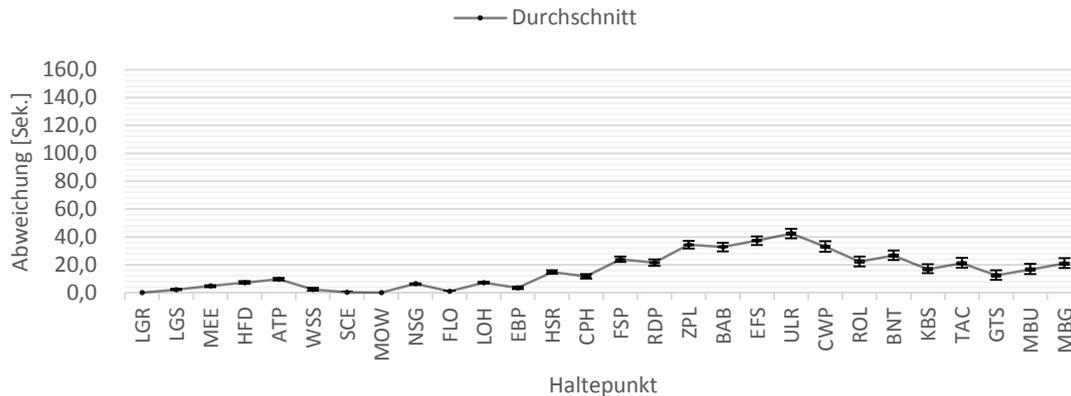


Abbildung 5.44: Abweichungsverlauf, Linie 6-T02, Fahrplan KVBT2001+

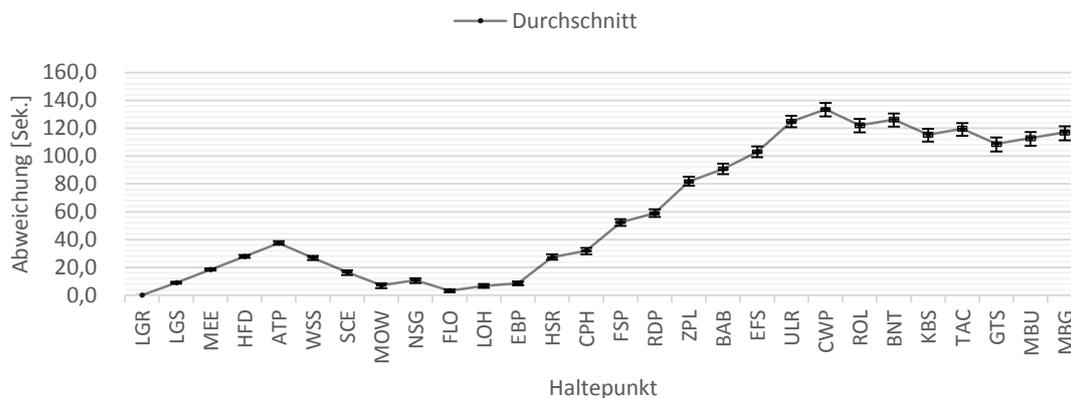


Abbildung 5.45: Abweichungsverlauf, Linie 6-T02, Fahrplan KVBT2001-

sieben Minuten hinter Fahrzeugen von Linie 15-T02 abfahren. In der Folge wird Fahrzeugen von Linie 6-T02 auch hier im Simulationsmodell eine erhöhte Haltezeit aufgrund erhöhter Passagierzahlen zugewiesen, die sie auf den innerstädtischen Streckenabschnitten mit ihren kurzen geplanten Fahrzeiten nicht wieder gutmachen können. Erst nachdem Fahrzeuge von Linie 6-T02 nach Haltestelle CWP wieder alleine verkehren, können sie einen Teil der eingefahrenen Verspätung aufholen. Da sich aber auch dort vermehrt kurze geplante Fahrzeiten finden, ist das Potenzial zum Aufholen von Verspätungen, fahrplanunabhängig, vergleichsweise gering. Unter Fahrplan KVBT2001<sup>+</sup> sind die Abfahrtszeiten an Haltestelle EBP besser koordiniert: Beginnend mit Fahrzeugen von Linie 17-T02 fahren Fahrzeuge von Linie 6-T02 in einem planmäßigen Abstand von vier Minuten ab, gefolgt von Fahrzeugen der Linie 19-T02 in einem Abstand von zwei Minuten. Wiederum zwei Minuten später fahren schließlich Fahrzeuge von Linie 15-T02 an Haltestelle EBP ab. Dies erklärt auch den ebenfalls unter Fahrplan KVBT2001<sup>+</sup> merkbaren – allerdings wesentlich schwächeren – Anstieg in der Verspätung von Fahrzeugen der Linie 6-T02 ab Haltestelle EBP. Auch unter diesem Fahrplan bekommen sie, aufgrund des im Vergleich zum Planwert von zwei Minuten leicht erhöhten Sicherheitsabstands, systematisch erhöhte Haltezeiten zugewiesen.

Fahrzeuge von Linie 15-T02 verkehren zwischen den Haltestellen *Thielenbruch (THB)* und *Ubierring (UBI)* und teilen sich ihre Starthaltestelle mit Fahrzeugen von Linie 3-T03. Hierin liegt auch der Grund für den stärkeren Anstieg in Verspätung zwischen den Haltestellen THB und *Dellbrück, Hauptstraße (DHS)*, sowie *Neufelder Str. (NFS)* und *Maria-Himmelfahrt Str. (MHS)* unter Fahrplan KVBT2001<sup>-</sup> (siehe Abbildung 5.46 und Abbildung 5.47). Die unter diesem Fahrplan unregelmäßigeren Sicherheitsabstände von neun bzw. einer Minute(n) führen auch hier zu systematisch längere Haltezeiten. An der Haltestelle *Holweide, Vischeringstr. (VRS)* kommen Fahrzeuge von Linie 13-T02 zu den Fahrzeugen der Linien 3-T02 und 15-T02 hinzu. Sie fahren unter Fahrplan KVBT2001<sup>-</sup> sechs Minuten vor Fahrzeugen

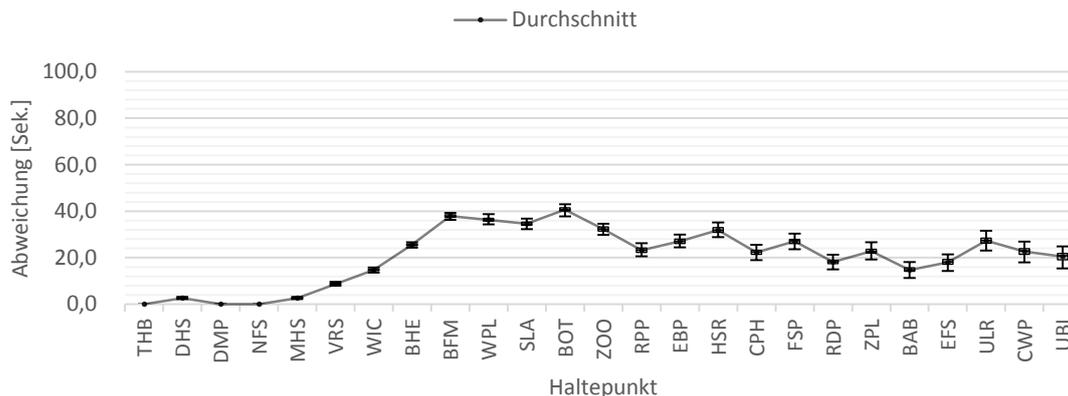


Abbildung 5.46: Abweichungsverlauf, Linie 15-T02, Fahrplan KVBT2001+

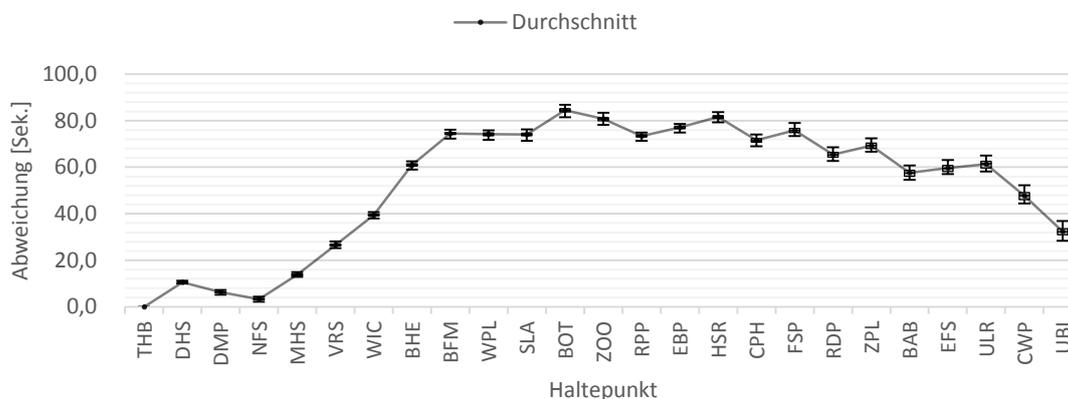


Abbildung 5.47: Abweichungsverlauf, Linie 15-T02, Fahrplan KVBT2001-

gen der Linie 15-T02 dort ab, wodurch sich deren Haltezeiten zwar leicht aber nicht ausreichend verkürzen, um die kurzen geplanten Fahrzeiten auf den Folgestrecken auszugleichen. An Haltestelle *Buchheim, Herler Str. (BHE)* stoßen Fahrzeuge von Linie 16-T02 zu den Fahrzeugen von Linien 3-T02, 13-T02 und 15-T02. Sie fahren dort unter Fahrplan KVBT2001- eine Minute nach Fahrzeugen von Linie 3-T02 bzw. zwei Minuten nach Fahrzeugen von Linie 15-T02 ab, haben also keine Auswirkungen auf deren Haltezeiten. Auch unter Fahrplan KVBT2001+ steigt die Verspätung zwischen den Haltestellen MHS und *Bahnhof Mühlheim (BFM)* an. Da Fahrzeuge von Linie 15-T02 unter ihm aber nur vier Minuten nach Fahrzeugen von Linie 3-T02 abfahren und entsprechend kürzere Haltezeiten zugewiesen bekommen, haben die kurzen geplanten Fahrzeiten unter diesem Fahrplan weniger starke Auswirkungen. Zwischen den Haltestellen BFM und EFS zeigen die Verläufe der Abfahrtszeitabweichungen unter beiden Fahrplänen dann ein ähnliches Verhalten, wenn auch um die entsprechende Ausgangsverspätung verschoben. Ab der Haltestelle *Ulrepforte (ULR)* kann schließlich aufgrund des einminütigen Sicherheitsabstandes zwischen Fahrzeugen von Linie 15-T02 und 17-T02 unter Fahrplan KVBT2001- im Vergleich zum vierminütigen Abstand unter Fahrplan KVBT2001+ eine stärkere Reduktion der Verspätung als unter Fahrplan KVBT2001+ beobachtet werden.

Der Streckenverlauf von Linie 17-T01 führt von der Haltestelle UBI zur Haltestelle *Niehl, Sebastianstraße (NIS)*. Der Verlauf der Abfahrtszeitabweichungen ihrer Fahrzeuge zeigt zunächst bis zur Haltestelle EFS unter beiden Fahrplänen das gleiche Verhalten (siehe Abbildung 5.48 und Abbildung 5.49). Ab dort sammeln Fahrzeuge dieser Linie unter Fahrplan KVBT2001- jedoch, wie bereits beschrieben, sukzessive Verspätung ein. Grund dafür ist zunächst eine schlechtere Koordination mit der an Haltestelle EFS hinzustoßenden Linie 12-T01, durch ungleichmäßige Sicherheitsabstände, zusammen mit der kurzen Fahrzeit zur Haltestelle *Barbarossaplatz (BAB)* von 60 Sekunden. Kurz vor dieser Haltestelle trennen sich darüber hinaus Linien 12-T01 und 16-T01 von Linien 6-T01, 15-T01 sowie 17-T01 und Linie 19-T01 stößt hinzu. In Folge dessen fahren Fahrzeuge von Linie 17-T01 unter Fahrplan KVBT2001- in einem Abstand von sieben Minuten zu den ihnen vorausfahrenden Fahrzeugen

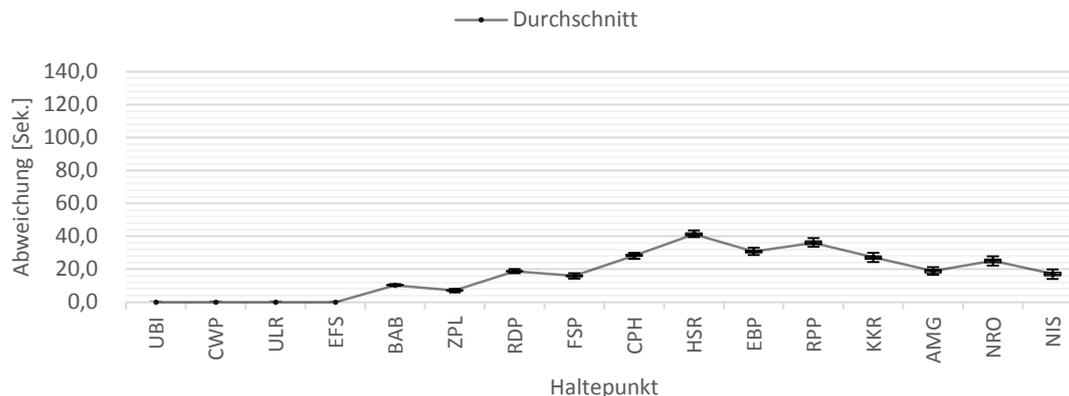


Abbildung 5.48: Abweichungsverlauf, Linie 17-T01, Fahrplan KVBT2001<sup>+</sup>

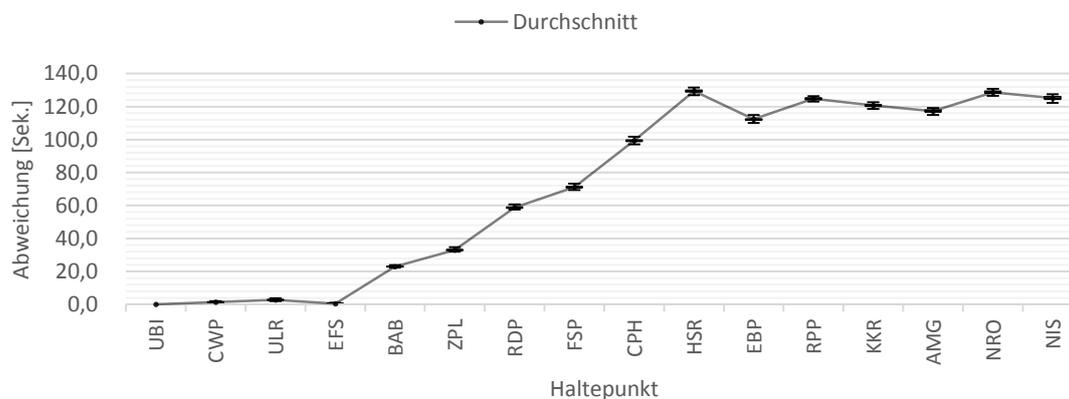


Abbildung 5.49: Abweichungsverlauf, Linie 17-T01, Fahrplan KVBT2001<sup>-</sup>

der Linie 15-T01, wodurch ihnen systematisch längere Haltezeiten zugewiesen werden. Hinzu kommt, dass die innerstädtischen Strecken sehr kurze geplante Fahrzeiten von 60 Sekunden aufweisen, sodass die durch die verlängerten Passagierwechsel eingefahrenen Verspätungen nicht gutgemacht werden können. Dieser Prozess kann erst durchbrochen werden, sobald sich Linien 17-T01 und 19-T01 von den restlichen Linien trennen und gemeinsam weiter zur Haltestelle NIS fahren. Zwar ist der Sicherheitsabstand zwischen diesen Linien unter Fahrplan KVBT2001<sup>-</sup> mit acht bzw. zwei Minuten größer als der Sicherheitsabstand zwischen Linie 17-T01 und 15-T02, liegt aber näher am örtlichen Planwert wodurch sich zusammen mit den längeren geplanter Fahrzeiten auf den Streckenabschnitten zwischen den Haltestellen *Reichenspergerplatz* (RPP) und NIS kein weiterer deutlicher Anstieg der Verspätung beobachten lässt. Unter Fahrplan KVBT2001<sup>+</sup> kommt es zwar ebenfalls zu einem leichten Anstieg der Verspätung ab Haltestelle EFS, aufgrund der gleichmäßigeren Sicherheitsabstände bleibt sie aber mit maximal 41,2 Sekunden an Haltestelle HSR deutlich niedriger.

Linie 19-T02 verläuft von der Haltestelle NIS zur Haltestelle *Klettenbergpark* (KLB) und ihre Fahrzeuge fahren bis zur Haltestelle BAB gemeinsam mit denen von Linie 17-T02. Ab der Haltestelle EBP kommen Fahrzeuge von Linien 6-T02 und 15-T02 hinzu, wodurch sich der Verspätungsanstieg zwischen den Haltestellen *Christophstr./Mediapark* (CHP) und BAB unter Fahrplan KVBT2001<sup>-</sup> erklärt (siehe Abbildung 5.50 und Abbildung 5.51). Er ist eine Folge der bereits beschriebenen Verspätungsansammlung durch Fahrzeuge von Linie 6-T02 ab der Haltestelle EBP (vgl. Abbildung 5.45). Da Fahrzeuge von Linie 19-T02 ab dieser Haltestelle unter Fahrplan KVBT2001<sup>-</sup> nur einen Sicherheitsabstand von einer Minute zu den ihnen vorausfahrenden Fahrzeugen von Linie 6-T02 besitzen, überträgt sich ein Teil der Verspätung dieser Fahrzeuge auf die Fahrzeuge von Linie 19-T02. Sobald sich der Verlauf von Linie 19-T02 nach der Haltestelle BAB von dem der anderen Linien trennt, kann ein Teil der Verspätung gutgemacht werden. Allerdings sorgt die schlechte Koordination mit Fahrzeugen der Linie 18-T02 ab der Haltestelle *Eifelwall* (EFW) und die daraus resultierenden verlängerten Haltezeiten zusammen mit den kurzen geplanten Fahrzeiten nur wenig später für einen

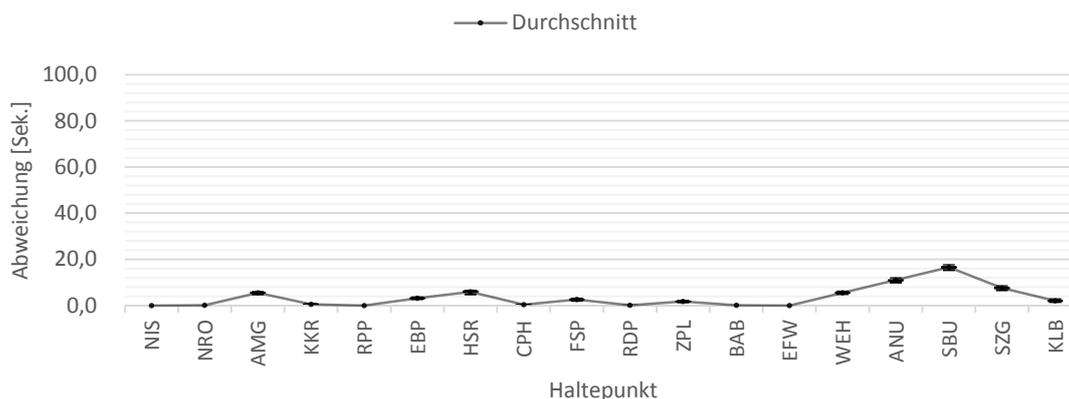


Abbildung 5.50: Abweichungsverlauf, Linie 19-T02, Fahrplan KVBT2001+

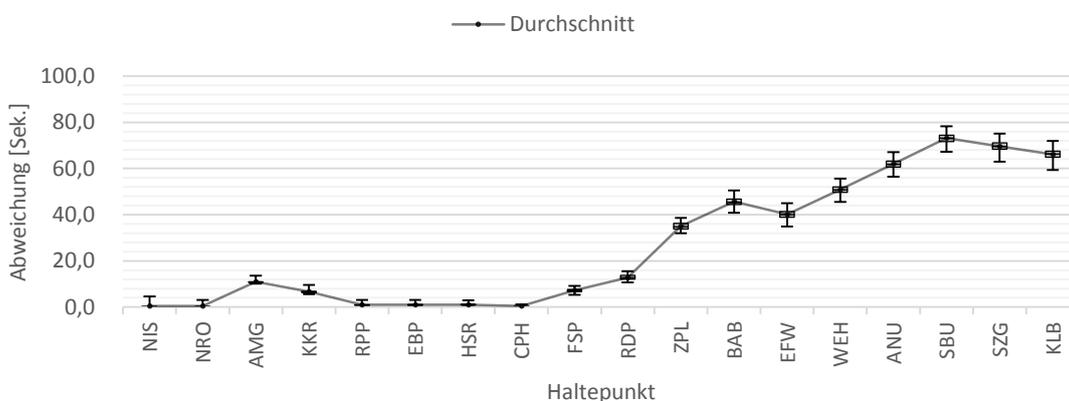


Abbildung 5.51: Abweichungsverlauf, Linie 19-T02, Fahrplan KVBT2001-

erneuten Anstieg der Verspätung. Unter Fahrplan KVBT2001+ kommt es hingegen, aufgrund der gleichmäßigeren Sicherheitsabstände, nicht zu solchen Problemen. Allerdings kann auch hier ein fahrzeitbedingter, leichter, systematischer Anstieg in der Verspätung ab Haltestelle EFW beobachtet werden.

**Fazit.** Die Untersuchungen bestätigen, dass Optimalfahrplan KVBT2001+ eine bessere Systemperformanz aufweist als Fahrplan KVBT2001-. Unter ihm zeigt sich eine signifikant geringere Zahl stark verspäteter Abfahrten und in der Folge eine stark reduzierte durchschnittliche Verspätung. Darüber hinaus zeigen die Untersuchungen der Verläufe der Abfahrtszeitabweichungen, dass die geringere Pünktlichkeit und höhere Zahl stärker verspäteter Abfahrten unter Fahrplan KVBT2001- in der Tat ein Resultat von fahrplanbedingten, unregelmäßigeren Sicherheitsabständen und damit einhergehenden Koordinierungs-problemen auf gemeinsam genutzten Netzwerkressourcen ist.

Allgemein bestätigen die Ergebnisse der Untersuchungen die Plausibilität des Modellverhaltens im Hinblick auf Stadtbahnsysteme und seine Eignung zur Bewertung von Stadtbahnfahrplänen. Es zeigt sich aber auch, dass es, wie bereits beim Bussystem in Abschnitt 5.3.1 gesehen, aufgrund der spärlichen Datenbasis zu kleineren Simulationsartefakten kommt; erneut seien dazu beispielhaft die systematischen leichten Verspätungen auf Strecken mit zu kurzer geplanter Fahrzeit erwähnt. Auch hier gilt, dass diese Artefakte erst reduziert werden können sobald eine umfangreichere Datenbasis vorliegt. Die auftretenden Artefakte sind dabei allerdings nicht signifikant genug, um die Plausibilität des Simulationsmodells in Frage zu stellen.

### 5.3.3. Leistungsfähigkeit der Simulationsanwendung

Zusätzlich zur Plausibilität des Simulationsmodells soll auch die Leistungsfähigkeit der Simulationsanwendung, in Form einer Laufzeitanalyse, untersucht werden. Dazu wird zusätzlich zu den Modellen der Netze UCL 2015 und KVB-T 2001 ein Modell des Kölner Busnetzes aus dem Jahr 2001 herangezogen.

Für jedes dieser Netze werden jeweils 100 Simulationsläufe für den Optimalfahrplan und den schlechtesten noch gültigen Fahrplan durchgeführt. Anschließend werden die ermittelten Kennzahlen, wenn möglich, über die Läufe beider Fahrpläne gemittelt. So wird vermieden, dass die ermittelten Laufzeiten vom eingesetzten Fahrplan abhängen. Diese Laufzeiten können dabei in vier Phasen aufgeteilt werden:

- *Initiale Preprocessingphase:* Diese Phase wird einmal pro Start der Anwendung durchlaufen. In ihr werden alle zur Durchführung der Simulationsläufe notwendigen Daten aus der durch die Nutzerin angegebenen Datenbank ausgelesen sowie die entsprechenden Simulationsentitäten erzeugt und dem Simulationsmodell hinzugefügt. Hier wird auch die Erstellung des (künstlichen) Umlaufplans angestoßen, falls keine entsprechenden Daten zur Verfügung stehen.
- *Simulationsphase:* Hierbei handelt es sich um die von einem einzelnen Simulationslauf benötigte reine Simulationszeit, d.h. die Zeit die zwischen der Verarbeitung der Ereignistypen SIMULATION\_START und SIMULATION\_END vergeht.
- *Laufabhängige Postprocessingphase:* Im Laufe dieser Phase werden die während eines einzelnen Simulationslaufs erfassten Rohdaten bzw. die diese Daten haltenden Objekte serialisiert und auf der Festplatte zwischengespeichert, um Arbeitsspeicher freizugeben. Insbesondere bei umfangreichen Experimenten mit vielen Simulationsläufen kann es andernfalls zu starken Leistungseinbrüchen kommen, weil statistische Daten bereits abgeschlossener Simulationsläufe noch im Speicher gehalten werden ohne dass sie verwendet werden und nachfolgende Simulationsläufe so weniger Speicher zur Verfügung haben. Im Rahmen dieser Phase werden, auf Wunsch der Nutzerin, auch statistische Kennzahlen für die einzelnen Simulationsläufe ermittelt und im CSV-Format abgespeichert. Falls bei der Analyse der statistischen Daten des Gesamtexperiments Unregelmäßigkeiten auftreten, kann so nachverfolgt werden ob es sich dabei um ein systematisches oder vereinzelt Problem handelt.
- *Finale Postprocessingphase:* Diese Phase wird ebenfalls nur einmal pro Start der Anwendung angestoßen und findet nach Abschluss des letzten Simulationslaufs statt. Hier werden die vorher während der einzelnen Simulationsläufe erfassten und serialisierten Rohdaten wieder von der Festplatte gelesen und für alle Entitäten die über alle Läufe aggregierten statistischen Kennzahlen ermittelt.

Tabelle 5.10 zeigt die ermittelten Laufzeiten für die drei Netze.

Netz	Preprocessing (Initial)	Simulation (pro Lauf)	Postprocessing (pro Lauf)	Postprocessing (Final)	Gesamt
<b>UCL 2015</b>	4,81	0,39	0,03	144,78	191,59
<b>KVB-T 2001</b>	6,19	6,77	0,15	1.270,72	1.968,91
<b>KVB-B 2001</b>	9,31	12,88	0,30	2.229,22	3.556,53

*Tabelle 5.10: Durchschnittliche Simulationslaufzeit in Sekunden, aufgeteilt in einzelne Phasen*

Wie zu erwarten, steigt die Laufzeit phasenübergreifend mit zunehmender Komplexität des simulierten Netzes. Je umfangreicher das Netz, desto mehr Daten müssen in der Preprocessingphase eingelesen und in Modellkomponenten überführt werden. In den Postprocessingphasen müssen folglich auch mehr Daten verarbeitet werden. Mit zunehmender Komplexität des Netzes steigt neben der Zahl der Modelentitäten darüber hinaus auch die Zahl der während eines Simulationslaufs zu verarbeitenden Ereignisse (siehe dazu auch Tabelle 5.11), wodurch sich die Steigerung der reinen Simulationszeit erklärt. Die Zahl der zu verarbeitenden Ereignisse ist darüber hinaus auch von der Länge des zu simulierenden Fahrplans abhängig.

Vergleicht man die in den einzelnen Phasen verbrachte Zeit miteinander, fällt vor allem der netzunabhängig extrem hohe Anteil der finalen Postprocessingphase auf. Für UCL 2015 macht sie 75,6 Prozent der Gesamtlaufzeit aus und auch für KVB-T 2001 und KVB-B 2001 ist sie mit 64,5 Prozent bzw. 62,7 Prozent für einen Großteil der Gesamtlaufzeit verantwortlich. Verursacht wird dieser hohe Anteil durch die Deserialisierung der während der einzelnen Simulationsläufe erfassten Rohdaten. Für Netz UCL 2015 werden in der finalen Postprocessingphase beispielsweise durchschnittlich ca. 1,45 Sekunden zur Verarbeitung der Rohdaten eines einzelnen Simulationslaufs benötigt. Von diesen 1,45 Sekunden sind allerdings alleine 1,4 Sekunden zur Deserialisierung der Objekte mit den statistischen Daten notwendig.

Ereignistyp	Netz		
	UCL 2015	KVB-T 2001	KVB-B 2001
OPERATIONAL_DAY_START	1	1	1
OPERATIONAL_DAY_END	1	1	1
ROTATION_START	-	13	132
ROTATION_END	-	-	120
DEADHEAD_START	420	1.827	4.080
SERVICE_TRIP_START	420	1.800	4.080
SERVICE_TRIP_END	842	3.585	8.150
BOARDING_START	4.740	50.470	89.015
BOARDING_END	4.740	50.470	89.015
DRIVING_START	4.320	60.746	84.951
DRIVING_END	4.320	62.913	84.961
<b>Gesamt</b>	<b>19.804</b>	<b>231.826</b>	<b>364.506</b>

Tabelle 5.11: Durchschnittliche Zahl verarbeiteter Ereignisse

Hier besteht folglich noch erhebliches Verbesserungspotential, etwa indem nicht das Standardprotokoll zur Objektserialisierung der Java-Standardbibliothek, sondern ein selbst entwickeltes Serialisierungsprotokoll oder eine Open Source Variante verwendet wird. Ein Entfernen des (De-)Serialisierungsprozesses kommt hingegen nur bedingt in Frage. Bei umfangreichen Experimenten auf Basis großer Netze – mit einer entsprechend hohen Zahl an Entitäten – nimmt sonst mit zunehmender Dauer des Experiments der notwendige Heap-Speicher zu, wodurch die Java-Laufzeitumgebung mehr Zeit zur Garbage Collection benötigt. In der Folge steigt die pro Simulationslauf notwendige Zeit und die Leistungsfähigkeit der Anwendung sinkt.

Lässt man die in der finalen Postprocessingphase verbrachte Zeit außer Acht, zeigt die Simulationsanwendung ein mehr als annehmbares Laufzeitverhalten. Sowohl die in der initialen Preprocessingphase als auch die in der laufabhängigen Postprocessingphase verbrachte Zeit ist netzunabhängig vernachlässigbar gering. Abbildung 5.52 zeigt dies beispielhaft für Netz UCL 2015. Sie zeigt, wie sich die Zahl der durchzuführenden Simulationsläufe auf die (Gesamt-)Laufzeit der einzelnen Phasen auswirkt. Dazu wurde die Simulationsanwendung jeweils fünfmal für die angegebene Zahl an Simulationsläufen gestartet und die erhobenen Laufzeitwerte gemittelt.

Wie zu erwarten, steigt die Gesamtsimulationszeit linear mit der Zahl der durchzuführenden Läufe. Die für die initiale Preprocessingphase und die laufabhängige Postprocessingphase notwendige Zeit bleibt hingegen nahezu konstant bzw. steigt nur unmerklich. Abbildung 5.52 verdeutlicht allerdings auch erneut, den hohen Anteil den die finale Postprocessingphase ausmacht.

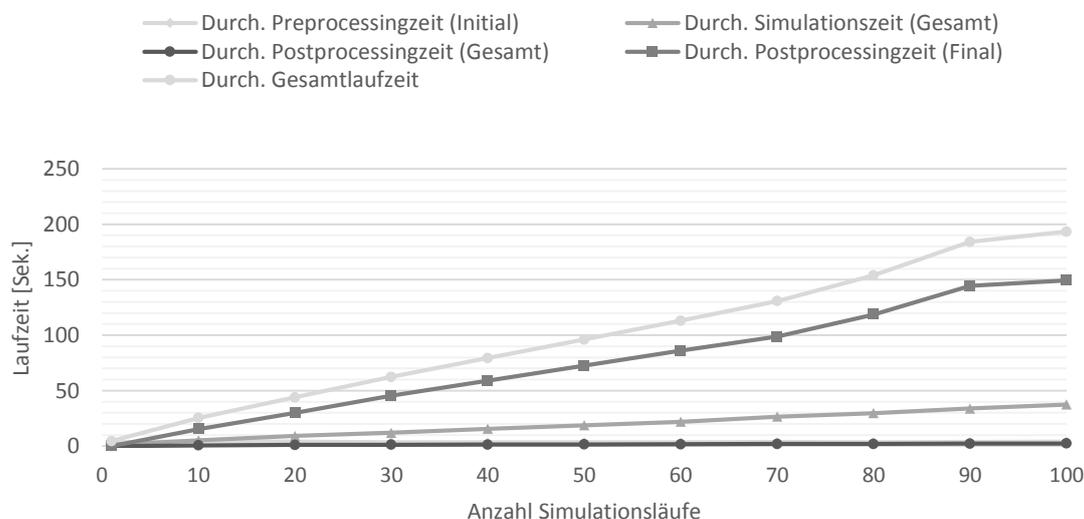


Abbildung 5.52: Durchschnittliche Laufzeit pro Phase in Abhängig der Zahl der durchzuführenden Simulationsläufe, UCL 2015

## 6. Verkehrsmanagement im Nah- und Fernverkehr

### 6.1. Einführung und Abgrenzung

Wie Schnabel und Lohse in [79] (S. 519) schreiben, ist Verkehrsmanagement die „zielorientierte Beeinflussung von Verkehrsangebot, Verkehrsnachfrage und Verkehrsabwicklung“. Es hat unter anderem die (bessere) Vernetzung der beteiligten Verkehrsträger, die Beeinflussung des Verkehrsverhaltens, die Erhöhung der Verkehrssicherheit und die Sicherstellung der Mobilität der Bevölkerung zum Ziel. Um diese Ziele zu erreichen werden Verkehrssysteme mit Hilfe von Telematiksystemen überwacht, die erhobenen Daten analysiert und bei Feststellung von Planabweichungen korrigierend in den operativen Betrieb eingegriffen (vgl. zu diesem Absatz [79], S. 519-523).

Wissenschaftlich liegt das Interesse im Bereich der *Entscheidungsunterstützungssysteme* (engl. *decision support systems*): Es sollen Methoden, Modelle und Systeme erforscht werden, die auf Basis von (Echtzeit-)Daten des Verkehrssystems dazu in der Lage sind Planabweichungen automatisch zu erkennen und potentielle Handlungsstrategien aufzuzeigen und/oder (halb-)automatisch zu implementieren. Häufig wird in diesem Zusammenhang auch von *dispatching control*, *delay management*, *conflict detection and resolution*, sowie *rescheduling* und *rerouting* gesprochen.

Die in der wissenschaftlichen Literatur eingesetzten Handlungsstrategien können grob in drei Kategorien eingeteilt werden: *Haltepunkt-Kontrolle* (engl. *station control*), *Verbindungs-Kontrolle* (engl. *interstation control*) und *Andere* (vgl. zu diesem und den folgenden Absätzen [30] sowie [109]).

Strategien der Kategorie Haltepunkt-Kontrolle umfassen haltepunktbezogene Maßnahmen. Dazu zählen die Anwendung von *Haltestrategien* zum bewussten Verzögern von Fahrzeugen an Kontrollpunkten, das *Überspringen von Haltepunkten*, aber auch die *Verkürzung von Fahrten* (engl. *short-turning*). Das Überspringen von Haltepunkten und die Verkürzung von Fahrten werden in der Regel nur bei beträchtlichen Störungen angewendet, während Haltestrategien auch schon bei kleineren Planabweichungen eingesetzt werden können. Haltestrategien können weiterhin – nicht zwingend trennscharf – danach unterschieden werden, ob die anzuwendende Verzögerung eines Fahrzeugs *abstands basiert* (engl. *headway based*) oder *fahrplanbasiert* bestimmt wird und ob die Anwendung der Haltestrategie *regelbasiert* oder aufgrund des Ergebnis eines *mathematischen Optimierungsmodells* initiiert wird. Abstands basierte Haltestrategien zielen auf die Einhaltung vorgegebener Fahrzeugabstände ab und finden sich vornehmlich in hochfrequenten Verkehrssystemen, in denen keine festen Abfahrtszeiten an Haltepunkten sondern eine (zu erwartende) Fahrzeugfrequenz angegeben sind. Fahrplanbasierte Haltestrategien dienen hingegen der Einhaltung geplanter Abfahrtszeiten und finden vor allem in Verkehrssystemen mit niedriger Taktfrequenz Anwendung.

Strategien der Kategorie Verbindungs-Kontrolle beinhalten z.B. Maßnahmen zur *Priorisierung von Fahrzeugen an Signalanlagen* um Fahrzeiten zu reduzieren, aber auch die *bewusste Verzögerung* von Fahrzeugen zwischen zwei Haltepunkten um bspw. Bunching-Effekte zu reduzieren.

Unter der Bezeichnung *Andere* finden sich schließlich Ansätze wie der Einsatz von *Leer- und Expressfahrten* (engl. *deadheading* resp. *expressing*) aber auch die *Verwendung von Reservefahrzeugen*. Expressfahrten bezeichnen einen Spezialfall von Leerfahrten. Während erstere das Überspringen von Haltepunkten durch leere Fahrzeuge bezeichnen, kann das Überspringen bei Expressfahrten auch durch mit Passagieren besetzte Fahrzeuge erfolgen.

Welche (Kombination von) Handlungsstrategie(n) in einem konkreten Fall eingesetzt werden sollte, ist ein komplexes Entscheidungsproblem, zu dem in den vergangenen 40 Jahren eine Reihe unterschiedlicher Ansätze entwickelt wurde. Eine Auswahl dieser Ansätze wird im Folgenden beschrieben.

### 6.2. Modelle für das Verkehrsmanagement im Nah- und Fernverkehr

Erste Modelle für das Verkehrsmanagement wurden unter anderem 1974 von Barnett in [8] und 1979 von Andersson et al. in [3] vorgestellt. Barnett beschreibt ein fahrplanbasiertes, konstruktives Optimierungsverfahren zur Approximation optimaler Haltezeiten individueller Fahrzeuge einer Linie an einem

einzelnen Kontroll-Haltepunkt. Ziel ist die Minimierung der erwarteten Wartezeit von am Kontroll-Haltepunkt zusteigenden Passagieren, sowie der Verspätung der bereits im Fahrzeug befindlichen Passagiere.

Die anzuwendenden Haltezeiten bestimmt Barnett vereinfachend auf Basis diskreter Zwei-Punkt-Verteilungen (engl. *two-point discrete distributions*) für die Verspätungssequenzen aufeinander folgender Fahrzeuge. Beginnend bei den ersten beiden aufeinander folgenden Fahrzeugen, werden in jedem Schritt des iterativen Verfahrens zunächst die möglichen Verspätungssequenzen, ihre Eintrittswahrscheinlichkeiten und damit die optimalen Haltezeiten der betroffenen Fahrzeuge berechnet. Anschließend wird auf Basis der Extremwerte der Zielfunktion geprüft, ob die Hinzunahme – d.h. vorsätzliche Verzögerung – eines weiteren Fahrzeugs eine Verbesserung der Zielfunktion verspricht. Ist dies der Fall, wird das entsprechende Fahrzeug der Menge zu verzögernder Fahrzeuge hinzugefügt und das Verfahren beginnt von vorne. Andernfalls wurden die optimale Zahl zu verzögernder Fahrzeuge sowie ihre zugehörigen Haltezeiten gefunden und das Verfahren wird beendet.

Die Anwendbarkeit seines Ansatzes demonstriert Barnett anhand eines Modells eines Teilabschnitts der U-Bahn von Boston. Es umfasst drei Haltepunkte, die von einer Linie im Fünf-Minuten-Takt bedient werden. Der Autor kann zeigen, dass sein Verfahren die durchschnittliche Wartezeit bereits bei der Berücksichtigung von nur vier aufeinander folgenden Fahrzeugen auf 2,62 Minuten reduziert und damit nah an den Optimalwert von 2,5 Minuten herankommt. Nach Beendigung des Verfahrens konnte die durchschnittliche Wartezeit schließlich auf 2,53 Minuten reduziert werden.<sup>19</sup>

Einen gänzlich anderen Ansatz, der auszugsweise bereits in Kapitel 4.2 beschrieben wurde, verfolgen Anderson et al. Die Autoren beschreiben ein mesoskopisches, ereignisbasiertes, interaktives Simulationsmodell, das durch die Nutzerin unterbrochen werden kann. Ist die Simulation unterbrochen, können über Konsolenbefehle die folgenden Verkehrsmanagementstrategien veranlasst werden:

- *EXTRABUS*: Einsetzen eines neuen Fahrzeugs.
- *TOGARAGE*: Ein zu spezifizierendes Fahrzeug beendet seine Fahrt an einem ebenfalls zu spezifizierenden Haltepunkt seiner Route und kehrt in das Depot zurück.
- *BREAKDOWN*: Ein zu spezifizierendes Fahrzeug kehrt in das Depot zurück, sobald es seine aktuelle Fahrt beendet hat.
- *SUBSTITUTEBUS*: Ein neues Fahrzeug wird als Ersatz für ein zu spezifizierendes Fahrzeug eingesetzt.
- *HOLDBUS*: Ein zu spezifizierendes Fahrzeug wird nach Abschluss des Passagierwechsels an einem zu spezifizierenden Haltepunkt für eine wiederum zu spezifizierende zusätzliche Zeitspanne verzögert.

Im Anschluss an die Befehlseingabe kann die Simulation fortgesetzt und die Folgen der gewählten Strategie untersucht werden. Die Autoren zeigen die korrekte Funktionalität ihrer Anwendung an einigen einfachen künstlichen Beispielen, stellen jedoch keine realen Anwendungsbeispiele vor.

Ein neuerer simulationsbasierter Ansatz wird von Borne et al. in [15] vorgestellt. Die Autoren beschreiben ein Entscheidungsunterstützungssystem, das auf Basis eines Multi-Agenten-Ansatzes und eines Genetischen Algorithmus Halte- und Fahrzeiten sowie Linienverläufe anpasst, um Warte-, Transfer- und Reisezeiten von Passagieren zu minimieren. Dazu verwenden Borne et al. fünf Typen von Agenten: *Fahrzeug-* und *Haltepunkt-Agenten* dienen der Ausführung und Überwachung des (theoretischen) Fahrplans, während *Störungs-*, *Zonepert-* und *Zonereg-Agenten* für die Abbildung, Analyse sowie Auflösung von Störungen verwendet werden.

Fahrzeugbewegungen werden im Modell von Borne et al. mesoskopisch abgebildet: In jedem Simulationsschritt berechnen Fahrzeug-Agenten ihre geschätzte Ankunftszeit am nächsten Haltepunkt auf Basis des aktuell simulierten Verkehrszustands, ihrer Entfernung zum vorhergehenden Haltepunkt, ihrer aktuellen Geschwindigkeit und der von ihnen auf der letzten Verbindung benötigten Fahrzeit.

Zur Überwachung des Fahrplans vergleichen Haltepunkt-Agenten die Ankunftszeiten von Fahrzeug-Agenten mit den im Fahrplan angegebenen. Überschreitet die gemessene Abweichung einen vorgegebenen

---

<sup>19</sup> Leider macht Barnett keine Angaben darüber wie viele Fahrzeuge zur Erreichung dieses Wertes verzögert werden mussten.

Grenzwert, wird sie als Störung eingestuft und Verkehrsmanagementmaßnahmen veranlasst, indem ein Störungs-Agent erzeugt wird. Dieser erzeugt zunächst einen Zonepert-Agenten zur Analyse der Störung und je nach ihrer Schwere auch einen Zonereg-Agenten zur Bestimmung der optimalen Verkehrsmanagementmaßnahmen.

Zonepert-Agenten schätzen die zu erwartende Entwicklung des Systems ohne Verkehrsmanagementeingriffe mittels Simulation ab und vergleichen diesen potentiellen, zukünftigen Zustand mit dem ursprünglich geplanten Zustand. Darüber hinaus können Zonepert-Agenten auf Basis von Verhaltensregeln einfache Handlungsstrategien formulieren.

Zonereg-Agenten wenden hingegen einen Genetischen Algorithmus an, um die optimale Kombination von Handlungsstrategien zu approximieren. Dabei werden Individuen der initialen Population erzeugt, indem zunächst entschieden wird an welchen Haltepunkten ihrer ursprünglichen Routen die betroffenen Fahrzeuge halten. Anschließend wird für aufeinander folgende Haltepunkte aus einer Menge vorab definierter Alternativ-Verbindungen die bestpassende ausgewählt. Potentielle Kriterien für diese Auswahl können die Entfernung zwischen den Haltepunkten, der Fahrzeugtyp, aber auch die Fahrzeit oder der aktuelle Verkehrsstatus sein. Sind Haltepunkte und Verbindungen ausgewählt wird ein Fahrplan generiert, indem die Halte- und Fahrzeiten festgelegt werden. Die initiale Population beinhaltet darüber hinaus auch Individuen, die aufgrund der durch den Zonepert-Agenten vorgeschlagenen Handlungsstrategien erzeugt wurden. Ausgehend von der initialen Population, werden neue Individuen über Crossover- und Mutations-Operatoren erzeugt und per Tournament-Selektion der Population hinzugefügt. Dies geschieht so lange bis ein, von Borne et al. nicht näher spezifiziertes, Abbruchkriterium erreicht wurde. Im Anschluss werden mittels Simulation die Auswirkungen der Implementierung der durch das beste Individuum vorgegebenen Handlungsstrategien den Auswirkungen bei unkontrolliertem Systemzustand gegenübergestellt und die resultierenden Warte-, Transfer- und Reisezeiten der Passagiere miteinander verglichen.

Die Plausibilität ihres Ansatzes zeigen die Autoren anhand eines kleinen, künstlichen, bi-modalen Verkehrssystems, in dem sie ein Bahnfahrzeug an einem Haltepunkt künstlich verzögern. Die Anwendung berechnet innerhalb von 20 Sekunden einen neuen Fahrplan, in dem die dem gestörten Fahrzeug vorausfahrende Bahn künstlich verzögert wird, um einen gleichmäßigen Fahrzeugabstand zu gewährleisten. Außerdem werden in der Folge zwei Busse verzögert, um Umsteigeverbindungen zwischen den beiden Verkehrsmodi sicherzustellen.

Ein allgemeiner Modellierungsansatz zur Abbildung von (regelbasiertem) Expertenwissen wird von Fay in [35] vorgestellt. Dazu modelliert er von Experten formulierte Verkehrsmanagementregeln mit Hilfe von z.T. hierarchischen Fuzzy-Regeln und Zugehörigkeitsregeln. Zur Auflösung konkreter Störsituationen – und graphischen Darstellung – verwendet der Autor wiederum Fuzzy-Petri-Netze. Zur Abschätzung der Güte der so erhaltenen Handlungsempfehlungen werden sie auch hier simuliert und den Simulationsergebnissen des unkontrollierten Falls gegenübergestellt.

Fay wendet sein Verfahren zur Validierung auf ein einfaches Beispiel an, in dem entschieden werden muss ob eine Umsteigeverbindung im Fernverkehr gehalten werden soll oder nicht, d.h. ob ein pünktlicher Zug auf ein verspätetes Zubringerfahrzeug warten soll. Anhand von 14 Szenarien, die sich anhand der Verspätung, der Zahl umsteigender Passagiere, des Takts und der noch zurückzulegenden Fahrtstrecke unterscheiden, zeigt der Autor, dass sein Verfahren ähnliche Entscheidungen wie reale Experten trifft.

Aktuellere (Echtzeit-)Optimierungsansätze mit Fokus auf Haltepunkt-Kontrolle finden sich unter anderem in [106], [109] und [40]. Zolfaghari et al. stellen in [109] ein abstands-basiertes, mathematisches Optimierungsmodell zur Bestimmung von Haltezeiten vor, das die Wartezeit aller Passagiere auf einer Route minimiert und Echtzeit-Informationen über Fahrzeuge sowie Fahrzeugkapazitäten und von diesen beeinflusste Schwankungen im Ein-/Ausstiegsprozess berücksichtigt. Das Modell ist für die Online-Optimierung ausgelegt und bezieht nur eine begrenzte Zahl an Haltepunkten pro Fahrzeug in die Optimierung ein. Darüber hinaus werden Haltezeiten nur für den aktuellen bzw. nächsten zu bedienenden Haltepunkt eines Fahrzeugs bestimmt.

Zur Lösung ihres Optimierungsproblems verwenden Zolfaghari et al. einen Simulated Annealing Ansatz (siehe z.B. [32]) mit inhomogenem und adaptivem Abkühlungsprozess, d.h. die Temperatur wird

nach jedem Lösungsübergang verringert, kann im Verlauf des Optimierungsprozesses aber auch wieder ansteigen.

Zur Validierung ihres Ansatzes wenden ihn die Autoren auf eine künstliche Ringlinie mit 26 Haltepunkten an, auf der 18 Busse verkehren. Zolfaghari et al. untersuchen ein Szenario in dem eines der Fahrzeuge zu früh ist und zeigen, dass ihr Verfahren die durch die Planabweichung induzierte zusätzliche Gesamtwarezeit im Vergleich zum unkontrollierten Fall um 47,5 Prozent reduziert.

In [106] stellen Yu und Yang ebenfalls einen Optimierungsansatz für Haltezeiten zur Minimierung von Passagierwartezeiten unter Berücksichtigung von Echtzeit-Informationen und Kapazitätsbeschränkungen vor. Anders als beim Ansatz von Zolfaghari et al., entwerfen Yu und Yang eine fahrplan- und regelbasierte Haltestrategie, die darüber hinaus die erwartete Pünktlichkeit am Folgehaltepunkt eines Fahrzeugs berücksichtigt. Nur wenn ein Fahrzeug an seinem aktuellen Haltepunkt zu früh ist und auch zu erwarten ist, dass es an seinem nächsten zu bedienenden Haltepunkt frühzeitig abfahren wird, wird es an seinem aktuellen Haltepunkt verzögert. Die anzuwendende Verzögerung bestimmen die Autoren dabei mit Hilfe eines Genetischen Algorithmus für alle zu verzögernden Fahrzeuge simultan. Der Algorithmus startet mit einer zufällig erzeugten Initialpopulation, in der einzelne Individuen die für jedes Fahrzeug anzuwendende Verzögerung codieren. Ausgehend von dieser Population werden mittels Ein-Punkt-Crossover, Roulette-Selektion und einem Mutationsoperator der einzelne Haltezeiten modifiziert so lange neue Generationen von Individuen erzeugt, bis eine maximale Zahl an Generationen erreicht wurde. Zusätzlich implementieren Yu und Yang Elitismus in ihrem Genetischen Algorithmus, d.h. eine feste Zahl der besten Individuen einer Generation wird in die Nachfolgeneration übernommen.

Ihr Verfahren validieren die Autoren durch Simulation der Buslinie 4 in Dalian, China, die 34 Haltepunkte umfasst. Sie vergleichen ihren Ansatz mit einer statischen fahrplanbasierten Haltestrategie sowie mit dem vollkommen unkontrollierten Fall. Wie zu erwarten, resultiert der Einsatz von Haltestrategien in einer Reduktion der Passagierwartezeit. Im Vergleich mit der statischen Haltestrategie sorgt der Ansatz von Yu und Yang aber für eine geringere Erhöhung der Wartezeit von Passagieren, die sich bereits im Fahrzeug befinden. Grund dafür ist, dass Fahrzeuge, die Gefahr laufen zu spät an ihrem Folgehaltepunkt abzufahren, nicht verzögert werden und die bereits im Fahrzeug befindlichen Passagiere so mit einer höheren Wahrscheinlichkeit pünktlich an ihrem Zielhaltepunkt eintreffen. Dies geschieht dabei auf Kosten von neu hinzusteigenden Passagieren, die ihr Fahrzeug verpassen, da es zu früh am Haltepunkt abfährt. In Folge dessen fällt die Reduktion der Wartezeit neu hinzusteigender Passagiere beim Ansatz von Yu und Yang leicht geringer aus als unter der statischen Haltestrategie.

Ginkel und Schöbel stellen in [40] Umsteigeverbindungen in den Mittelpunkt. Sie formulieren auf Basis eines Ereignis-Aktivitäts-Netzwerks ein bi-kriterielles lineares ganzzahliges Optimierungsproblem, das sowohl die Fahrzeugverspätungen als auch die Zahl der verpassten Umsteigeverbindungen minimiert. Für den Spezialfall, bei dem die Menge einzuhaltender Umsteigeverbindungen fixiert ist, zeigen die Autoren, dass der optimale Fahrplan (i.S.d. Fahrzeugverspätung) mit Hilfe der *Critical Path Methode (CPM)* bestimmt und die in [13] vorgestellte Schrumpfmethode zur Reduzierung umfangreicher Netzwerke angewendet werden kann. Diese Schrumpfmethode wendet die drei Operatoren *serielles/paralleles Verschmelzen von Kanten* und *Knotenreduktion* so lange iterativ auf das Netzwerk an, bis es nur noch aus einer Kante zwischen dem Start- und Zielknoten besteht, deren Kantengewicht die minimal notwendige Projektdauer angibt.

Auf Basis dieser Überlegungen entwickeln Ginkel und Schöbel anschließend einen Backtracking-Algorithmus zur Bestimmung der pareto-optimalen Lösungen des Ursprungsproblems, der eine Erweiterung des in [29] vorgestellten Ansatzes zur Lösung des *discrete time/cost trade-off Problems* ist. Dazu erweitern sie alle Umsteige-Kanten des EANs mit zwei Labeln, die je eine der potentiellen Entscheidungen (Verbindung halten oder nicht) sowie die damit verbundenen Folgen (Wartezeit bzw. Zahl nicht bedienter Passagiere) abbilden. Das so modifizierte Netzwerk wird mit Hilfe einer leicht adaptierten Schrumpfmethode soweit wie möglich reduziert. Kann das Netzwerk nicht weiter reduziert werden, wird die früheste Umsteige-Kante im Netzwerk gewählt, eines der beiden Label fixiert und das Netzwerk mit den resultierenden neuen Kantenkosten weiter reduziert. Besteht das Netzwerk schließlich nur noch aus einer Kante wird die so erhaltene Lösung mit den bereits in vorherigen Schritten berechneten

Lösungen verglichen und dominierte Lösungen aus der Menge aller Lösungen gestrichen. Anschließend wird zum letzten Fixierungsschritt zurückgegangen, das dort nicht ausgewählte Label fixiert und das Verfahren wiederholt.

Ihr Verfahren validieren Ginkel und Schöbel anhand eines Modells der Lautertalbahn sowie angrenzenden Verkehrsmodi des Verkehrsverbund Rhein-Neckar. Das Anwendungsbeispiel umfasst 823 Haltestellen, 1.314 Fahrzeuge und je nach Szenario 45.000-85.000 Aktivitäten von denen bis zu 40.000 Umsteigeaktivitäten darstellen. Anhand diverser Szenarien mit unterschiedlichen Eingangsverspätungen und Zeitfenstern zeigen die Autoren, dass ihr Verfahren in vertretbarer Zeit – ca. eine Minute für ein Zeitfenster von einer Stunde – effiziente Lösungen findet. Dabei hängt die Laufzeit des Verfahrens, wie zu erwarten, signifikant von der Zahl zu berücksichtigender Umsteigeverbindungen ab.

Untersuchungen mit Fokus auf Methoden der Verbindungs-Kontrolle werden unter anderem von Corman et al. in [23] und [24] vorgestellt. In [23] beschreiben die Autoren die Anwendung des von ihnen entwickelten Entscheidungsunterstützungssystems *ROMA* (*Railway traffic Optimization by Means of Alternative graphs*) zur Evaluierung unterschiedlicher Lösungsalternativen nach Auftritt einer Störung. Das System nimmt eine Menge von durch Experten vorgegebenen, unterschiedlichen Lösungsalternativen als Input, berechnet neue gültige Fahrpläne und ermöglicht ihren Vergleich anhand diverser Performance-Indikatoren wie bspw. Reisezeit, Wartezeit oder Fahrzeugverspätung. Die zu bewertenden Handlungsalternativen können dabei das lokale Umleiten eines Fahrzeugs zwischen zwei Haltepunkten auf ein anderes Gleis, das globale Umleiten eines Fahrzeugs über andere Haltepunkte zu seinem Zielhaltepunkt sowie das Abbrechen von Fahrten umfassen.

Einen Ansatz zur Erstellung von Teilen solcher Handlungsalternativen, die Generierung optimaler Fahrzeugrouten, stellen Corman et al. in [24] vor. Dazu formulieren sie das *Conflict Detection and Resolution Problem (CDR)*, bei dem es darum geht für eine gegebene Menge von Fahrzeugen und gegebenen Fahrplan optimale Routen durch ein Streckennetz sowie Fahrzeugreihenfolgen auszuwählen, sodass der Fahrplan konfliktfrei und mit minimaler Verspätung durchgeführt werden kann. Ein Konflikt bezeichnet in diesem Kontext eine Situation in der zwei oder mehr Fahrzeuge zeitgleich den gleichen Gleisabschnitt durchqueren ohne einen vorgegeben Minimalabstand einzuhalten.

Zur Lösung des Problems teilen Corman et al. es in zwei Subprobleme auf: Auswahl von Routen aus einer Menge potentieller (vorab definierter) Routen und Terminierung der Fahrzeugreihenfolgen für eine gegebene Menge von Routen. Letzteres ist ein Spezialfall des CDR mit fixierten Routen (*Conflict Detection and Resolution with Fixed Routes, CDRFR*) und kann als disjunktives Programm, d.h. als lineares Programm mit Konjunktionen und Disjunktionen in den Nebenbedingungen, formuliert werden. Corman et al. verwenden zur Formulierung des CDRFR als disjunktives Programm *Alternativ-Graphen* (engl. *Alternative Graphs*). Ein Alternativ-Graph  $G = (N, F, A)$  ist ein Graph in dem die Knotenmenge  $N$  je einen Knoten für jede Entscheidungsvariable enthält; im Falle des CDRFR sind dies die Eintrittszeiten von Fahrzeugen in einen Gleisabschnitt. Die Mengen  $F$  und  $A$  sind Kantenmengen und stellen Beziehungen zwischen Entscheidungsvariablen, d.h. Nebenbedingungen dar.  $F$  ist die Menge der fixierten Kanten, d.h. Nebenbedingungen die zwingend eingehalten werden müssen. Beispiele für solche Nebenbedingungen sind bspw. minimale Abstände zwischen dem Eintreten eines Fahrzeugs in aufeinander folgende Gleisabschnitte. Die Menge  $A$  beinhaltet hingegen Paare von Alternativ-Kanten, repräsentiert also Paare von Nebenbedingungen von denen genau eine erfüllt sein darf/muss. Diese Nebenbedingungen bilden bspw. potentielle Alternativen in der Fahrzeugreihenfolge ab. In [27] stellen D'Ariano et al. einen Branch-and-Bound Algorithmus zur Lösung des CDRFR vor.

Durch die Modellierung mit Hilfe von Alternativ-Graphen entspricht die Anpassung von Fahrzeugrouten im CDR Änderungen der Menge  $F$ , so dass das Problem informal wie folgt definiert werden kann: Finde die Menge  $F$ , für die das CDRFR (möglichst) optimal – d.h. mit minimaler Fahrzeugverspätung – gelöst wird.

Um die optimale Menge  $F$  zu approximieren entwerfen Corman et al. in [24] einen Tabu-Suche Algorithmus. Dabei handelt es sich um eine deterministische Metaheuristik basierend auf lokaler Suche und einer Tabu-Liste, die unzulässige Lösungseigenschaften enthält. Letzteres soll vermeiden, dass das

Verfahren in lokalen Optima verweilt. Entscheidend für die Tabu-Suche ist die Definition der Nachbarschaft einer Lösung. Corman et al. definieren deswegen unterschiedliche Nachbarschaften, von denen die umfangreichste alle Mengen  $F'$  enthält, die sich von der aktuellen Lösung  $F$  genau in einer Route unterscheiden.

Das von den Autoren entworfene Verfahren verläuft nun wie folgt: Ausgehend von der planmäßigen Lösung wird mit Hilfe der Tabu-Suche eine optimale Routen-Menge bestimmt. Anschließend wird das CDRFR gelöst, bspw. mit dem Ansatz von D'Ariano et al. Erfüllt die erhaltene Lösung ein – nicht weiter spezifiziertes – Abbruchkriterium, wird das Verfahren beendet. Andernfalls wird die Tabu-Suche wieder aufgenommen und das Verfahren wiederholt. Das Gesamtproblem wird dabei nur für einen vorab definierten Zeithorizont, d.h. eine feste Anzahl von Fahrzeugen gelöst.

Zur Validierung ihres Ansatzes wenden ihn Corman et al. auf das niederländische Eisenbahnnetz um Utrecht und Den Bosch an. Dabei betrachten sie ein 15-Minuten-Fenster des Netzabschnitts zwischen Utrecht Lunetten und Den Bosch, in dem 13 Fahrzeuge verkehren und für das sie die Optimallösung des CDR mit Hilfe vollständiger Enumeration berechnen können. Sie untersuchen 15 verschiedene Störungsszenarien mit unterschiedlichen Verteilungen der Fahrzeugverspätungen und vergleichen die Optimallösung des CDR sowohl mit der Optimallösung des CDRFR ohne Routenänderung als auch mit der mit dem Tabu-Suche Algorithmus berechneten Lösung. Darüber hinaus vergleichen sie ihren Tabu-Suche Ansatz auch mit einem früheren Ansatz auf Basis einer einfachen lokalen Suche, der von D'Ariano et al. in [26] beschrieben wird. Die Ergebnisse zeigen, dass der Ansatz mittels Tabu-Suche stets innerhalb einer Sekunde die optimale Lösung findet, während der Ansatz mit lokaler Suche im Durchschnitt einen relativen Fehler von 127 Prozent aufweist.

Ansätze die sich mit anderen Kontrollstrategien beschäftigen werden beispielsweise in [37] und [107] vorgestellt. Fu et al. formulieren in [37] ein nicht-lineares ganzzahliges Programm zur Minimierung von Warte- und Reisezeiten mittels Expressfahrten. Dabei betrachten sie jeweils Paare von aufeinander folgenden Fahrzeugen einer Linie, von denen das hinterherfahrende Fahrzeug unter Umständen Haltepunkte überspringen darf, während das vorneweg fahrende Fahrzeug alle Haltepunkte der Route bedienen muss. Die Absicht der Autoren ist dabei, eine Mindestservicerate aufrecht zu erhalten um zu vermeiden, dass einige Haltepunkte nicht mehr bedient werden.

Ein effizientes Verfahren zur Lösung des von ihnen formulierten Optimierungsproblems stellen die Autoren leider nicht vor. Stattdessen bestimmen sie für einen einfachen Testfall einer 28 Haltepunkte umfassenden Buslinie in Waterloo, Kanada mit Hilfe vollständiger Enumeration die Optimallösung und führen mit Hilfe des Simulationsmodells *SimTransit* eine Sensitivitätsanalyse durch. Die Ergebnisse zeigen, dass die von Fu et al. gewählte Expressfahrt-Strategie nur bei hoher Nachfrage, moderater Fahrzeitvarianz und hoher Taktfrequenz (unter 10 Minuten) in einer Reduktion der Warte- und Reisezeit mündet.

Eine Leerfahrtstrategie zur Minimierung von Passagierwartezeiten und Leerfahrtkosten unter Berücksichtigung von Fahrzeugkapazitäten wird von Yu et al. in [107] besprochen. Anhand der Passagierfrage deklarieren die Autoren Hin- oder Rückrichtung einer Buslinie als Hauptrichtung und erlauben nach Beendigung einer Fahrt auf der Hauptrichtung eine Leerfahrt zu einem Haltepunkt der gleichen Richtung. Ihre Absicht ist, die stark ausgelastete Hauptrichtung zu entlasten ohne zusätzliche Fahrzeuge einsetzen zu müssen. Yu et al. formulieren dazu ein konstruktives Verfahren das mit einem rollenden Zeithorizont arbeitet und nur eine bestimmte Zahl an Fahrzeugen betrachtet. Zunächst bestimmt das Verfahren den aktuellen Servicelevel (hier auf Basis des Varianzkoeffizienten des Linientaktes) und die aktuelle und prognostizierte Zuverlässigkeit der Hauptrichtung (hier mit der *k-Nearest-Neighbor Methode*, siehe [9]). Unterschreitet die prognostizierte Zuverlässigkeit einen vorab definierten Grenzwert, wird überprüft ob es sinnvoll ist eine Leerfahrt zu veranlassen. Dazu wird sukzessive durch alle Haltepunkte der Hauptrichtung iteriert, um den optimalen Startpunkt der nächsten Fahrt zu bestimmen. Dabei wird die Wartezeitreduzierung auf der Hauptrichtung der Wartezeiterhöhung auf der Nebenrichtung gegenübergestellt und auch die notwendige Fahrzeit zum Starthaltepunkt berücksichtigt. Wurde am Ende ein Haltepunkt gefunden der die Gesamtzielfunktion minimiert, wird eine Leerfahrt veranlasst. Andernfalls führt das entsprechende Fahrzeug eine normale Fahrt auf der Nebenrichtung aus.

Ihr Verfahren testen Yu et al. an Daten für den morgendlichen Verkehr (07.00 Uhr bis 08.30 Uhr) der Buslinie 23 in Dalian, China. Sie umfasst insgesamt 19 Haltepunkte und wird in einem Takt von 2,5 Minuten befahren. Simulationen mit dem mikroskopischen Simulationsmodell *Paramics* zeigen, dass Busse, wie zu erwarten, Leerfahrten in der Regel nicht am Starthaltepunkt beenden, sondern vermehrt in der Mitte der Hauptrichtung, näher am Starthaltepunkt der Nebenrichtung. Grund dafür ist die Berücksichtigung der notwendigen Fahrzeit im Ansatz von Yu et al.

Der Vergleich mit dem unkontrollierten Fall zeigt, dass der Ansatz der Autoren in diesem Fall die Wartezeit- und Leerfahrtkosten um 11,8 Prozent reduziert. Leider vergleichen sie ihr Verfahren aber nicht mit dem Einsatz zusätzlicher und/oder größerer Fahrzeuge. Die Tatsache, dass Passagiere auf der Hauptrichtung bei einem Takt von 2,5 Minuten aufgrund von Kapazitätsengpässen mehr als ein Fahrzeug auslassen müssen, legt die Vermutung nahe, dass man es hier mit einer systematischen Unterschätzung der Nachfrage zu tun hat, die eine permanente Erhöhung der Kapazität rechtfertigen könnte.

Ein Optimierungsansatz der zeitgleich mehrere Handlungsstrategien unterschiedlicher Kategorien berücksichtigt wird von Hadas und Ceder in [42] vorgestellt und von Nesheli und Ceder in [70] verfeinert. Ziel ist die Minimierung der Reise- und Wartezeit von Passagieren in Verkehrssystemen mit stark synchronen Fahrplänen. In [42] werden dazu fünf Handlungsstrategien zur Erhöhung gleichzeitiger Abfahrten an ausgewählten Haltepunkten verwendet. Diese Strategien sind das Verzögern von Fahrzeugen an Haltepunkten, das Überspringen von Haltepunkten, die Anpassung von Fahrzeiten bzw. Geschwindigkeiten auf Verbindungen, das Verkürzen von Fahrten und der Einsatz von Expressfahrten. Wie die Mehrheit der bisher vorgestellten Modelle verwendet auch dieses Modell einen rollenden Zeithorizont sowie Echtzeitdaten über den Zustand des Verkehrssystems.

Hadas und Ceder formulieren sowohl einen Ansatz zur Online-Optimierung als auch einen Lösungsansatz auf Basis Dynamischer Programmierung und zeigen die Anwendbarkeit des Dynamischen Programmierungs Ansatzes anhand einer Simulation eines einfachen künstlichen Netzes, in dem drei unidirektionale Buslinien und eine – ebenfalls uni-direktionale – Bahnlinie an drei Umsteigestationen mit Hilfe von Haltezeit- und Fahrzeitanpassungen synchronisiert werden sollen. Während die Fahrzeiten der Busse dabei stochastischen Einflüssen ausgesetzt sind, sind die Fahrzeiten der Bahnen deterministisch modelliert. Darüber hinaus werden Haltepunkte und mit ihnen verbundene Passagierwechsel im Simulationsmodell nicht explizit abgebildet. Stattdessen wird eine Wahrscheinlichkeitsfunktion definiert, die den Fahrzeugabständen am ersten vorgelagerten gemeinsamen Segment eine Wahrscheinlichkeit für eine erfolgreiche Synchronisation zuordnet. Im Simulationsmodell wird dann über eine zufallsverteilte Variable entschieden, ob eine Synchronisation erfolgreich war oder nicht.

Anhand von 11 Szenarien mit unterschiedlichen Fahrplänen (synchronisiert oder nicht), Zeithorizonten (10-30 Minuten), Taktfrequenzen (hoch/niedrig) und Fahrzeitvarianz (niedrig, normal, hoch) untersuchen Hadas und Ceder schließlich den Nutzen ihres Ansatzes. Die Ergebnisse zeigen, dass durch die Anwendung der Haltezeit- und Fahrzeitstrategien eine Erhöhung der synchronen Abfahrten erzielt wird.

Nesheli und Ceder bauen in [70] auf diesem Modell auf, reduzieren allerdings die Menge betrachteter Taktiken. Sie betrachten nur noch das Verzögern von Fahrzeugen an Haltepunkten, das Überspringen von Haltepunkten und/oder Segmenten und das Verkürzen von Fahrten. Dabei werden beim Überspringen von Haltepunkten/Segmenten und Fahrtverkürzen Strafkosten berücksichtigt, um abzubilden dass Passagiere auf das nächste reguläre Fahrzeug warten oder eventuell zu ihrem Zielhaltepunkt laufen müssen.

Die Autoren formulieren ein gemischt-ganzzahliges Programm, das sie mit Hilfe von Methoden der Mathematischen Programmierung und Constraint Programming lösen. Darüber hinaus kombinieren sie ihre Optimierungsanwendung mit dem agentenbasierten Simulationsframework *MatSim*<sup>20</sup>, das zur Generierung von Echtzeitdaten und zur Validierung verwendet wird.

Die Validierung ihres Verfahrens führen Nesheli und Ceder an einem Teil des Busnetzes von Auckland durch, der drei Buslinien und acht explizit für Umsteigeverbindungen deklarierte Haltepunkte enthält. Sie untersuchen anhand von 14 Szenarien mit unterschiedlichen Taktfrequenzen und Fahrplanabweichungen unterschiedliche Kombinationen der drei möglichen Handlungsstrategien und vergleichen

---

<sup>20</sup> Leider machen Hadas und Ceder keine weiteren Angaben zum konkreten Simulationsmodell.

sie mit dem Fall ohne Kontrolle. Die Ergebnisse belegen, dass die Kombination aller drei Strategien die besten Ergebnisse bzgl. der Gesamtreisezeit und der Zahl synchronisierter Abfahrten erzielt. Allgemein zeigt sich, dass die angewendeten Strategien bei Szenarien mit hoher Taktfrequenz besser geeignet sind als bei Szenarien mit niedriger Taktfrequenz und dass die Gesamtreisezeit, wie zu erwarten, in Szenarien in denen nur geringe Fahrplanabweichungen untersucht werden stärker reduziert werden kann als in Szenarien mit starken Fahrplanabweichungen.

Der Mehrheit der bisher besprochenen Ansätze ist gemein, dass sie Optimallösungen innerhalb rollender Zeithorizonte bestimmen um den Umfang des Optimierungsproblems zu reduzieren, aber auch weil die Entwicklung des Systems nicht verlässlich für einen beliebig langen Zeithorizont vorausprojiziert werden kann. Genau dies kann, wie Kliewer und Suhl in [53] zeigen, ein Nachteil dieser Ansätze sein: Eine Entscheidung kann zwar für das untersuchte Zeitfenster optimal sein, im weiteren Verlauf kann sich jedoch herausstellen, dass eine andere Entscheidung besser geeignet gewesen wäre. Deswegen vergleichen die Autoren ein Online-Optimierungsverfahren für Haltezeiten zur Minimierung unerwarteter Passagierwartezeiten bei Umsteigeverbindungen mit den folgenden neun regelbasierten Haltezeitverfahren:

- *Regular Waiting Time*: Fahrzeuge warten so lange auf ein (verspätetes) Zubringerfahrzeug bis ein vorab spezifizierter Grenzwert erreicht wurde.
- *Never Wait*: Fahrzeuge warten nie auf Zubringerfahrzeuge.
- *Always Wait*: Fahrzeuge warten immer auf Zubringerfahrzeuge.
- *Train Category Based Waiting 1*: Fahrzeuge werden in hierarchische Kategorien eingeteilt (bspw. Fern- und Nahverkehr). Fahrzeuge einer untergeordneten Kategorie warten immer auf Fahrzeuge einer übergeordneten Kategorie, während umgekehrt nicht gewartet wird. Außerdem warten Fahrzeuge der gleichen Kategorie ebenfalls nicht aufeinander.
- *Train Category Based Waiting 2*: Wie Train Category Based Waiting 1, hier warten aber auch Fahrzeuge der gleichen Kategorie aufeinander.
- *Transit Passenger/Passengers*: Es wird genau dann auf ein Zubringerfahrzeug gewartet, falls das Verhältnis zwischen der Zahl der Passagiere im Zubringerfahrzeug die umsteigen wollen zur Gesamtzahl der Passagiere im wartenden Fahrzeug einen spezifizierten Grenzwert überschreitet.
- *Passengers/Passengers*: Wie Transit Passengers/Passengers, nur wird hier die Gesamtzahl aller Passagiere im Zubringerfahrzeug verwendet.
- *Sum of Transit Passengers/Passengers*: Wie Transit Passengers/Passengers, nur wird hier die Summe aller umsteigenden Passagiere in allen Zubringerfahrzeugen betrachtet.
- *Sum of Passengers/Passengers*: Wie Passengers/Passengers, nur dass hier die Summe aller Passagiere in allen Zubringerfahrzeugen verwendet wird.

Zusätzlich vergleichen Kliewer und Suhl die Ergebnisse der Online-Optimierung und der regelbasierten Verfahren mit einer Offline-Version des Optimierungsverfahrens mit der sie eine ex post Analyse durchführen. Den Vergleich führen die Autoren anhand von Netz- und Fahrplandaten des Fernverkehrsnetzes der Deutschen Bahn im Raum Frankfurt aus dem Jahr 2000 durch. Dazu simulieren sie mit einer deterministischen Simulation acht unterschiedliche Szenarien, die sich in der Zahl der Verspätungen und der Zahl der Passagiere unterscheiden. Die Ergebnisse dieses Vergleichs zeigen, dass die Online-Optimierung keine besseren Ergebnisse liefert als regelbasierte Verfahren und dass die Verfahren auf Basis von Passagierzahlen häufig (leicht) bessere Ergebnisse liefern. Aber auch die einfach umzusetzende Regular Waiting Time schneidet häufig gut ab. Abschließend folgern Kliewer und Suhl aus den Ergebnissen ihres Vergleichs, dass die Online-Optimierung nur dann besser geeignet ist als regelbasierte Verfahren, wenn die Zahl der Eingangsverspätungen gering ist, was nicht verwundert, da sich das System unter diesen Bedingungen relativ stabil verhält und die Begrenzung des Zeithorizonts keine signifikanten Auswirkungen hat.

## 7. Ein Simulationsmodell für regelbasiertes Verkehrsmanagement im Öffentlichen Personennahverkehr

### 7.1. Modellerweiterungen

Im Folgenden werden Erweiterungen für das in Kapitel 5 präsentierte Simulationsmodell vorgestellt, die es dazu in die Lage versetzen, ähnlich wie bei den Ansätzen von Borne et al. in [15] oder Fay in [35], selbstständig Entscheidungen zu Verkehrsmanagementstrategien zu treffen. Der Argumentation von Kliewer und Suhl in [53] folgend, werden dabei in der hier vorgestellten ersten Version keine komplexen Optimierungsverfahren zur Entscheidungsfindung eingesetzt, sondern einfache, durch die Nutzerin parametrisierbare Entscheidungsregeln verwendet. Das Basis-Simulationsmodell muss dazu lediglich um eine zusätzliche Klasse und ca. 1.000 Zeilen Quellcode erweitert werden. Wie bereits beim Basis-Simulationsmodell, ist auch hier ein Ziel der Modellierung, die zukünftige Erweiterung um komplexere (optimierungsbasierte) Ansätze zu unterstützen.

#### 7.1.1. Störungen

Der Einsatz von Verkehrsmanagementstrategien hat seinen Ausgangspunkt immer im Auftreten unvorhergesehener Planabweichungen. Das Basis-Simulationsmodell ermöglicht die Abbildung kleiner, unplanmäßiger Störungen bereits durch die Randomisierung von Fahrzeiten und die Abbildung von Bunching-Effekten beim Passagierwechsel. Um auch größere Störungen und umfangreichere Störungsszenarien, wie bspw. ein blockiertes Gleis, abbilden zu können, wird das Modell um eine entsprechende Klasse *Disruption* erweitert. Sie repräsentiert einzelne Planabweichungen und zeichnet sich durch eine Zielentität, einen Eintrittszeitpunkt, eine – nicht zwingend endliche – Dauer und einen auf die Zielentität bzw. einen Teil von ihr wirkenden Effekt aus. Störungen können damit als Spezialfall von Ereignissen verstanden werden und werden im Modell auch als solcher abgebildet (vgl. Abbildung 7.1).

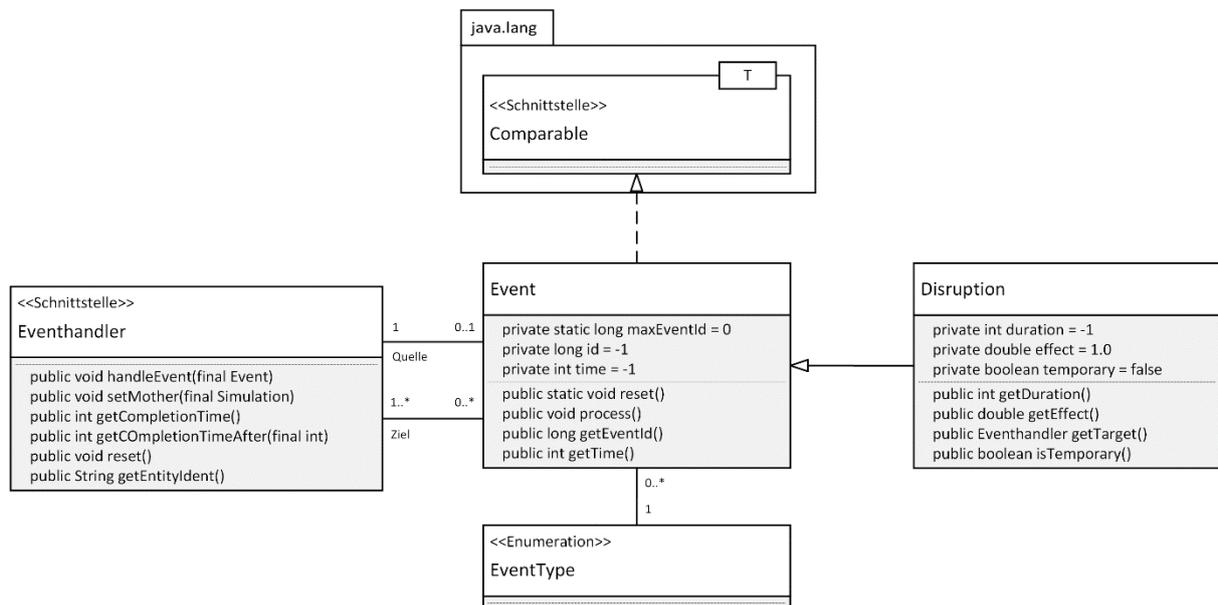


Abbildung 7.1: UML-Klassendiagramm für die Erweiterung des Simulationsmodells um Störungen

Zur Abbildung unterschiedlicher Störungstypen werden die in Tabelle 7.1 aufgelisteten neuen Ereignistypen eingeführt. Sie können danach unterschieden werden, ob sie temporäre oder permanente Störungen abbilden, welchen Teil einer Entität sie betreffen und welche Auswirkungen sie haben.

Die Ereignistypen *TMP/PERM\_DISRUPTED\_TRAVEL\_TIME* bilden Störungen von Fahrzeiten ab und können Verbindungen, Segmente oder Fahrzeuge betreffen. Die Ereignistypen *TMP/PERM\_DISRUPTED\_DWELL\_TIME* bilden hingegen Störungen im Ein-/Ausstiegsprozess ab und können entsprechend nur Haltepunkte oder Fahrzeuge betreffen. Die Effekte, die Störungen der Fahr- bzw. Haltezeiten

Ereignistyp	Beschreibung
<b>TMP_DISRUPTED_TRAVEL_TIME</b>	Temporäre Störung der Fahrzeit
<b>TMP_DISRUPTED_DWELL_TIME</b>	Temporäre Störung der Haltezeit
<b>TMP_BLOCKED</b>	Temporäre Blockade einer Ressource
<b>PERM_DISRUPTED_TRAVEL_TIME</b>	Permanente Störung der Fahrzeit
<b>PERM_DISRUPTED_DWELL_TIME</b>	Permanente Störung der Haltezeit
<b>PERM_BLOCKED</b>	Permanente Blockierung einer Ressource
<b>RESTORE_TRAVEL_TIME</b>	Fahrzeit auf Initialwert zurücksetzen
<b>RESTORE_DWELL_TIME</b>	Haltezeit auf Initialwert zurücksetzen
<b>UNBLOCK</b>	Blockade aufheben

*Tabelle 7.1: Ereignistypen zur Abbildung von Störungen*

haben, werden als relative Veränderung der Fahr-/Haltezeit modelliert und als Gleitkommazahl in der Klasse Disruption gehalten.

Die Ereignistypen *TMP/PERM\_BLOCKED* bilden die temporäre bzw. permanente Blockade einer Netzwerkressource oder eines Fahrzeugs ab. Als Resultat kann die betroffene Entität nicht mehr durch andere Entitäten verwendet werden bzw. den Regelbetrieb nicht mehr aufrechterhalten.

Zur Verarbeitung von Störungen werden die allgemeinen Knoten- und Fahrzeugklassen mit Datenfeldern zum Speichern des sie aktuell betreffenden Störungseffekts sowie einem Blockade-Indikator erweitert. Die konkrete Verarbeitung einer Störung entspricht dann dem Verarbeiten eines neuen Ereignisses, bei dem, je nach Typ, der aktuelle Störungseffekt bzw. Blockade-Indikator der Zielentität überschrieben wird. Die Berechnungen von Fahr- und Haltezeiten werden entsprechend angepasst, indem die Ergebnisse ihrer Basisberechnungen mit dem relativen Störungseffekt multipliziert werden.

Die Aufhebung von temporären Störungen erfolgt mit Hilfe der drei Ereignistypen *RESTORE\_TRAVEL\_TIME*, *RESTORE\_DWELL\_TIME* und *UNBLOCK*. Sie werden von der durch die Störung betroffenen Entität terminiert, sobald die Störung verarbeitet wurde. Bei Eintritt dieser Ereignisse setzt die betroffene Zielentität ihren Störungseffekt bzw. den Blockade-Indikator auf den Basiswert zurück, wodurch die Störung beendet ist.

Die Einbindung von Störungen in den Simulationsverlauf erfolgt als exogene Ereignisse. Sie werden vor dem Beginn des ersten Simulationslaufs aus der gleichen Datenbank gelesen in der auch die Netz- und Fahrplandaten gespeichert sind. Vor dem Start jedes Simulationslaufs werden sie dann in die Future Event List injiziert, um bei ihrem Eintritt von der betroffenen Entität verarbeitet zu werden. Umfangreiche Störungsszenarien werden abgebildet, indem einzelne Störungen in der Datenbank zu Szenarien zusammengefasst werden. Eine Beschreibung des Aufbaus der entsprechenden Tabellen findet sich in Anhang A.

### 7.1.2. Fahrten

In Abschnitt 5.1.3 wurde bereits ausgeführt, dass Fahrten eigene Mengen von Haltepunkt- und Verbindungs-Knoten verwalten, um eine reibungsarme Modifikation der Routen individueller Fahrten zu ermöglichen. Sie werden darüber hinaus um Methoden für die elementare Modifikation ihres Routenverlaufs, wie bspw. das Verkürzen oder Ersetzen (von Teilen) ihres Linienverlaufs, erweitert.

Außerdem werden Fahrten um zwei Indikatoren erweitert, die angeben ob lediglich ihre Route angepasst wurde oder ob auch ihre Abfahrtszeiten, d.h. der Fahrplan, angepasst wurde.

### 7.1.3. Dispatcher

Die Erkennung, Analyse und Adressierung von Planabweichungen fällt in den Aufgabenbereich des Dispatchers, dessen Erweiterung entsprechend umfangreich ausfällt. In den Abschnitten 5.1.2 und 5.1.4.3 wurde bereits erläutert, dass der Dispatcher während des Regelbetriebs durch Fahrzeug-Entitäten vom Ende ihres Passagierwechsels in Kenntnis gesetzt wird und daraufhin Performance-Kennzahlen sowie eventuell notwendige Verkehrsmanagementmaßnahmen bestimmt. Diese Maßnahmen umfassen dabei nicht nur Haltestrategien, sondern auch andere Maßnahmen der Haltepunkt-Kontrolle. So können sich spontan im Simulationsverlauf entwickelnde Störungen durch den Dispatcher bei der Verarbeitung von *BOARDING\_END*-Ereignissen erkannt und adressiert werden. Umfangreiche exogene Störungen werden hingegen erkannt, indem der Dispatcher ihren exogenen Ereignissen als Zielentität hinzugefügt

wird und seine Ereignisverarbeitungsroutine um entsprechende Abschnitte erweitert wird. Es wird also angenommen, dass der Dispatcher unmittelbar bei Eintritt einer Störung in Kenntnis gesetzt wird. Dies gilt allerdings nur für Störungen, die eine Netzwerkressource oder ein Fahrzeug blockieren. Störungen, die Halte- oder Fahrzeiten modifizieren werden hingegen, wie spontan auftretende Störungen, während der Verarbeitung von BOARDING\_END-Ereignissen adressiert.

Zur Umsetzung der einzelnen Verkehrsmanagementmaßnahmen wird der Dispatcher um Methoden erweitert, die die für einzelne Maßnahmen notwendige Logik kapseln, sodass spätere Modifikationen – etwa zur Einbindung von Optimierungsverfahren – nur lokal erfolgen müssen. Eine Beschreibung dieser Methoden erfolgt im folgenden Abschnitt.

## 7.2. Strategien

Die im Dispatcher umgesetzten Strategien lassen sich in zwei Bereiche einteilen: Haltepunkt-Kontrolle und Störfallmanagement. Der Bereich Haltepunkt-Kontrolle umfasst auf einzelne Fahrzeuge abzielende Methoden für fahrplan- und regelbasierte Haltestrategien sowie Verfahren zur begrenzten dynamischen Modifikation von Linienverläufen einzelner Fahrten.

Der Bereich Störfallmanagement beinhaltet hingegen Methoden zur umfassenden Anpassung von Linienverläufen und Fahrplänen ganzer Mengen von Fahrzeugen bei Eintritt einer signifikanten Störung. In der hier besprochenen Version werden dabei nur Blockierungen von Haltepunkten, Verbindungen und Segmenten durch Störfallmanagement adressiert. Blockierungen von Fahrzeugen werden hingegen zunächst vernachlässigt, da die vorhandene Datenbasis keine belastbaren Aussagen zum Fahrzeugpool und dessen Aufteilung auf die unterschiedlichen Depots des Streckennetzes und damit zum (sinnhaften) Einsatz von Ersatzfahrzeugen zulässt.

### 7.2.1. Haltepunkt-Kontrolle

#### 7.2.1.1. Haltestrategien

Der Dispatcher ermöglicht die Anwendung von zwei unterschiedlichen Haltestrategien, die jeweils in zwei unterschiedlichen Varianten zur Verfügung stehen. Während bei der ersten Strategie, mit *Anti-Bunching* bezeichnet, Fahrzeuge, die ihrem Fahrplan voraus sind, an ausgewählten Haltepunkten vorläufig verzögert werden, um Bunching-Effekte zu reduzieren, erfolgt die Verzögerung bei der zweiten Strategie, genannt *Pro-Synchronisierung*, mit dem Ziel Umsteigeverbindungen zu sichern. Beide Haltestrategien werden durch den Dispatcher während der Verarbeitung von Ereignissen des Typs BOARDING\_END implementiert.

Wie in Algorithmus 7.1 im Pseudocode zu sehen, wird bei der ersten Variante von Anti-Bunching die Pünktlichkeit eines Fahrzeugs an seinem aktuellen Haltepunkt (Zeile 04) mit einem durch die Nutzerin zu spezifizierenden Grenzwert  $a_{max}^{Diff}$  verglichen. Wird dieser Grenzwert überschritten, wird das Fahrzeug um die Differenz verzögert, andernfalls wird ihm die sofortige Weiterfahrt gestattet (Zeile 05).

Die zweite Variante der Anti-Bunching Strategie wird nach entsprechender Parametrisierung der Simulation (Zeile 06) durch die Nutzerin aktiv und bezieht, den Überlegungen von Yu und Yang in [106] folgend, neben der Abfahrtszeitabweichung am aktuellen Haltepunkt auch die Pünktlichkeit des Fahrzeugs an seinem nächsten planmäßigen Haltepunkt mit ein. Dazu wird mit Hilfe der Statistikmodule, die die Simulationsdaten erfassen, die bei sofortiger Weiterfahrt zu erwartende Abfahrtszeitabweichung am nächsten Haltepunkt  $\hat{a}_{next}^{Diff}$  berechnet (Zeile 07/08). Lässt dieser Schätzwert darauf schließen, dass das Fahrzeug selbst bei sofortiger Weiterfahrt zu spät am nächsten Haltepunkt ankommt, wird es – unter Missachtung von  $a_{max}^{Diff}$  – nicht verzögert (Zeile 09/10). Ist hingegen zu vermuten, dass das Fahrzeug auch dann am nächsten Haltepunkt zu früh abfahren würde, wird die Wartezeit so bestimmt, dass eine pünktliche Abfahrt am nächsten Haltepunkt gewährleistet ist (Zeile 11 /12). Dies kann dabei zur Folge haben, dass der Grenzwert der Abfahrtszeitabweichung am aktuellen Haltepunkt ebenfalls missachtet wird, wenn dadurch vermieden wird, dass das Fahrzeug wegen einer aufgrund lokaler Informationen erzwungenen Verzögerung zu spät am nächsten Haltepunkt abfahren würde.

**Input:** Fahrzeug  $f$ , Simulationszeitpunkt  $t$ , Maximale Abfahrtszeitabweichung  $a_{max}^{Diff}$

**Output:** Wartezeit  $w_t$

```

01 procedure BestimmeWartezeit( $f$ ,  $t$ )
02 begin
03   Betriebsfahrt  $bf$  := First(Fahrten( $f$ ))
04    $a_{curr}^{Diff}$  := Abfahrtszeit( $bf$ ) -  $t$ 
05    $w_t$  := max{0,  $a_{curr}^{Diff} - a_{max}^{Diff}$ }
06   if lokaleInformation = false and Beendet( $bf$ ) = false and  $a_{curr}^{Diff} > 0$  then
07      $\hat{F}_{next}$  := SchätzeAbfahrtszeit( $f$ ,  $t$ )
08      $\hat{a}_{next}^{Diff}$  := Abfahrtszeit( $bf$ , Haltepunkt( $f$ )+1) -  $\hat{F}_{next}$ 
09     if  $\hat{a}_{next}^{Diff} \leq 0$  then
10        $w_t$  := 0
11     else if  $a_{curr}^{Diff} > a_{max}^{Diff}$  then
12        $w_t$  := min{ $\hat{a}_{next}^{Diff}$ ,  $a_{curr}^{Diff} - a_{max}^{Diff}$ }
13     fi
14   fi
15 end

```

*Algorithmus 7.1: Pseudocode zur Bestimmung zusätzlicher Wartezeit*

Die Pro-Synchronisierung Strategie folgt den Überlegungen von Hadas und Ceder in [42] bzw. Nesheli und Ceder in [70] zur Sicherung von Umsteigeverbindungen bei (stark) synchronen Fahrplänen und unterscheidet Umsteigeverbindungen dazu weiter in Rendezvousverbindungen und gerichtete Umsteigeverbindungen. *Rendezvousverbindungen* sind Umsteigeverbindungen, bei denen sich in einem spezifizierten Zeitfenster Fahrzeuge der unterschiedlichen Linien einer Haltestelle an dieser sammeln, um ihren Passagieren den Umstieg in die jeweils anderen Fahrzeuge zu erlauben. Zu diesem Zweck warten die betroffenen Fahrzeuge an ihrem jeweiligen Haltepunkt und fahren erst ab, nachdem alle betroffenen Fahrzeuge an der Haltestelle angekommen sind und eine Pufferzeit zur Ermöglichung des eigentlichen Umsteigevorgangs verstrichen ist. *Gerichtete Umsteigeverbindungen* sind eine Verallgemeinerung von Rendezvousverbindungen die sich, wie die Umsteigevorgaben aus Kapitel 3, auf genau zwei Linien und zwei Haltepunkte einer Haltestelle beziehen.

Sowohl Rendezvousverbindungen als auch gerichtete Umsteigeverbindungen müssen vorab durch die Nutzerin spezifiziert und in der dem Simulationsmodell zugrundeliegenden Datenbank gespeichert werden.<sup>21</sup>

Algorithmus 7.2 zeigt in Pseudocode am Beispiel von Rendezvousverbindungen und leicht verkürzt, die verwendete Entscheidungslogik. Zunächst wird überprüft ob eine Rendezvousverbindung für die aktuelle Haltestelle des Fahrzeugs definiert wurde und ob es sich bei dieser Haltestelle nicht um die letzte Haltestelle der Fahrt des Fahrzeugs handelt (Zeile 05). Da an Endhaltepunkten kein Zustieg von Passagieren gestattet ist, würde die Inklusion von Fahrzeugen die sich am Ende ihrer Route befinden nur zu ihrer unnötigen Verzögerung führen.

Wurde eine Rendezvousverbindung definiert, können drei Fälle unterschieden werden:

1. *Die Rendezvousverbindung ist erfüllt (Zeile 06-08):* In diesem Fall muss der Dispatcher dem betroffenen Fahrzeug lediglich den Zeitpunkt mitteilen an dem es seine Fahrt fortsetzen darf. Dabei muss es mindestens die durch die Nutzerin zu definierende minimale Umsteigezeit  $\tau_t$  einhalten, darf aber nicht vor der im Fahrplan angegeben Abfahrtszeit losfahren.
2. *Die Rendezvousverbindung ist noch nicht erfüllt und die maximale Wartezeit wurde erreicht (Zeile 09/10):* Neben der minimalen Umsteigezeit muss durch die Nutzerin auch die maximale Wartezeit  $w_{t_c}$  definiert werden. Sie gibt die Dauer an, die ein Fahrzeug über seine geplante Abfahrtszeit hinweg auf Zubringerfahrzeuge wartet. Ist diese Zeit überschritten, wird die Rendezvousverbindung nicht gehalten und dem Fahrzeug wird die Weiterfahrt gestattet. Dieser Ansatz entspricht der durch Kliewer und Suhl in [53] definierten *Regular Waiting Time*.

<sup>21</sup> Für eine Beschreibung der entsprechenden Tabellen siehe Anhang A.

**Input:** Fahrzeug  $f$ , Simulationszeitpunkt  $t$ , Minimale Umsteigezeit  $t_t$ , Maximale Wartezeit  $wt_c$ ,  
Abfrageintervall  $t_q$

**Output:** Anweisung an das betroffene Fahrzeug in Form eines Ereignisses

```

01 procedure Rendezvousverbindung( $f$ ,  $t$ )
02 begin
03   Haltestelle  $hs$  := Haltestelle(Haltepunkt( $f$ ))
04   Betriebsfahrt  $bf$  := First(Fahrten( $f$ ))
05   if RendezvousVerbindung( $hs$ ,  $t$ ) = true and Beendet( $bf$ ) = false then
06     if RendezvousVerbindungErfüllt( $hs$ ,  $t$ ) = true then
07        $t_{e+1}$  := max{ $t + t_t$ , Abfahrtszeit( $bf$ )}
08       /* Teile Fahrzeug per Ereignis mit, dass es um  $t_{e+1}$  losfahren darf */
09     else if  $t \geq$  Abfahrtszeit( $bf$ ) +  $wt_c$  then
10       /* Teile Fahrzeug mit, dass es sofort losfahren darf. */
11     else /* Kriterien für Rendezvousverbindung noch nicht erfüllt */
12        $\hat{a}_{last}$  := SchätzeAnkunftszeit( $hs$ )
13       if  $\hat{a}_{last} < 0$  then
14         if  $t + t_q >$  Abfahrtszeit( $bf$ ) +  $wt_c$  then
15           /* Warte bis Abfahrtszeit( $bf$ ) +  $wt_c$  und teste erneut */
16         else
17           /* Warte bis  $t + t_q$  und teste erneut */
18         fi
19       else if  $\hat{a}_{last} <$  Abfahrtszeit( $bf$ ) +  $wt_c$  then
20         /* Warte bis das letzte Fahrzeug ankommt und teste erneut */
21       else
22         if Fahrzeuge( $hs$ ) < Fahrzeuge( $hs$ , Abfahrtszeit( $bf$ )+ $wt_c$ ) then
23           /* Warte bis Abfahrtszeit( $bf$ ) +  $wt_c$  und teste erneut */
24         else
25            $t_{e+1}$  := max{ $t$ , Abfahrtszeit( $bf$ )}
26           if Fahrzeuge( $hs$ ) > 1 then
27              $t_{e+1}$  := max{ $t_{e+1}$ , Ankunftszeit( $f$ ) +  $t_t$ }
28           fi
29           /* Teile Fahrzeug mit dass es zum Zeitpunkt  $t_{e+1}$  losfahren darf */
30         fi
31       fi
32     fi
33 end

```

**Algorithmus 7.2: Pseudocode der Entscheidungslogik bei Rendezvousverbindungen**

3. Die Rendezvousverbindung ist noch nicht erfüllt und die maximale Wartezeit wurde noch nicht erreicht (Zeile 11-32): In diesem Fall wird zunächst abgeschätzt zu welchem Simulationszeitpunkt die Rendezvousverbindung erfüllt sein wird, d.h. wann das letzte betroffene Fahrzeug die Haltestelle erreichen wird (Zeile 12). Abhängig von dieser geschätzten Ankunftszeit sind wiederum drei Fälle zu unterscheiden:
  - i. Auf den Zubringerverbindungen der unbelegten Haltepunkte befinden sich keine Fahrzeuge (Zeile 13-18): Da sich die noch fehlenden Fahrzeuge zu weit entfernt befinden, kann keine Ankunftszeit abgeschätzt werden. In diesem Fall wird  $t_q$  Zeitschritte gewartet und erneut versucht die Ankunftszeiten abzuschätzen. Dieses Verfahren wird wiederholt bis die maximale Wartezeit erreicht ist oder Ankunftszeiten abgeschätzt werden können.
  - ii. Das letzte Fahrzeug wird vor Erreichen der maximalen Wartezeit ankommen (Zeile 19/20): In diesem Fall wird dem bereits wartenden Fahrzeug mitgeteilt, dass es so lange warten soll, bis das letzte Fahrzeug eingetroffen ist.
  - iii. Das letzte Fahrzeug wird nicht vor Erreichen der maximalen Wartezeit eintreffen (Zeile 21-31): In diesem Fall versucht der Dispatcher die Rendezvousverbindung mit so vielen Fahrzeugen wie möglich zu halten. Dazu wägt er ab, ob sich bis zur Erreichung der maximalen Wartezeit noch weitere Fahrzeuge an der Haltestelle einfinden werden oder nicht. Ist dies der Fall, wird dem bereits wartenden Fahrzeug mitgeteilt dass es weiter warten soll (Zeile 22/23). Andernfalls wird die Rendezvousverbindung mit den vorhandenen Fahrzeugen durchgeführt, falls sich mehr als ein Fahrzeug an

der Haltestelle befindet. Befindet sich lediglich ein einzelnes Fahrzeug an der Haltestelle, wird es angewiesen seine Fahrt sofort fortzusetzen sobald seine geplante Abfahrtszeit erreicht wurde (Zeilen 24-30).

Die Entscheidungslogik für gerichtete Umsteigeverbindungen entspricht weitestgehend der Logik für Rendezvousverbindungen. Bei ihr entfällt allerdings die Abwägung bezüglich der Zahl der beteiligten Fahrzeuge (Zeilen 22-24 und 26-28 in Algorithmus 7.2). Sind an einer Haltestelle mehrere gerichtete Umsteigeverbindungen mit gleicher Ziellinie definiert, d.h. soll ein Fahrzeug auf mehrere Zubringerfahrzeuge warten, wird während des Wartevorgangs ein von der zugehörigen Haltestelle verwaltetes Verzeichnis bisher (noch nicht) erfüllter Umsteigeverbindungen aktualisiert. Dieses Verzeichnis wird dann dazu herangezogen, die Erfüllung der gerichteten Umsteigeverbindungen zu prüfen, womit – anders als bei den Rendezvousverbindungen – nicht alle Zubringerfahrzeuge zeitgleich an der Haltestelle sein müssen, um die Verbindung zu erfüllen.

Anti-Bunching und Pro-Synchronisierung können innerhalb eines Verkehrssystems in Kombination miteinander verwendet werden, stehen an Kontrollpunkten an denen beide eingesetzt werden können aber in einer hierarchischen Beziehung. An solchen Punkten überschreibt die Pro-Synchronisierung Strategie, die Anti-Bunching Strategie. Darüber hinaus besteht zwischen den beiden Varianten der Pro-Synchronisierung Strategie ebenfalls eine hierarchische Beziehung: Ist an einer Haltestelle sowohl eine Rendezvousverbindung als auch eine gerichtete Umsteigeverbindung definiert, wird letztere durch erstere überschrieben.

#### 7.2.1.2. Dynamische Modifikation des Linienvverlaufs

Mit den vier Haltestrategievarianten ist der Dispatcher in der Lage die Frühzeitigkeit von Fahrzeugen – und damit einhergehende Bunching-Effekte – sowie die Sicherung von Umsteigeverbindung zu kontrollieren. Er kann allerdings keine Störungen adressieren, deren Effekte so groß sind, dass sie nicht durch die Struktur des Fahrplans aufgefangen werden können und zeitgleich zu gering sind, um ein umfangreiches Störfallmanagement zu rechtfertigen.

Deswegen wird der Dispatcher um eine Methode erweitert, die den Linienvverlauf einzelner Fahrten, wenn möglich, in begrenztem Maß verändern kann. Der Einsatz dieser Methode wird dabei, wenn von der Nutzerin per Parameter aktiviert, wie die Haltestrategien jedes Mal in Erwägung gezogen, nachdem eine Fahrzeug-Entität den Passagierwechsel beendet hat und dies mittels eines BOARDING\_END-Ereignisses an den Dispatcher meldet. Algorithmus 7.3 zeigt den Pseudocode der Methode.

Sie bestimmt zunächst an welcher Position im Linienvverlauf seiner Fahrt sich ein Fahrzeug befindet (Zeile 05-07). Anschließend wird, wie bei der Anti-Bunching Strategie, die Pünktlichkeit des Fahrzeugs an seinem nächsten planmäßigen Haltepunkt geschätzt (Zeile 08/09). Überschreitet die Abweichung den durch die Nutzerin festzulegenden Grenzwert  $ar_{max}^{Diff}$ , wird ein Umleiten (engl. *rerouting*) des Fahrzeugs erwogen (Zeile 10). Kommt ein Umleiten in Betracht, wird die ursprüngliche Route des Fahrzeugs zwischen seinem aktuellen und nächsten Haltepunkt als Initiallösung definiert (Zeile 11) und eine Menge potentieller, unblockierter Alternativrouten bestimmt (Zeilen 12/13). Diese Menge kann prinzipiell mit einem beliebigen Wegfinde-Algorithmus bestimmt werden. Um eine möglichst geringe Kopplung zwischen den Statistikmodulen und dem das Netzwerk repräsentierenden Graphen zu erhalten, wird in der aktuellen Version ein einfacher Breitensuche-Algorithmus verwendet, der ohne weitere (Kosten-)Informationen alle einfachen Wege zwischen zwei Knoten des Graphen bestimmt und dessen maximale Tiefe durch die Nutzerin mittels  $wl_{max}$  beschränkt wird.<sup>22</sup> Potentielle Alternativrouten müssen dabei keine direkte Verbindung zwischen dem aktuellen und nächsten Haltepunkt des Fahrzeugs darstellen, sondern können auch über zusätzliche Haltepunkte führen.

Sind die Alternativrouten bestimmt, wird, je nach Parametereinstellung durch die Nutzerin, entweder die schnellste oder die pünktlichste dieser Routen bestimmt (Zeilen 14-22). Sind zwei Routen gleich

---

<sup>22</sup> Sollen ganze Straßennetze, inklusive Individualverkehr, simuliert werden, würde sich ein Kürzeste-Wege-Verfahren mit multiplen Kanten-/Knotenlabeln zur Berechnung der pareto-optimalen Lösungen eher eignen. Für die hier betrachteten ÖPNV-Netze mit ihrer relativ geringen Zahl an Verzweigungen, reicht die gewählte Breitensuche-Implementierung jedoch zunächst aus.

schnell bzw. gleich pünktlich, wird die Alternative gewählt, die am wenigsten vom Verlauf der Originalroute abweicht, d.h. die geringere Zahl an zwischenliegenden Haltepunkten aufweist. Wurde eine Route gefunden, die pünktlicher/schneller als die Originalroute ist, wird der entsprechende Abschnitt im Linienverlauf der Fahrt durch diese neue Route ersetzt (Zeilen 23-25).

Eine Besonderheit des Verfahrens ist, dass es – die entsprechende Einstellung durch die Nutzerin vorausgesetzt – die Modifikation des Linienverlaufs auch ermöglicht wenn erwartet wird, dass ein Fahrzeug zu früh an seinem nächsten Haltepunkt abfahren wird. In diesem Fall kann eine pünktlichere, d.h. langsamere Route, verwendet werden. Dies kann bspw. sinnvoll sein, wenn ein Fahrzeug weder am aktuellen noch am nächsten Haltepunkt verzögert werden kann. Dieser Ansatz ist vergleichbar mit der Anpassung von Fahrzeiten im Modell von Hadas und Ceder in [42]. Dort werden Fahrzeiten angepasst, indem die auf einer Verbindung zu fahrende Geschwindigkeit reduziert bzw. erhöht wird. Während ersteres in einem isolierten Bahnsystem noch ohne größere Komplikationen umgesetzt werden kann, wird diese Herangehensweise in einem Bussystem mit hoher Wahrscheinlichkeit zu Missmut unter den Teilnehmern des Individualverkehrs führen, sofern keine dedizierten Busspuren verwendet werden. Sollen Fahrzeuge zwischen zwei Haltepunkten verzögert werden, erscheint es daher sinnvoller dies, wenn möglich, über geringfügige Umleitungen zu erzielen.

Konnte keine bessere Route gefunden werden, wird als letzte Maßnahme, sofern durch die Nutzerin erlaubt, noch eine Verkürzung der Fahrt in Betracht gezogen (Zeile 26-28). Das dazu verwendete Verfahren zeigt Algorithmus 7.4.

---

**Input:** Fahrzeug  $f$ , Simulationszeitpunkt  $t$ , Rerouting-Grenzwert  $ar_{max}^{Diff}$ ,  
Maximale Weglänge  $wl_{max}$

**Output:** Modifikations-Indikator  $mod$ , Modifizierter Linienverlauf

```

01 procedure ModifiziereLinienverlauf( $f$ ,  $t$ )
02 begin
03    $mod := false$ 
04   Betriebsfahr  $bf := First(Fahrten(f))$ 
05   if Beendet( $bf$ ) = true then
06     /* Das Fahrzeug befindet sich am letzten Haltepunkt seiner Fahrt */
07   else
08      $\hat{F}_{next} := SchätzeAbfahrtszeit(f, t)$ 
09      $\hat{a}_{next}^{Diff} := Abfahrtszeit(bf, Haltepunkt(f)+1) - \hat{F}_{next}$ 
10     if  $|\hat{a}_{next}^{Diff}| > ar_{max}^{Diff}$  then
11        $r_{opt} := Route(bf, Haltepunkt(f), Haltepunkt(f)+1)$ 
12        $R := AllSimplePaths(Haltepunkt(f), Haltepunkt(f)+1, wl_{max})$ 
13        $R := R \setminus \{r \in R \mid r = r_{opt} \vee Blockiert(r) = true\}$ 
14       if  $R \neq \emptyset$  then
15         BestimmeFahrzeiten( $R$ )
16         if  $\hat{a}_{next}^{Diff} < 0$  then
17           /* Bestimme pünktlichste/schnellste Route aus  $R$ ; speichere sie in
18              $r_{opt}$  */
19         else if modifiziereWennZuFrüh = true then
20           /* Bestimme pünktlichste Route aus  $R$ ; speichere sie in  $r_{opt}$  */
21         fi
22       fi
23       if  $r_{opt} \neq Route(bf)$  then
24         Replace( $bf$ , Haltepunkt( $f$ ), Haltepunkt( $f$ )+1,  $r_{opt}$ )
25          $mod := true$ 
26       else if verkürzenErlaubt = true then
27          $mod := Verkürzen(f, t)$ 
28       fi
29     fi
30   fi
31 end

```

---

Algorithmus 7.3: Pseudocode zur Modifikation des Linienverlaufs aufgrund größerer Störungen

**Input:** Fahrzeug  $f$ , Simulationszeitpunkt  $t$ , Grenzwert der Abfahrtszeitabweichung  $ast_{min}^{Diff}$ ,  
Mindestrountenanteil  $ra_{min}$

**Output:** Modifikations-Indikator  $mod$ , Modifizierter Linienverlauf

```

01 procedure Verkürzen( $f$ ,  $t$ )
02 begin
03    $mod := false$ 
04   Betriebsfahrt  $bf := First(Fahrten(f))$ 
05   if Beendet( $bf$ ) = true then
06     /* Fahrzeug befindet sich am letzten Haltepunkt seiner Fahrt */
07   else
08      $\dot{F}_{next} := SchätzeAbfahrtszeit(f, t)$ 
09      $\dot{a}_{next}^{Diff} := Abfahrtszeit(bf, Haltepunkt(f)+1) - \dot{F}_{next}$ 
10     if  $\dot{a}_{next}^{Diff} < 0$  and  $|\dot{a}_{next}^{Diff}| > ast_{min}^{Diff}$  and Wenden(Haltestelle( $f$ )) = true then
11       if  $|Linien(Haltepunkt(f)) \cap Linien(Haltepunkt(f)+1)| > 1$  and
12         Position( $bf$ , Haltepunkt( $f$ )) /  $|Haltepunkte(bf)| \geq ra_{min}$  then
13         CutOff( $bf$ , Haltepunkt( $f$ ))
14         if  $\overline{bf} \neq 0$  then /* Fahrzeug besitzt eine Folgefahrt */
15            $r := Route(\overline{bf})$ 
16           CutOff( $\overline{bf}$ , Haltestelle(Haltepunkt( $f$ )))
17           Startzeit( $\overline{bf}$ ) := Startzeit( $\overline{bf}$ ) +
18                                 Fahrzeit( $r$ , Haltestelle(Haltepunkt( $f$ )))
19         fi
20        $mod := true$ 
21     fi
22   fi
23 fi
24 end

```

#### Algorithmus 7.4: Pseudocode zur Verkürzung von Fahrten aufgrund von Verspätungen

Eine Fahrt wird nur verkürzt, wenn die Verspätung eines Fahrzeugs hinreichend signifikant ist, d.h. den durch die Nutzerin zu spezifizierenden Grenzwert  $ast_{min}^{Diff}$  überschreitet, und an der aktuellen Fahrzeugposition die Möglichkeit zum Wenden besteht (Zeile 10). Zusätzlich muss für Passagiere die Möglichkeit bestehen mit einer anderen Linie zumindest den nächsten planmäßigen Haltepunkt zu erreichen (Zeile 11). Dies dient zum einen als (schwacher) Indikator dafür, dass mindestens eine weitere Linie einen Teil des restlichen Linienverlaufs abdeckt und zum anderen wird so vermieden, dass Passagiere kurz vor dem Endhaltepunkt einer Fahrt stranden. Zuletzt muss noch sichergestellt sein, dass die zu verkürzende Fahrt einen durch die Nutzerin zu spezifizierenden Mindestanteil  $ra_{min}$  ihres ursprünglichen Linienverlaufs zurückgelegt hat (Zeile 12), damit ein Mindestservicelevel gewährleistet bleibt.

Wird die aktuelle Fahrt eines Fahrzeugs verkürzt, hat dies unmittelbare Auswirkungen auf die nächste Fahrt des Fahrzeugs. Entweder muss das Fahrzeug eine Leerfahrt bis zum Endhaltepunkt seiner aktuellen Fahrt durchführen um die Route der nächsten Fahrt beizubehalten oder die nächste Fahrt des Fahrzeugs muss ebenfalls verkürzt werden und an der aktuellen Position des Fahrzeugs starten. Ersteres kann erfolgsversprechend sein, wenn die Verspätung des Fahrzeugs gering genug ist, um durch den Ausfall der Passagierwechsel an den restlichen Haltepunkten der Route kompensiert werden zu können, ist den von den Ausfällen betroffenen Passagieren aber nur schwerlich zu erklären. Aus diesem Grund verwendet der Dispatcher die zweite Variante. Wird eine Fahrt verkürzt und muss das ausführende Fahrzeug anschließend eine weitere Fahrt  $\overline{bf}$  durchführen (Zeile 14), wird auch diese Fahrt verkürzt. Um den ursprünglichen Fahrplan aufrecht zu halten, wird die Startzeit von  $\overline{bf}$  auf ihre planmäßige Abfahrtszeit am entsprechenden Haltepunkt gesetzt (Zeilen 17/18), d.h. das betroffene Fahrzeug verweilt bis dahin an der aktuellen Haltestelle.

Ein potentieller Nachteil von Algorithmus 7.4 ist, dass er nicht zwingend sicherstellt, dass betroffene Passagiere die Möglichkeit haben die restlichen Haltepunkte einer Route zu erreichen. Eine, nicht implementierte, allgemeine Variante der Methode, die dies sicherstellt, wird in Algorithmus 7.5 beschrieben. Er garantiert, dass bei einer Verkürzung mindestens ein – eventuell über mehrere Linien führender – alternativer Weg durch das Streckennetz zu den Folgehaltestellen der verkürzten Fahrt existiert. Das Verfahren arbeitet dabei auf dem Level von Haltestellen anstatt von Haltepunkten, da angenommen

werden kann, dass es für Passagiere im Allgemeinen nicht relevant ist an welchem Haltepunkt einer Haltestelle sie ankommen, so lange sie ihre Zielhaltestelle erreichen. Zusätzlich ermöglicht dies auch die Berücksichtigung komplexere Alternativwege.

Die Unterschiede in den Algorithmen finden sich in Zeilen 11 bis 22. Zunächst wird die Vereinigungsmenge aller Linien (außer der zu verkürzenden Linie) bestimmt, die die Haltestellen der zu verkürzenden Fahrt bedienen (Zeile 11/12). Anschließend wird die Vereinigungsmenge aller Haltestellen dieser Linien  $H_L^+$ , sowie die Menge aller Folgehaltestellen der zur verkürzenden Fahrt  $H_{bf}$  bestimmt (Zeile 13/14). Mit diesen Mengen können zwei notwendige Bedingungen überprüft werden (Zeile 15):

1. Die Haltestelle an der die Fahrtverkürzung durchgeführt werden soll ist Teil von  $H_L^+$ , d.h. mindestens eine weitere Linie bedient diese Haltestelle.
2. Es befindet sich ein durch die Nutzerin zu definierender Mindestanteil der Folgehaltestellen in  $H_L^+$ , d.h. durch die Linien, die die Haltestellen in  $H_L^+$  bedienen, werden mindestens auch  $hsa_{min}$  Prozent der Folgehaltestellen der zu verkürzenden Fahrt bedient.

Mit diesen beiden Bedingungen wird dabei nur sichergestellt, dass die relevanten Haltestellen durch weitere Linien bedient werden, nicht aber dass Passagiere, unter Verwendung dieser Linien, auch die Möglichkeit haben von der Haltestelle an der die Verkürzung stattfindet zu den Folgehaltestellen zu gelangen. Abbildung 7.2 zeigt dazu ein einfaches Beispiel mit drei Haltestellen und vier Linien: Würde

---

**Input:** Fahrzeug  $f$ , Simulationszeitpunkt  $t$ , Grenzwert der Abfahrtszeitabweichung  $ast_{min}^{Diff}$ ,  
 Mindestrountenanteil  $ra_{min}$ , Mindesthaltestellenanteil  $hsa_{min}$   
**Output:** Modifikations-Indikator  $mod$ , Modifizierter Linienverlauf

```

01 procedure Verkürzen( $f$ ,  $t$ )
02 begin
03    $mod := false$ 
04   Betriebsfahrt  $bf := First(Fahrten(f))$ 
05   if Beendet( $bf$ ) = true then
06     /* Fahrzeug befindet sich am letzten Haltepunkt seiner Fahrt */
07   else
08      $\hat{F}_{next} := SchätzeAbfahrtszeit(f, t)$ 
09      $\hat{a}_{next}^{Diff} := Abfahrtszeit(bf, Haltepunkt(f)+1) - \hat{F}_{next}$ 
10     if  $\hat{a}_{next}^{Diff} < 0$  and  $|\hat{a}_{next}^{Diff}| > ast_{min}^{Diff}$  and Wenden(Haltestelle( $f$ )) = true then
11        $L := \bigcup_{hs \in Haltestellen(bf)} Linien(hs)$ 
12        $L := L \setminus Linie(bf)$ 
13        $H_L^+ := \bigcup_{l \in L} Haltestellen(l)$ 
14        $H_{bf} := \{hs \in Haltestellen(bf) \mid hs > Haltestelle(f)\}$ 
15       if Haltestelle( $f$ )  $\in H_L^+$  and  $|H_L^+ \cap H_{bf}| / |H_{bf}| \geq hsa_{min}$  then
16         MLKGHS := Multi-Linienkonflikt-GraphHS()
17         MLKGHS := MLKGHS \ Linie( $bf$ )
18          $R := \emptyset$ 
19         for alle  $hs \in H_L^+ \cap H_{bf}$  do
20            $R := R \cup Path(MLKG_{HS}, Linien(Haltestelle(f)), Linien(hs))$ 
21         od
22         if  $|R| \geq |H_L^+ \cap H_{bf}|$  then
23           CutOff( $bf$ , Haltepunkt( $f$ ))
24           if  $\overline{bf} \neq 0$  then
25              $r := Route(\overline{bf})$ 
26             CutOff( $\overline{bf}$ , Haltestelle(Haltepunkt( $f$ )))
27             Startzeit( $\overline{bf}$ ) := Startzeit( $\overline{bf}$ ) +
28                                   Fahrzeit( $r$ , Haltestelle(Haltepunkt( $f$ )))
29           fi
30            $mod := true$ 
31         fi
32       fi
33     fi
34   fi
35 end

```

---

**Algorithmus 7.5:** Pseudocode für eine allgemeine Variante zur Verkürzung von Fahrten, die sicherstellt dass Passagiere einen Mindestanteil der restlichen Haltepunkte erreichen

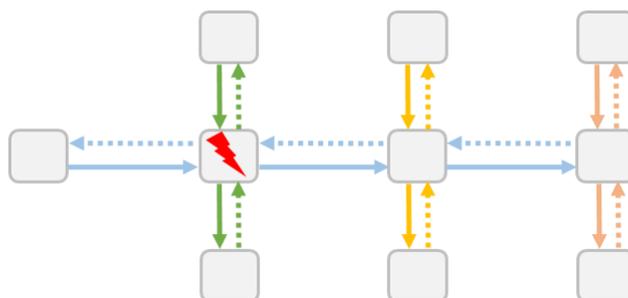


Abbildung 7.2: Beispiel zur Fahrverkürzung, bei dem die notwendigen, aber nicht die hinreichenden Bedingungen erfüllt sind

die horizontal fahrende Linie an der ersten Haltestelle verkürzt, wäre gesichert, dass sowohl die Haltestelle an der verkürzt wird als auch alle Folgehaltestellen durch Alternativlinien bedient würden. Es wäre aber nicht gesichert, dass Passagiere die Folgehaltestellen mit Hilfe der Alternativlinien erreichen können.

Um sicherzustellen, dass Passagiere die Folgehaltestellen erreichen können, wird ein Multi-Linienkonflikt-Graph für das Gesamtnetzwerk erstellt (Zeile 16). Dieser Multi-Linienkonflikt-Graph unterscheidet sich von dem in Kapitel 2.2.1 vorgestelltem, insofern als dass ein Linienkonflikt – und damit eine Kante – bereits existiert, wenn zwei Linien die gleiche Haltestelle anfahren anstatt den gleichen Haltepunkt. Aus diesem Graphen wird der Knoten der zu verkürzenden Linie samt inzidenter Kanten entfernt (Zeile 17), da Alternativwege diese Linie nicht verwenden dürfen. In dem so erzeugten Multi-Linienkonflikt-Graphen müssen anschließend Wege gefunden werden, die die Linien – d.h. Knoten – die die Haltestelle bedienen an denen die Fahrt verkürzt wird, mit den Linien verbindet, die die Folgehaltestellen bedienen (Zeile 18-21). Zu beachten ist hierbei, dass ein Weg bereits existiert, wenn Start- und Zielknoten – d.h. Linie – identisch sind, da dies bedeutet, dass mehrere Haltepunkte von der gleichen Alternativlinie bedient werden. Enthält die so bestimmte Menge mindestens so viele Elemente wie durch Alternativlinien bediente Folgehaltestellen ist sichergestellt, dass Passagiere eine Möglichkeit haben diese Haltestellen, eventuell unter Verwendung mehrerer Linien, bei einer Fahrverkürzung zu erreichen.

### 7.2.2. Störfallmanagement

Die Kombination der bisher vorgestellten Verfahren erlaubt dem Dispatcher die Reaktion auf einen Großteil spontan auftretender Störungen, die den Betrieb nicht zu stark beeinflussen. Störungen nach denen der Regelbetrieb nicht mehr aufrechterhalten werden kann, wie bspw. ein Oberleitungsschaden, können mit diesen Herangehensweisen jedoch nicht adressiert werden. Sie erfordern umfangreichere Maßnahmen, die Veränderungen an mehreren Fahrten, inklusive Fahrplananpassungen, umfassen.

Um auch auf solche Störungen reagieren zu können wurde der Dispatcher um vier weitere Verfahren erweitert: Umleiten, Verkürzen, Überspringen und Auftrennen. Sie alle verwenden, wie bereits die Methoden der Haltepunkt-Kontrolle, nur die durch das Streckennetz und den Fahrplan zur Verfügung stehenden Informationen, nicht aber weiteres Expertenwissen. Wie Algorithmus 7.6 beispielhaft für eine blockierte Verbindung zeigt, werden die Methoden bei Eintritt einer entsprechenden Störung im Rahmen der zugehörigen Ereignisverarbeitung des Dispatchers aktiviert und stehen in einer hierarchischen Beziehung zueinander. Zunächst wird versucht eine allgemeingültige Umleitung, d.h. eine die ursprünglichen Haltepunkte weiterhin bedienende Alternativroute, zu finden. Ist dies nicht erfolgreich werden Einzellösungen für alle durch die Störung betroffenen Linien gesucht. Dabei wird zuerst geprüft ob eine Verkürzung – ähnlich wie bei der dynamischen Modifikation des Linienverlaufs – zweckmäßig ist. Ist dies nicht der Fall, wird versucht die gestörte Netzwerkressource zu überspringen bevor als letzte Möglichkeit eine Auftrennung der Linie in Erwägung gezogen wird.

Abhängig von der durch die Störung betroffenen Komponente unterscheidet sich die Ereignisverarbeitung leicht. Ist beispielsweise ein Haltepunkt blockiert, d.h. kann nicht angefahren werden, kommt eine Umleitung nicht in Betracht. Ist hingegen ein einzelnes Segment von einer Störung betroffen, muss

---

**Input:** Störung  $d$

**Output:** Modifizierte Linienverläufe für alle betroffenen Fahrten

```

01 procedure handleEvent(d)
02 begin
03   /* Nehmen an: Störung betrifft eine Verbindung */
04
05   if Umleiten(d, Ziel(d), Zeit(d)) = false then
06     L := Linien(Ziel(d))
07     for alle  $l \in L$  do
08       if Verkürzen( $l$ ,  $d$ , Ziel( $d$ ), Zeit( $d$ )) = false then
09         if Überspringen( $l$ ,  $d$ , Ziel( $d$ ), Zeit( $d$ )) = false then
10           Auftrennen( $l$ ,  $d$ , Ziel( $d$ ), Zeit( $d$ ))
11         fi
12       fi
13     od
14   fi
15 end

```

---

*Algorithmus 7.6: Pseudocode Ausschnitt der Ereignisverarbeitung bei einer blockierten Verbindung*

für jede dieses Segment enthaltende Verbindung eine Überprüfung der einzelnen Lösungsmöglichkeiten durchgeführt werden.

Die umgesetzte Entscheidungslogik der einzelnen Methoden und ihre Unterschiede werden im Folgenden anhand einer blockierten Verbindung kurz beschrieben.

### 7.2.2.1. Umleiten

Ist eine Verbindung oder ein Segment blockiert, versucht der Dispatcher zunächst einen alternativen Weg durch das Netzwerk zu finden, bei dem alle Haltepunkte der ursprünglichen Route auch weiterhin bedient werden. Das dazu verwendete Verfahren ist in Algorithmus 7.7 dargestellt und ähnelt sehr stark dem von Algorithmus 7.3 zur dynamischen Modifikation des Linienverlaufs.

---

**Input:** Störung  $d$ , Verbindung  $c$ , Simulationszeitpunkt  $t$ , Maximale Weglänge  $wl_{max}$ ,

Maximale Fahrzeitabweichung  $Fr_{max}^{Diff}$ , Pufferzeit  $t_d$

**Output:** Lösungs-Indikator found, Modifizierte Linienverläufe für alle betroffenen Fahrten

```

01 procedure Umleiten(d, c, t)
02 begin
03   found := false
04   L := Linien(c)
05   R := AllSimplePaths(Start(c), Ziel(c),  $wl_{max}$ )
06   R := R \ { $r \in R \mid c \in r \vee \text{Blockiert}(r) = \text{true}$ }
07   if  $R \neq \emptyset$  then
08      $r_{opt} = \text{First}(R)$ 
09      $\hat{F}_{next}^{Diff} := |\text{Fahrzeit}(c) - \text{Fahrzeit}(r_{opt})|$ 
10     /* Bestimme pünktlichste/schnellste Route aus R; speichere sie in  $r_{opt}$  und
11       ihre Fahrzeitdifferenz in  $\hat{F}_{next}^{Diff}$  */
12
13     if  $\hat{F}_{next}^{Diff} \leq Fr_{max}^{Diff}$  then
14       found := true
15       for alle  $l \in L$  do
16          $BF_l := \{bf \in \text{Fahrten}(l) \mid \text{Beendet}(bf) = \text{false} \wedge$ 
17           Position( $bf$ )  $\leq$  Position( $bf$ , Start( $c$ ))  $\wedge$ 
18           Startzeit( $bf$ ) + Fahrzeit( $bf$ , Start( $c$ ))  $\leq t + \text{Dauer}(d) + t_d\}$ 
19         for alle  $bf \in BF_l$  do
20           Replace( $bf$ , Start( $c$ ), Ziel( $c$ ),  $r_{opt}$ )
21         od
22       od
23     fi
24   fi
25 end

```

---

*Algorithmus 7.7: Pseudocode der Entscheidungslogik bei Fahrtumleitung aufgrund einer blockierten Verbindung*

Im Unterschied zur dynamischen Modifikation des Linienverlaufs, wird hier allerdings nicht die Originalroute als Initiallösung verwendet, da diese Route blockiert ist und nicht mehr verwendet werden kann (Zeile 08). Wie in Algorithmus 7.3 wird auch hier aus allen in Frage kommenden Alternativen, abhängig von den durch die Nutzerin getätigten Einstellungen, die schnellste oder pünktlichste Route bestimmt (Zeile 10/11). Die so bestimmte Route muss hier allerdings noch ein weiteres Kriterium erfüllen: Ihre Fahrzeit zwischen den Haltepunkten darf nicht mehr als  $Fr_{max}^{Diff}$  Zeiteinheiten von der Fahrzeit der Originalroute abweichen (Zeile 13). Hier findet also ein, durch die Nutzerin beeinflussbarer, Trade-Off zwischen der Sicherung des Service an allen ursprünglichen Haltepunkten und der dafür zusätzlich notwendigen Fahrzeit statt.

Erfüllt die schnellste/pünktlichste gefundene Route dieses Kriterium, müssen abschließend für alle betroffenen Linien alle von der Störung betroffenen Fahrten bestimmt werden (Zeile 16-18). Dies sind alle noch nicht beendeten Fahrten (Zeile 16), deren ausführendes Fahrzeug den Ort der Störung noch nicht erreicht hat (Zeile 17) – dies sind insbesondere auch alle noch nicht gestarteten Fahrten – und bei denen angenommen werden kann, dass sie den Ort der Störung vor deren Ablauf erreichen (Zeile 18). Da der Betrieb des Verkehrssystems auch während der Störung unvorhergesehenen Schwankungen unterliegt, ist es ratsam den Zeitrahmen der zu betrachtenden Fahrten weiter zu fassen als die eigentliche Dauer der Störung. Dazu wird die durch die Nutzerin zu definierende Pufferzeit  $t_a$  auf den Endzeitpunkt der Störung aufaddiert.

#### 7.2.2.2. Verkürzen

Ist ein Umleiten von Fahrzeugen nicht möglich, wird für alle betroffenen Linien zunächst überprüft ob sich der Ort der Störung so spät auf dem Linienverlauf befindet, dass ein Verkürzen von Fahrten sinnvoll ist. Auch dieses Verfahren ähnelt in seiner grundlegenden Funktionsweise seinem bereits beschriebenen Pendant mit gleichem Namen. Wie Algorithmus 7.8 aber zeigt, finden sich signifikante Unterschiede.

Zunächst ist die Position an der die Fahrtverkürzung stattfindet nicht fest. Stattdessen wird von dem der Störung vorgelagerten Haltepunkt aus rückwärts die erste Position gesucht, an der gewendet werden kann und die gewährleistet, dass der durch die Nutzerin zu definierende Mindestroutenanteil  $ra_{min}$  zur Sicherung eines Mindestservicelevels eingehalten wird (Zeile 04-11). Darüber hinaus muss nicht gewährleistet sein, dass Passagiere die von der Verkürzung betroffen sind einen alternativen Weg zu den restlichen Haltepunkten ihrer Route haben. Existiert eine zulässige Wendemöglichkeit, werden auch hier alle betroffenen Fahrten bestimmt und entsprechend verkürzt (Zeile 12-17).

Zwei weitere Besonderheiten im Vergleich zum Verfahren aus Algorithmus 7.4 finden sich in Zeile 18-23 und 27-29: Bewegt sich ein Fahrzeug auf den Ort der Störung zu und befindet sich direkt an der Position an der gewendet werden soll, müssen eventuell bereits terminierte Ereignisse aus der Future Event List entfernt und dem Fahrzeug explizit durch ein Ereignis mitgeteilt werden, dass es seine aktuelle Fahrt beenden soll (Zeile 18-23). Grund dafür ist, dass der Eintritt einer Störung mit dem Start des Fahrvorgangs durch das Fahrzeug zusammenfallen kann, d.h. der Dispatcher hat dem Fahrzeug zu einem früheren Simulationszeitpunkt bereits die Erlaubnis zur Weiterfahrt erteilt, das Fahrzeug befindet sich aber noch am Wendepunkt und steht kurz davor die nächste Fahraktivität zu starten. Würde dieses Ereignis in der FEL verbleiben, würde das Fahrzeug seine Fahrt fälschlicherweise fortsetzen und auf die blockierte Ressource auflaufen.

Die zweite Besonderheit betrifft die Startzeit der Folgefahrt  $\overline{bf}$  (Zeile 27-29). Anders als in Algorithmus 7.4 kann sie nicht einfach auf die planmäßige Abfahrtszeit am entsprechenden Haltepunkt gesetzt werden. Da von der Blockierung potentiell viele aufeinanderfolgende Fahrten betroffen sind, muss eine langfristige Stauung von Fahrzeugen am Wendepunkt vermieden werden. Deswegen wird die Startzeit der Folgefahrt soweit vorgezogen wie möglich. Dabei wird darauf geachtet, dass die Minute innerhalb des Taktes zu der  $\overline{bf}$  startet mit dem geplanten Wert übereinstimmt, d.h. eine Fahrt die planmäßig um 10:15 Uhr startet, sollte möglichst um XX:05 Uhr, XX:15 Uhr, XX:25 Uhr, etc. starten. Dies geschieht allerdings nur, wenn dadurch nicht die durch die Nutzerin zu definierte Wendezeit  $t_w$  verletzt wird.

**Input:** Linie  $l$ , Störung  $d$ , Verbindung  $c$ , Simulationszeitpunkt  $t$ , Mindestrouutenanteil  $ra_{min}$ , Pufferzeit  $t_d$ , Wendezeit  $t_w$

**Output:** Lösungs-Indikator found, Modifizierte Linienverläufe für alle betroffenen Fahrten

```

01 procedure Verkürzen( $l, d, c, t$ )
02 begin
03   found := false
04   pos := Position( $l, \text{Start}(c)$ )
05   while found = false and pos / |Haltepunkte( $l$ )|  $\geq ra_{min}$  do
06     if Wenden(Haltestelle( $l, pos$ )) = true then
07       found := true
08     else
09       pos := pos - 1
10     fi
11   od
12   if found = true then
13      $BF_l := \{bf \in \text{Fahrten}(l) \mid \text{Beendet}(bf) = \text{false} \wedge$ 
14       Position( $bf$ )  $\leq pos \wedge$ 
15       Startzeit( $bf$ ) + Fahrzeit( $bf, \text{Haltepunkt}(l, pos)$ )  $\leq t + \text{Dauer}(d) + t_d\}$ 
16     for alle  $bf \in BF_l$  do
17       CutOff( $bf, \text{Haltepunkt}(l, pos)$ )
18       if Position( $bf$ ) = pos then
19         if EntferneEreignisse(Fahrzeug( $bf$ )) then
20           ErzeugeEreignis(SERVICE_TRIP_END, max $\{t+1, \text{Zeit}(\text{Fahrzeug}(bf))\}$ ,
21             Fahrzeug( $bf$ ))
22         fi
23       fi
24       if  $\overline{bf} \neq 0$  then
25          $R := \text{Route}(\overline{bf})$ 
26         CutOff( $\overline{bf}, \text{Haltestelle}(\text{Haltepunkt}(l, pos))$ )
27          $a_{bf} := \text{Startzeit}(bf) + \text{Fahrzeit}(bf) /* \text{Geplante Abfahrtszeit von } bf */$ 
28          $a_{\overline{bf}} := \text{Startzeit}(\overline{bf}) + \text{Fahrzeit}(r, \text{Haltestelle}(\text{Haltepunkt}(l, pos)))$ 
29         Startzeit( $\overline{bf}$ ) :=  $a_{bf} + \max\{|a_{bf} - a_{\overline{bf}}| + \text{Takt}(l) \bmod \text{Takt}(l) |, t_w\}$ 
30       fi
31     od
32   fi
33 end

```

*Algorithmus 7.8: Pseudocode der Entscheidungslogik bei Fahrtverkürzung aufgrund einer blockierten Verbindung*

### 7.2.2.3. Überspringen

Bei dem in Abschnitt 7.2.2.1 vorgestellten Verfahren wurde versucht eine alternative Route zu bestimmen, bei der die Haltepunkte der ursprünglichen Route weiterhin bedient werden können. Ist eine solche Route nicht vorhanden und kommt ein Verkürzen der betroffenen Fahrten nicht in Frage, wird als nächstes versucht eine Alternativroute zu finden bei der Haltepunkte der ursprünglichen Route auch übersprungen werden können.

Die dazu verwendete Methode unterscheidet sich nur in dem zur Bestimmung der Alternativrouten verwendeten Ansatz vom Umleitungsverfahren und benutzt einen eigenständigen, durch die Nutzerin zu spezifizierenden, Grenzwert  $Fsk_{max}^{Diff}$  anstatt  $Fr_{max}^{Diff}$  für die Fahrzeitabweichung.

Das Verfahren zur Bestimmung der Alternativrouten das Zeile 06/07 in Algorithmus 7.7 ersetzt, zeigt Algorithmus 7.9. Es bestimmt alle einfachen Wege im Netzwerk, die mindestens alle Haltepunkte zwischen den Indizes  $l_0$  und  $h_i$  überspringen, aber nicht mehr als  $rs_{max}$ , wobei dieser Grenzwert ebenfalls von der Nutzerin festzulegen ist. Dabei startet  $l_0$  an dem der Störung unmittelbar vorgelagerten Haltepunkt, während  $h_i$  am unmittelbar nachgelagerten Haltepunkt startet. Zwischen diesen Indizes werden alternative unblockierte Wege gesucht, die eine Maximallänge – hier die Zahl der Haltepunkt auf der Ursprungsrouten – nicht überschreiten (Zeile 10/11). Wird kein solcher Weg gefunden werden die Indizes erhöht resp. verringert und das Verfahren wiederholt (Zeile 13/17), sofern der Maximalwert zu überspringender Ressourcen nicht verletzt wird.

**Input:** Linie  $l$ , Haltepunkt-Indizes  $l_0$  und  $h_i$ , Maximale Umleitungslänge  $r_{s_{max}}$ ,  
Maximale Weglänge  $wl_{max}$

**Output:** Menge  $R_{skip}$  der einfachen Wege, die Ressourcen zwischen  $l_0$  und  $h_i$  überspringen,  
Indizes  $l_0$  und  $h_i$  zwischen denen das Überspringen stattfindet

```

01 procedure AllSimpleSkippingPaths( $l$ ,  $l_0$ ,  $h_i$ )
02 begin
03    $R_{skip} := \emptyset$ 
04   found := true
05   tmp :=  $h_i$ 
06   while  $R_{skip} = \emptyset$  and  $l_0 \geq 0$  do
07      $h_i := tmp$ 
08     while  $R_{skip} = \emptyset$  and  $h_i < |\text{Haltepunkte}(l)|$  and
09        $(h_i - l_0) - 1 / |\text{Haltepunkte}(l)| \leq r_{s_{max}}$  do
10        $R_{skip} := \text{AllSimplePaths}(\text{Haltepunkt}(l, l_0), \text{Haltepunkt}(l, h_i), wl_{max})$ 
11        $R_{skip} := R_{skip} \setminus \{r \in R_{skip} \mid \text{Blockiert}(r) \wedge |\text{Haltepunkte}(r)| > |\text{Haltepunkte}(l)|\}$ 
12       if  $R_{skip} = \emptyset$  then
13          $h_i := h_i + 1$ 
14       fi
15     od
16     if  $R_{skip} = \emptyset$  then
17        $l_0 := l_0 - 1$ 
18     fi
19   od
20 end

```

*Algorithmus 7.9: Pseudocode zur Bestimmung aller einfachen, überspringenden Wege*

#### 7.2.2.4. Auftrennen

Die bisher vorgestellten Verfahren haben versucht die Effekte von Störungen mit relativ moderaten Eingriffen in die Linienverläufe der unmittelbar betroffenen Fahrten und weitestgehend ohne Änderungen am Fahrplan einzudämmen. Ist keiner dieser Ansätze erfolgreich, wird als letzte Maßnahme eine umfangreiche Modifikation in Erwägung gezogen, die über die unmittelbar durch die Störung betroffenen Fahrten hinausgeht: Für den Zeitraum der Störung werden die Fahrten der betroffenen Linien auf beiden Seiten des Orts der Störung bzw. möglichst nah an diesem Ort verkürzt und der Betrieb auf dem gestörten Bereich eingestellt. Dabei wird zur Reduzierung der Effekte der Störung in Kauf genommen, dass auch auf den verkürzten Bereichen kein durchgehender Betrieb gewährleistet werden kann, da nur noch eine Untermenge der eigentlich auf ihnen verkehrenden Fahrzeuge zur Verfügung steht.<sup>23</sup> Algorithmus 7.10 zeigt im Pseudocode einen Ausschnitt der zur Umsetzung dieses Ansatzes verwendeten Entscheidungslogik.

Zunächst werden die dem Ort der Störung nächst gelegenen Wendepunkte bestimmt. Dies geschieht auf der, aus Sicht der gestörten Richtung, zur Störung hinführenden Seite, indem von dem der Störung vorgelagerten Haltepunkt aus rückwärts dem Linienverlauf gefolgt wird (Zeile 03-06). Auf der von der Störung wegführenden Seite geschieht dies entsprechend analog (Zeile 08-12). Im Gegensatz zum einfachen Verkürzen oder Überspringen, muss hier kein Mindestprozentsatz der Ursprungsrouten erfüllt oder eine maximale Anzahl überspringbarer Ressourcen beachtet werden.

Wurden zwei Wendepunkte gefunden (Zeile 14)<sup>24</sup>, müssen die betroffenen Fahrten bestimmt und verkürzt werden. Dies muss für beide Seiten der Störung erfolgen, wird in Algorithmus 7.10 der Übersichtlichkeit halber aber nur für die Seite beschrieben die aus Sicht der direkt betroffenen Linie zur Störung hinführt.

<sup>23</sup> In einem Realsystem würde hier, je nach Umfang der Störung, Ersatzverkehr eingerichtet. Aufgrund fehlender Daten zum Fahrzeugpool und dem das ÖPNV-Netz umgebenden Straßennetz kann dies aktuell leider nicht mit eingebunden werden.

<sup>24</sup> Prinzipiell wäre es möglich das Verfahren auch nur für eine Seite der Störung durchzuführen, d.h. nur eine der Bedingungen der if-Abfrage in Zeile 14 wäre erfüllt. Abhängig davon auf welcher Seite der Störung verkürzt würde, würde dann aber entweder die Taktfrequenz der ungestörten Richtung ungewollt verdoppelt oder der Teilabschnitt der ungestörten Richtung der (für diese Richtung) hinter der Störung läge würde gar nicht mehr bedient. Beide Resultate sind wenig wünschenswert.

**Input:** Linie  $l$ , Störung  $d$ , Verbindung  $c$ , Simulationszeitpunkt  $t$ , Pufferzeit  $t_d$ , Wendezeit  $t_w$

**Output:** Modifizierte Linienverläufe für alle betroffenen Fahrten

```

01 procedure Auftrennen( $l$ ,  $d$ ,  $c$ ,  $t$ )
02 begin
03    $tp_{start} :=$  Position( $l$ , Start( $c$ ))
04   while Wenden(Haltestelle( $l$ ,  $tp_{start}$ )) = false and  $tp_{start} >= 1$  do
05      $tp_{start} := tp_{start} - 1$ 
06   od
07
08    $tp_{ziel} :=$  Position( $l$ , Ziel( $c$ ))
09   while Wenden(Haltestelle( $l$ ,  $tp_{ziel}$ )) = false and
10      $tp_{ziel} < |Haltestellen(l)| - 1$  do
11      $tp_{ziel} := tp_{ziel} + 1$ 
12   od
13
14   if  $tp_{start} > 0$  and  $tp_{ziel} < |Haltestellen(l)| - 1$  then
15      $\bar{l} :=$  Rückrichtung( $l$ )
16      $BF_l := \{bf \in$  Fahrten( $l$ ) | Beendet( $bf$ ) = false  $\wedge$ 
17       Position( $bf$ )  $\leq$   $tp_{start}$   $\wedge$ 
18       Startzeit( $bf$ ) + Fahrzeit( $bf$ , Haltepunkt( $l$ ,  $tp_{start}$ ))  $\leq$   $t +$  Dauer( $d$ ) +  $t_d$ \}
19      $BF_l := BF_l \cup \{bf \in$  Fahrten( $\bar{l}$ ) | Beendet( $bf$ ) = false  $\wedge$ 
20       Position( $bf$ )  $>$  Position(Haltestelle( $l$ ,  $tp_{ziel}$ ))  $\wedge$ 
21       Startzeit( $bf$ ) + Fahrzeit( $bf$ , Haltestelle( $l$ ,  $tp_{ziel}$ ))  $\leq$   $t +$  Dauer( $d$ ) +  $t_d$ \}
22      $F_l :=$  Fahrzeuge( $BF_l$ )
23     for alle  $f \in F_l$  do
24       for alle  $bf \in$  Fahrten( $f$ ):  $bf \in BF_l$  do
25         if Linie( $bf$ ) =  $l$  then
26           if Position( $bf$ ) =  $tp_{start}$  then
27             if EntferneEreignisse( $f$ ) then
28               ErzeugeEreignis(SERVICE_TRIP_END, max( $t + 1$ , Zeit( $f$ )),  $f$ )
29             fi
30             CutOff( $bf$ , Haltepunkt( $l$ ,  $tp_{start}$ ))
31           else if Position( $bf$ )  $<$   $tp_{start}$  then
32             CutOff( $bf$ , Haltepunkt( $l$ ,  $tp_{start}$ ))
33             if  $bf \neq$  First(Fahrten( $f$ )) and Modifiziert( $bf - 1$ ) = true then
34                $a_{bf-1} :=$  Startzeit( $bf - 1$ ) + Fahrzeit( $bf - 1$ )
35                $a_{bf} :=$  Startzeit( $bf$ )
36               Startzeit( $bf$ ) :=  $a_{bf-1} +$  max( $|a_{bf-1} - a_{bf}| +$  Takt( $l$ ) mod Takt( $l$ ),  $t_w$ )
37             fi
38           fi
39           else if Linie( $bf$ ) =  $\bar{l}$  and  $bf \neq$  First(Fahrten( $f$ )) then
40             if Modifiziert( $bf - 1$ ) = true then
41               CutOff( $bf$ , Haltestelle( $l$ ,  $tp_{ziel}$ ))
42                $a_{bf-1} :=$  Startzeit( $bf - 1$ ) + Fahrzeit( $bf - 1$ )
43                $a_{bf} :=$  Startzeit( $bf$ )
44               Startzeit( $bf$ ) :=  $a_{bf-1} +$  max( $|a_{bf-1} - a_{bf}| +$  Takt( $l$ ) mod Takt( $l$ ),  $t_w$ )
45             fi
46           fi
47         od
48       od
49     /* Analog für Fahrten der Rückrichtung samt zugehöriger Fahrzeuge */
50   fi
51 end

```

**Algorithmus 7.10:** Pseudocode zur Entscheidungslogik beim Auftrennen von Fahrten aufgrund einer Störung

Wie bei den bisherigen Verfahren sind zunächst die Fahrten betroffen, die noch nicht beendet wurden, deren ausführendes Fahrzeug sich vor oder auf dem identifizierten Wendepunkt befindet und bei denen zu erwarten ist, dass sie vor Auflösung der Störung am Wendepunkt eintreffen (Zeile 16-18). Darüber hinaus sind aber auch die Fahrten der entgegengesetzten Richtung betroffen, die noch nicht beendet wurden, deren ausführendes Fahrzeug sich bereits hinter dem Wendepunkt der anderen Seite der Störung befindet und für die anzunehmen ist, dass sie diesen Punkt vor Auflösung der Störung erreichen werden (Zeile 19-21).

Sind alle betroffenen Fahrten bestimmt, muss für die sie ausführenden Fahrzeuge über die entsprechenden Fahrten iteriert und diese modifiziert werden. Dabei sind folgende Fälle zu unterscheiden:

1. *Die Fahrt gehört zur gestörten Richtung und führt zur Störung hin (Zeile 25-38):* In diesem Fall müssen zwei weitere Fälle unterschieden werden.
  - i. *Das ausführende Fahrzeug befindet sich an der Wendeposition (Zeile 27-30):* Dies ist der gleiche Fall wie beim einfachen Verkürzen (vgl. Algorithmus 7.8). Bereits terminierte Ereignisse müssen aus der FEL entfernt und ein neues Ereignis zur Signalisierung des Fahrtendes muss erzeugt werden.
  - ii. *Das ausführende Fahrzeug befindet sich vor der Wendeposition (Zeile 31-38):* Hier kann zunächst ganz normal verkürzt werden. Anders als beim Verfahren aus Algorithmus 7.8 muss hier allerdings zusätzlich überprüft werden ob auch die Startzeit angepasst werden muss, da die aktuell betrachtete Fahrt nicht zwangsläufig auch die aktuell durch das Fahrzeug ausgeführte Fahrt sein muss, sondern auch eine zukünftige Fahrt des Fahrzeugs sein kann. Die Startzeit darf dabei nur angepasst werden, wenn die Fahrt nicht bereits ausgeführt wird (Zeile 33, erste Bedingung) und wenn die vorhergehende Fahrt des Fahrzeugs  $bf - 1$  modifiziert wurde (Zeile 33, zweite Bedingung). Die eigentliche Modifikation findet dann analog zum Verfahren aus Algorithmus 7.8 statt.
2. *Die Fahrt gehört zur nicht gestörten Richtung und führt von der Störung weg (Zeile 39-46):* In diesem Fall darf eine Verkürzung der aktuell betrachteten Fahrt nur erfolgen, wenn sie nicht bereits ausgeführt wird (Zeile 39, zweite Bedingung) und die vorhergehende Fahrt des Fahrzeugs  $bf - 1$  modifiziert wurde (Zeile 40). Auch hier erfolgt die Verkürzung und Berechnung der neuen Startzeit analog zu Algorithmus 7.8.

Auf diese Weise werden alle Fahrten der Fahrzeuge modifiziert, die später, aus Sicht der gestörten Richtung, vor der Störung verkehren. Anschließend müssen in analoger Weise ebenfalls noch die Fahrten der Fahrzeuge modifiziert werden die hinter der Störung verkehren werden.

Kann der Dispatcher eine Störung selbst durch das Auftrennen von Linien nicht auflösen, wird versucht den Betrieb des Systems ohne explizite Adressierung der Störung fortzuführen. Dabei ist, insbesondere bei langfristigen Störungen, zu erwarten, dass das System sehr instabil wird und unrealistisch hohe Verspätungskennzahlen generiert, da Fahrten nicht abgebrochen werden.

Zusammenfassend lässt sich festhalten, dass der Dispatcher mit den in diesem Abschnitt beschriebenen Verfahren in der Lage ist alle Störungen zu adressieren, die ohne den Einsatz von Ersatzfahrzeugen bzw. Ersatzverkehr auskommen und kein weiteres Expertenwissen benötigen. Letzteres ist beispielsweise nötig, wenn Störungen dadurch adressiert werden sollen, indem Fahrten in andere Bereiche des Netzes ausweichen, d.h. ab einem spezifizierten Punkt im Netz dem Verlauf einer anderen Linie folgen. Die Basis zur Umsetzung solcher Verfahren ist allerdings bereits vorhanden, da sowohl im Liniennanager als auch in der Datenbank die Möglichkeit besteht vordefinierte Alternativlinien zu verwalten, auf die im Falle einer Störung ausgewichen werden kann.

## 7.3. Validierung

Die Funktionalität und Plausibilität der beschriebenen Verfahren wird auszugsweise mittels einer funktions- und theoriebezogenen Validierung überprüft. Um dabei einen möglichst großen Querschnitt aller relevanter Ansätze abzudecken, werden die Anti-Bunching Haltestrategie, die dynamische Modifikation von Linienvläufen, sowie das Auftrennen des Linienvlaufs bei Eintritt einer Störung untersucht.

Alle Experimente werden auf einem Intel Core i7-740QM Quad Core mit 1,73 GHz und 6 GB RAM ausgeführt.

### 7.3.1. Haltepunkt-Kontrolle

Für die Untersuchungen zur Haltepunkt-Kontrolle wird erneut das künstliche Netz UCL 2015 herangezogen, da es wegen seiner Übersichtlichkeit eine einfache Kontrolle ermöglicht. Für alle Untersuchungen wird der bereits in Abschnitt 5.3 untersuchte Optimalfahrplan UCL2015<sup>+</sup> mit den dort angegeben

Fahrzeitparametern sowie den in Tabelle 5.4 angegebenen Ankunftsdaten verwendet. Alle Kennzahlen basieren auf Durchschnittswerten über jeweils 100 Simulationsläufe und werden im Simulationszeitraum zwischen acht Uhr morgens und sechs Uhr abends erhoben.

### 7.3.1.1. Anti-Bunching

Die Anti-Bunching Haltestrategie wird von drei Parametern beeinflusst: Welche Haltepunkte werden als Kontrollpunkte festgelegt, welche Information über die Abweichung des Fahrzeugs wird zur Entscheidungsfindung herangezogen und welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung  $a_{max}^{Diff}$  an. Zur Untersuchung der Funktionalität und Plausibilität werden diese Parameter systematisch variiert und die so formulierten unterschiedlichen Strategien miteinander und mit dem Fall ohne Einsatz einer Haltestrategie verglichen. Dazu werden die beobachteten Bunching-Effekte sowie die über die Haltepunkte des Netzes gemessene durchschnittliche (maximale) Verspätung, Frühzeitigkeit und Wartezeit als Kennzahlen herangezogen. Tabelle 7.2 und Tabelle 7.3 zeigen die erhaltenen Ergebnisse.

Da selbst beim relativ übersichtlichen Netz UCL 2015 die Untersuchung aller potentiell möglichen Kombinationen von Kontrollpunkten nicht in akzeptabler Zeit – und innerhalb eines für Leser zumutbaren Seitenumfangs – möglich ist, wird sich bei der Untersuchung auf zwei unterschiedliche Parametervariationen konzentriert: Anwendung der Haltestrategie an den Starthaltepunkten der einzelnen Linien (vgl. dazu auch Abschnitt 5.3.1 und Abbildung 3.2) und Anwendung der Strategie an allen Haltepunkten. Der Grenzwert der Abfahrtszeitabweichung wird in Zehn-Sekunden-Schritten zwischen null und 30 Sekunden variiert. Zusätzlich wird auch ein größerer Grenzwert von 60 Sekunden untersucht. Zusammen mit den beiden möglichen Ausprägungen für die zur Entscheidungsfindung heranzuziehende Information (nur die lokale Abweichung oder auch Umgebungsinformationen, d.h. die Pünktlichkeit am nächsten Haltepunkt) ergeben sich so 20 unterschiedliche Variationen der Anti-Bunching Strategie.

Der Vergleich der beobachteten Bunching-Effekte in Tabelle 7.2 bestätigt die Funktionalität der Haltestrategie und zeigt zu erwartende Muster: Die Anwendung einer beliebigen Strategie führt, im Vergleich zum Fall ohne Haltestrategie, zu einer Reduktion des Anteils reduzierter Sicherheitsabstände. Die Reduktion fällt dabei mit zunehmendem Wert von  $a_{max}^{Diff}$  ab und fällt stärker aus wenn alle Haltepunkte als Kontrollpunkte fungieren. Ersteres erklärt sich dadurch, dass mit zunehmendem  $a_{max}^{Diff}$  die Abfahrtszeitabweichung nicht mehr durch diesen Grenzwert, sondern durch die anhand der Simulationsparameter bzw. empirischen Daten festgelegten Fahr- und Haltezeiten limitiert wird, sich also dem unkontrollierten Fall annähert.

Auf den ersten Blick überraschend ist, dass es keine signifikanten Unterschiede bei den Bunching-Effekten zwischen der Verwendung von lokalen Informationen und der Einbeziehung der Pünktlichkeit am nächsten Haltepunkt gibt. Um zu verstehen warum dies der Fall ist muss man sich klarmachen wann die zweite dieser Varianten überhaupt zu einer Anpassung der Haltezeit, gegenüber der Variante mit lokaler Information, führt. Dies ist genau dann der Fall wenn ein Fahrzeug an seinem aktuellen Haltepunkt zu früh ist, die Abweichung über dem Grenzwert  $a_{max}^{Diff}$  liegt und die zu erwartende Frühzeitigkeit am nächsten Haltepunkt kleiner als die potentielle Wartezeit am aktuellen Haltepunkt ist bzw. zu erwarten ist, dass das Fahrzeug am nächsten Haltepunkt verspätet sein wird. Betrachtet man nun erneut die Abweichungsverläufe der einzelnen Linien in Netz UCL 2015 aus Abschnitt 5.3.1, stellt man fest, dass nur an wenigen Punkten des Netzes alle diese Bedingungen erfüllt sind. Entsprechend hat die zur Entscheidungsfindung verwendete Information unter Fahrplan UCL2015+ keine signifikanten Auswirkungen auf die beobachteten Bunching-Effekte.

Die Kennzahlen zur Verspätung, Frühzeitigkeit und Wartezeit aus Tabelle 7.3 bestätigen die beobachteten Muster. Die Kennzahlen wurden dabei berechnet, indem an jedem Haltepunkt des Netzes die durchschnittliche und maximale Verspätung/Frühzeitigkeit/Wartezeit über die dort stattfindenden Abfahrten bestimmt wurde. Diese Werte wurden anschließend über die durchgeführten Simulationsläufe und Haltepunkte des Netzes gemittelt.

Strategie			Durch. kum. Sicherheitsabstandsreduktion [%]					
Kontrollpunkte	Information	$a_{max}^{Diff}$ [Sek.]	> 0	>= 10	>= 20	>= 30	>= 40	>= 50
			-	-	-	24,72	9,52	2,25
Start-HP	Lokal	0	22,90	6,56	1,18	0,22	0,04	0,01
Start-HP	Lokal	10	23,05	6,51	1,16	0,22	0,04	0,01
Start-HP	Lokal	20	23,72	7,63	1,42	0,25	0,04	0,01
Start-HP	Lokal	30	24,28	8,83	1,78	0,28	0,04	0,01
Start-HP	Lokal	60	24,63	9,48	2,22	0,40	0,06	0,01
Alle	Lokal	0	<b>20,24</b>	3,28	<b>0,53</b>	0,16	<b>0,03</b>	<b>0,00</b>
Alle	Lokal	10	20,56	3,36	<b>0,53</b>	0,17	0,04	0,01
Alle	Lokal	20	22,91	4,47	0,61	<b>0,13</b>	<b>0,03</b>	<b>0,00</b>
Alle	Lokal	30	23,79	6,33	0,84	0,16	<b>0,03</b>	0,01
Alle	Lokal	60	24,24	9,11	1,84	0,32	0,05	0,01
Start-HP	Umgb.	0	23,51	6,10	1,16	0,32	0,10	0,02
Start-HP	Umgb.	10	23,05	6,57	1,16	0,22	0,05	0,01
Start-HP	Umgb.	20	23,79	7,58	1,41	0,25	0,05	0,01
Start-HP	Umgb.	30	24,30	8,73	1,74	0,30	0,05	0,01
Start-HP	Umgb.	60	24,62	9,54	2,28	0,41	0,07	0,01
Alle	Umgb.	0	20,92	<b>3,20</b>	0,68	0,24	0,05	0,01
Alle	Umgb.	10	20,70	3,34	0,54	0,17	<b>0,03</b>	<b>0,00</b>
Alle	Umgb.	20	22,86	4,46	0,62	0,13	<b>0,03</b>	0,01
Alle	Umgb.	30	23,74	6,38	0,85	0,15	<b>0,03</b>	0,01
Alle	Umgb.	60	24,41	9,21	1,85	0,30	0,05	0,01

*Tabelle 7.2: Durchschnittlicher, kumulierter Anteil reduzierter Sicherheitsabstände unter unterschiedlichen Haltestrategien, UCL 2015. Erste Spalte: Start-HP – Nur die Starthaltepunkte fungieren als Kontrollpunkte; Alle – Alle Haltepunkte fungieren als Kontrollpunkte. Zweite Spalte: Lokal – Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. – Die erwartete Pünktlichkeit am nächsten Haltepunkt wird mit zur Entscheidungsfindung herangezogen. Dritte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an*

Wie zu erwarten, steigt die durchschnittliche und maximale Verspätung mit sinkendem  $a_{max}^{Diff}$ , da Fahrzeuge nicht mehr in der Lage sind Zeitpuffer zu bilden und potentielle Verspätungen so aufzufangen. Gleichzeitig sinken die Frühzeitigkeit und Wartezeit. Ebenfalls plausibel ist, dass die maximale Wartezeit höher ausfällt, wenn nur ausgewählte Haltepunkte als Kontrollpunkte verwendet werden, da Fahrzeuge zwischen ihnen größere Zeitpuffer herausfahren, als wenn alle Haltepunkte als Kontrollpunkte fungieren. Da die durch Fahrzeuge vor Kontrollpunkten eingefahrenen Zeitpuffer direkt in zusätzliche Wartezeit umgewandelt werden, steigt so die maximale Wartezeit

Betrachtet man alle erhobenen Kennzahlen in Kombination, erscheint eine Haltestrategie bei der alle Haltepunkte als Kontrollpunkte fungieren, nur lokale Informationen verwendet werden und  $a_{max}^{Diff}$  auf zehn Sekunden gesetzt wird für Fahrplan UCL2015<sup>+</sup> ideal. Diese Strategie weist keine signifikant höheren Bunching-Effekte als die Strategie mit erzwungener planmäßiger Abfahrt auf, resultiert gleichzeitig aber in einer weniger starken Erhöhung der durchschnittlichen und maximalen Verspätung, bei gleichzeitig deutlicher Reduktion der durchschnittlichen und maximalen Frühzeitigkeit.

Strategie		$\alpha_{max}^{Diff}$ [Sek.]	Verspätung [Sek.]		Frühzeit. [Sek.]		Wartezeit [Sek.]	
Kontrollpunkte	Information		$\emptyset$	Max.	$\emptyset$	Max.	$\emptyset$	Max.
-	-	-	4,4	84,6	14,1	104,8	<b>0,0</b>	<b>0,0</b>
Start-HP	Lokal	0	8,5	95,6	4,6	61,7	1,5	57,0
Start-HP	Lokal	10	4,6	85,6	10,7	71,4	0,4	58,2
Start-HP	Lokal	20	4,4	84,3	12,5	81,3	0,2	55,3
Start-HP	Lokal	30	4,4	84,1	13,5	88,5	0,1	44,9
Start-HP	Lokal	60	4,4	<b>82,3</b>	14,1	104,0	<b>0,0</b>	16,9
Alle	Lokal	0	10,4	98,4	<b>0,0</b>	<b>0,0</b>	2,6	27,9
Alle	Lokal	10	5,4	89,6	5,0	10,0	1,5	27,6
Alle	Lokal	20	4,4	82,8	8,8	20,0	0,8	26,7
Alle	Lokal	30	<b>4,3</b>	85,4	11,1	30,0	0,5	25,3
Alle	Lokal	60	4,4	84,7	13,7	60,0	0,1	20,5
Start-HP	Umgb.	0	7,1	98,4	5,6	62,4	1,3	58,4
Start-HP	Umgb.	10	4,6	86,9	10,8	69,9	0,5	57,8
Start-HP	Umgb.	20	<b>4,3</b>	84,8	12,6	81,0	0,2	56,4
Start-HP	Umgb.	30	<b>4,3</b>	82,8	13,5	90,3	0,1	47,1
Start-HP	Umgb.	60	4,4	86,4	14,1	104,3	<b>0,0</b>	15,2
Alle	Umgb.	0	8,4	100,4	0,7	13,4	2,4	27,7
Alle	Umgb.	10	5,3	89,7	5,2	15,2	1,5	27,6
Alle	Umgb.	20	4,4	84,6	8,8	20,0	0,9	26,7
Alle	Umgb.	30	<b>4,3</b>	84,6	11,2	30,0	0,5	24,9
Alle	Umgb.	60	4,4	83,8	13,7	60,0	0,1	20,4

*Tabelle 7.3: Verspätung, Frühzeitigkeit und Wartezeit unter unterschiedlichen Haltestrategien, UCL 2015. Erste Spalte: Start-HP – Nur die Starthaltepunkte fungieren als Kontrollpunkte; Alle - Alle Haltepunkte fungieren als Kontrollpunkte. Zweite Spalte: Lokal – Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. – Die erwartete Pünktlichkeit am nächsten Haltepunkt wird zur Entscheidungsfindung herangezogen. Dritte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an*

### 7.3.1.2. Dynamische Modifikation von Linienverläufen

Um die Funktionalität der dynamischen Modifikation von Linienverläufen zu überprüfen wird zunächst ein simples Störfallszenario simuliert. Dieses Szenario besteht lediglich aus einer einzelnen Störung, die für die komplette Dauer des Betriebstages die Fahrzeit auf der Verbindung zwischen den Haltepunkten 1121 und 1131 um 80 Prozent erhöht. Die planmäßige Fahrzeit auf dieser Verbindung beträgt 180 Sekunden und im Regelbetrieb benötigen Fahrzeuge durchschnittlich 149 Sekunden um die Verbindung zu passieren. Durch die Störung wird die durchschnittliche Fahrzeit entsprechend auf 268 Sekunden erhöht, wodurch der Betrieb deutlich beeinträchtigt werden sollte. Ziel dieser Untersuchung ist zu überprüfen, ob der Dispatcher dazu in der Lage ist die Störung zu erkennen und durch dynamisches Umleiten der Fahrzeuge der betroffenen Linien die Effekte der Störung zu minimieren.

Um die Modifikation des Linienverlaufs der betroffenen Linien zu ermöglichen, ist eine alternative Verbindung zwischen den Haltepunkten notwendig. Netz UCL 2015 wird deswegen um eine entsprechende Verbindung zwischen den Haltepunkten 1121 und 1131, mit einer planmäßigen Fahrzeit von 180 Sekunden, erweitert.

Zur Abschätzung der Güte des Verfahrens werden vier verschiedene Strategien miteinander und mit dem ungestörten Basisfall verglichen. Letzterer bezeichnet dabei den bereits in Abschnitt 5.3.1 beschriebenen ungestörten Regelbetrieb, bei dem die Starthaltepunkte aller Linien als Kontrollpunkte fungieren, Haltezeiten nur aufgrund lokaler Informationen bestimmt werden und der Grenzwert der Abweichung null Sekunden beträgt. Die vier untersuchten Strategien sind:

Keine Kontrolle (-), Anwendung der Haltestrategie des Basisfalls ohne Modifikation von Linienverläufen (A), Modifikation von Linienverläufen bei Detektion einer erwarteten Verspätung von mindestens 60 Sekunden ohne Einsatz einer Haltestrategie (B) und die kombinierte Anwendung der Strategien A und B (C).

Tabelle 7.4 und Tabelle 7.5 zeigen die unter Verwendung der unterschiedlichen Strategien und im Basisfall auftretende durchschnittliche Verspätung bzw. Frühzeitigkeit der einzelnen Linien, während Tabelle 7.6 an Haltepunkt 1131 gemessene relevante Kennzahlen auflistet.

Zunächst lässt sich festhalten, dass ohne Modifikation der Linienverläufe der Betrieb des Systems erheblich beeinträchtigt wird. Ohne jegliche Haltepunkt-Kontrolle steigt die durchschnittliche Verspätung pro Linie um 206,7 Prozent und die durchschnittliche Frühzeitigkeit um 106,0 Prozent, im Vergleich zum Basisfall. Die Steigerung in der Verspätung schlägt sich dabei einzig in den direkt durch die Störung betroffenen Linien 1-B01, 2-B02 und 3-B01 nieder, während die anderen Linien, wie zu erwarten, ihre Verspätung – im Ausgleich für eine erhöhte Frühzeitigkeit – reduzieren können.

Unter Strategie A, fällt die Erhöhung der durchschnittlichen Verspätung mit 289,3 Prozent noch deutlich höher aus. Darüber hinaus greifen die Effekte der Störung unter dieser Strategie auch auf die nur indirekt betroffene Linie 4-B02 über, deren Verspätung sich um 59,5 Prozent erhöht. Die durchschnittliche Frühzeitigkeit kann unter dieser Strategie hingegen um 32,0 Prozent reduziert werden, wobei diese Reduktion vollständig auf die direkt durch die Störung betroffenen Linien ausfällt.

Die an Haltepunkt 1131 gemessenen Kennzahlen bestätigen diese Beobachtungen. Ohne Haltepunkt-Kontrolle und unter Strategie A sind von 150 im Messzeitraum stattfindenden Abfahrten 150 verspätet und keine dieser Abfahrten weist dabei eine geringere Verspätung als 25,5 Sekunden bzw. 68,6 Sekunden auf.

Unter den Strategien B und C scheint die Störung hingegen keinerlei Effekt zu haben. Während die durchschnittliche Linienverspätung unter Strategie B erwartungsgemäß auf Kosten einer erhöhten Frühzeitigkeit, im Vergleich zum Basisfall, um 46,7 Prozent sinkt, zeigt das System unter Strategie C weder bei den Linienkennzahlen noch bei den Haltepunktkennzahlen eine signifikante Abweichung zum Basisfall. Der Dispatcher scheint die Störung, wie gewünscht, zu erkennen und betroffene Fahrten dynamisch auf die ungestörte Verbindung umzuleiten.

Um die Leistungsfähigkeit der unterschiedlichen dem Dispatcher zur Verfügung stehenden Strategien zur Linienmodifikation – auch in Kombination mit unterschiedlichen Haltestrategien – zu überprüfen, werden weitere experimentelle Untersuchungen durchgeführt. Dazu wird Netz UCL 2015 so modifiziert, dass zwischen allen benachbarten Haltepunkten jeweils drei Verbindungen mit planmäßigen Fahrzeiten von 60, 120 und 180 Sekunden existieren. Dies ist ein bewusst gewähltes Extrembeispiel um

Linie	Strategie				Basisfall
	-	A	B	C	
1-B01	51,0	55,9	<b>3,8</b>	7,9	8,4
1-B02	8,7	14,6	<b>8,8</b>	14,7	14,7
2-B01	0,7	4,4	<b>0,7</b>	4,4	4,4
2-B02	65,6	72,8	<b>0,5</b>	0,8	0,8
3-B01	46,1	61,1	<b>9,3</b>	11,4	11,4
3-B02	1,6	4,6	<b>1,4</b>	4,3	4,4
4-B01	3,6	6,6	<b>4,4</b>	7,7	7,6
4-B02	6,6	13,4	<b>2,7</b>	8,4	8,4
∅	23,0	29,2	<b>4,0</b>	7,5	7,5

*Tabelle 7.4: Durchschnittliche Linienverspätung bei Auftritt einer Störung unter unterschiedlichen Strategien, UCL 2015. A: Nur Haltestrategie, ausgewählte Kontrollpunkte, lokale Information, Grenzwert von null Sekunden. B: Modifikation des Linienverlaufs bei Detektion einer erwarteten Verspätung von mindestens 60 Sekunden. C: Kombinierte Anwendung von A und B*

Linie	Strategie				Basisfall
	-	A	B	C	
1-B01	2,8	<b>0,3</b>	6,2	2,1	1,9
1-B02	7,2	<b>1,6</b>	7,2	<b>1,6</b>	1,6
2-B01	17,0	<b>4,5</b>	17,4	<b>4,5</b>	4,6
2-B02	5,5	<b>1,6</b>	28,0	12,1	12,2
3-B01	11,3	<b>1,9</b>	18,4	5,8	5,9
3-B02	15,5	<b>5,4</b>	15,5	<b>5,4</b>	5,4
4-B01	13,5	6,8	12,7	<b>6,1</b>	6,2
4-B02	9,9	4,9	7,7	<b>2,0</b>	2,0
∅	10,3	<b>3,4</b>	14,1	4,9	5,0

*Tabelle 7.5: Durchschnittliche Frühzeitigkeit der Linien bei Auftritt einer Störung unter unterschiedlichen Strategien, UCL 2015. A: Nur Haltestrategie, ausgewählte Kontrollpunkte, lokale Information, Grenzwert von null Sekunden. B: Modifikation des Linienverlaufs bei Detektion einer erwarteten Verspätung von mindestens 60 Sekunden. C: Kombinierte Anwendung von A und B*

Strategie	Abf.	Versp. Abf.	Frühz. Abf.	Min. Versp. [Sek.]	Durch. Versp. [Sek.]	Max. Versp. [Sek.]	Min. Frühz. [Sek.]	Durch. Frühz. [Sek.]	Max. Frühz. [Sek.]
-	150,0	150,0	0,0	25,5	101,2	162,2	0,0	0,0	0,0
<b>A</b>	150,0	150,0	0,0	68,6	117,4	172,8	0,0	0,0	0,0
<b>B</b>	150,0	45,8	101,7	1,1	5,8	72,5	1,1	17,9	67,7
<b>C</b>	150,0	67,5	80,1	1,1	6,8	47,7	1,1	8,9	43,6
<b>Basisfall</b>	150,0	68,2	79,3	1,0	6,9	48,1	1,1	8,9	43,8

**Tabelle 7.6: Relevante Kennzahlen von Haltepunkt 1131 bei Auftritt einer Störung unter unterschiedlichen Strategien, UCL 2015. A: Nur Haltestrategie, ausgewählte Kontrollpunkte, lokale Information, Grenzwert von null Sekunden. B: Modifikation des Linienverlaufs bei Detektion einer erwarteten Verspätung von mindestens 60 Sekunden. C: Kombinierte Anwendung von A und B**

dem Dispatcher die Möglichkeit zu geben, an jedem Kontrollpunkt mittels Linienverlaufsmodifikation sowohl auf verspätete als auch frühzeitige Fahrzeuge reagieren zu können.

Da unter Fahrplan UCL2015<sup>+</sup> im Regelbetrieb nur in Einzelfällen eine Verspätung von mehr als 20 Sekunden auftritt (vgl. Abschnitt 5.3.1), würden unter realistischen Umständen nur selten Modifikationen des Linienverlaufs erfolgen. Um die Effekte der Strategien besser untersuchen zu können, wird der Grenzwert der Abfahrtszeitabweichung  $a_{max}^{Diff}$ , der festlegt ab wann eine Linienverlaufsmodifikation in Frage kommt, deshalb für alle Untersuchungen auf einen unrealistisch niedrigen Wert von zehn Sekunden gesetzt. Durch diese Fixierung sind die Strategien zur Linienmodifikationen nur noch von zwei Parametern abhängig: Wird bei einer Modifikation die schnellste oder pünktlichste Route gewählt und wird der Linienverlauf nur bei erwarteter Verspätung oder auch bei erwarteter Frühzeitigkeit modifiziert. Tabelle 7.7 und Tabelle 7.8 zeigen die unter Anwendung dieser Strategien – mit und ohne zeitgleicher Verwendung ausgewählter Anti-Bunching Strategien – gemessenen Kennzahlen zu Bunching-Effekten bzw. Verspätung, Frühzeitigkeit und Wartezeit.

Strategie				Durch. kum. Sicherheitsabstandsreduktion [%]					
Linienmodifikation		Anti-Bunching		> 0	>= 10	>= 20	>= 30	>= 40	>= 50
Route	Zu Früh	Info.	$a_{max}^{Diff}$ [Sek.]						
-	-	-	-	24,72	9,52	2,25	0,41	0,08	0,01
S	Nein	-	-	23,34	7,50	1,77	0,28	0,04	0,01
P	Nein	-	-	24,36	8,01	1,68	0,26	0,03	0,00
S	Ja	-	-	23,06	5,85	0,95	0,08	0,01	0,00
P	Ja	-	-	23,55	6,28	1,11	0,13	0,01	0,00
S	Nein	Lokal	10	19,38	1,19	0,03	0,00	0,00	0,00
P	Nein	Lokal	10	19,92	2,28	0,22	0,02	0,00	0,00
S	Ja	Lokal	10	19,67	1,34	0,09	0,01	0,00	0,00
P	Ja	Lokal	10	19,93	2,27	0,23	0,02	0,00	0,00
S	Nein	Umgb.	10	19,70	1,23	0,04	0,00	0,00	0,00
P	Nein	Umgb.	10	20,15	2,26	0,23	0,02	0,00	0,00
S	Nein	Lokal	0	<b>17,36</b>	<b>0,40</b>	<b>0,01</b>	<b>0,00</b>	<b>0,00</b>	<b>0,00</b>
S	Nein	Umgb.	0	18,56	0,56	0,03	0,00	0,00	0,00

**Tabelle 7.7: Durchschnittlicher, kumulierter Anteil reduzierter Sicherheitsabstände unter unterschiedlichen Kombinationen von dynamischen Linienmodifikationen und Haltestrategien, UCL 2015. Erste Spalte: S – Schnellste Route; P – Pünktlichste Route. Zweite Spalte: Nein – Linienmodifikationen kommen nur bei erwarteter Verspätung zum Einsatz; Ja – Linienmodifikationen kommen auch bei erwarteter Frühzeitigkeit zum Einsatz. Dritte Spalte: Lokal - Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. - Die erwartete Pünktlichkeit am nächsten Haltepunkt wird zur Entscheidungsfindung herangezogen. Vierte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an**

Strategie				Verspätung [Sek.]		Frühzeitigkeit [Sek.]		Wartezeit [Sek.]	
Linienmodifikation		Anti-Bunching		Ø	Max.	Ø	Max.	Ø	Max.
Route	Zu Früh	Info.	$a_{max}^{Diff}$ [Sek.]						
-	-	-	-	4,4	84,6	14,1	104,8	<b>0,0</b>	<b>0,0</b>
S	Nein	-	-	<b>1,4</b>	<b>43,5</b>	18,1	118,6	<b>0,0</b>	<b>0,0</b>
P	Nein	-	-	2,8	56,8	14,4	103,0	<b>0,0</b>	<b>0,0</b>
S	Ja	-	-	<b>1,4</b>	43,6	14,5	94,5	<b>0,0</b>	<b>0,0</b>
P	Ja	-	-	2,8	57,9	12,2	73,5	<b>0,0</b>	<b>0,0</b>
S	Nein	Lokal	10	2,0	45,3	5,7	10,0	2,6	84,4
P	Nein	Lokal	10	3,3	60,0	5,3	10,0	1,6	27,9
S	Ja	Lokal	10	2,4	67,8	5,7	10,0	2,6	83,5
P	Ja	Lokal	10	3,3	58,7	5,3	10,0	1,6	27,7
S	Nein	Umgb.	10	2,0	44,9	5,8	16,2	2,5	84,6
P	Nein	Umgb.	10	3,3	60,4	5,3	14,9	1,6	27,6
S	Nein	Lokal	0	4,3	45,7	<b>0,0</b>	<b>0,0</b>	6,0	98,1
S	Nein	Umgb.	0	3,3	44,9	0,8	15,9	4,7	96,4

*Tabelle 7.8: Verspätung, Frühzeitigkeit und Wartezeit unter unterschiedlichen Kombinationen von dynamischen Linienmodifikationen und Haltestrategien, UCL 2015. Erste Spalte: S – Schnellste Route; P – Pünktlichste Route. Zweite Spalte: Nein – Linienmodifikationen kommen nur bei erwarteter Verspätung zum Einsatz; Ja – Linienmodifikationen kommen auch bei erwarteter Frühzeitigkeit zum Einsatz. Dritte Spalte: Lokal - Nur die Abweichung am aktuellen Kontrollpunkt wird zur Entscheidungsfindung herangezogen; Umgb. - Die erwartete Pünktlichkeit am nächsten Haltepunkt wird zur Entscheidungsfindung herangezogen. Vierte Spalte: Welchen Wert nimmt die maximal zulässige Abfahrtszeitabweichung an*

Zunächst lässt sich beobachten, dass die alleinige Anwendung von Strategien zur Linienmodifikation, im Vergleich zum unkontrollierten Fall, stets in einer Reduktion von Bunching-Effekten mündet. Dabei fällt die Reduktion unter Verwendung der schnellsten Route stärker aus als unter Verwendung der pünktlichsten Route. Außerdem sind die gemessenen Bunching-Effekte geringer wenn der Linienverlauf auch bei erwarteter Frühzeitigkeit modifiziert wird. Beide Effekte sind zu erwarten. Die Verwendung der schnellsten Route hat zur Folge, dass die Wahl der zu verwendenden Alternative für alle von einer Linienmodifikation betroffenen Fahrzeuge eindeutig ist: Es wird stets die 60-Sekunden-Route gewählt. Dies resultiert wiederum in einer geringeren Streuung der Fahrzeiten und der von diesen abhängenden Haltezeiten. Im Gegensatz dazu ist die Routenwahl bei Verwendung der pünktlichsten Alternative nicht eindeutig. Zwei aufeinander folgende Fahrzeuge können in diesem Fall, abhängig von ihrer erwarteten Abfahrtszeitabweichung, unterschiedliche Routen zugewiesen bekommen, wodurch eine erhöhte Streuung der Fahrzeiten und stärkere Bunching-Effekte zu erwarten sind.

Interessant ist, dass die alleinige Modifikation des Linienverlaufs für eine ähnlich starke Reduktion der Bunching-Effekte sorgt, wie die alleinige Anwendung von Haltestrategien an den Starthaltepunkten des Netzes (vgl. Tabelle 7.2). Dabei fällt die durchschnittliche und maximale Verspätung bei der Linienmodifikation, wie zu erwarten, wesentlich geringer aus. Dies geht allerdings zu Lasten der Frühzeitigkeit, die insbesondere bei den Maximalwerten deutlich höher ausfällt.

Um diesem Effekt entgegenzuwirken, werden die Strategien zur Linienmodifikation mit Anti-Bunching Strategien kombiniert. Dabei werden zunächst Kombinationen mit der im vorhergehenden Abschnitt ausgewogensten Strategie untersucht. D.h. zusätzlich zur Linienmodifikation wird die Verzögerung von Fahrzeugen an allen Haltepunkten unter Verwendung lokaler Informationen gestattet, falls ein Fahrzeug mehr als zehn Sekunden zu früh abfahren würde. Wie Tabelle 7.7 und Tabelle 7.8 zeigen, kann durch diese Kombinationen eine deutliche Reduktion des Anteils reduzierter Sicherheitsabstände sowie der Verspätung und Frühzeitigkeit erzielt werden. Im Vergleich zur alleinigen Anwendung der Haltestrategie kann der Gesamtanteil reduzierter Sicherheitsabstände noch mal um bis zu 1,18 Prozent reduziert werden. Insbesondere kann aber der Anteil leicht und stärker reduzierter Sicherheitsabstände (Reduktion um mindestens zehn Prozent) um bis zu 2,17 Prozent reduziert werden. Gleichzeitig kann die durchschnittliche und maximale Verspätung im besten Fall, verglichen mit der alleinigen Anwen-

Linie		Verspätung [Sek.]		
		Ohne SFM	Mit SFM	Basisfall
1-T01	Vollst.	6.311,2	43,7	7,2
	Bis NEU	-	12,2	-
	Ab HEU	-	8,9	-
1-T02	Vollst.	3.987,9	42,6	5,3
	Bis HEU	-	5,4	-
	Ab NEU	-	10,8	-
3-T01		2,5	2,6	2,5
3-T02		10,2	10,2	10,7
4-T01		6,5	6,4	6,7
4-T02		2,7	2,6	2,6
5-T01		2,6	2,6	2,6
5-T02		2,6	2,6	2,7
6-T01		3,8	3,8	3,9
6-T02		14,4	14,1	15,5
7-T01	Vollst.	5.719,1	8,8	1,6
	Bis NEU	-	1,7	-
	Ab HEU	-	4,9	-
7-T02	Vollst.	4.197,6	7,6	1,3
	Bis HEU	-	1,7	-
	Ab NEU	-	5,5	-
8-T01	Vollst.	5.557,3	13,5	1,2
	Bis HEU	-	0,0	-
	Ab NEU	-	8,0	-
8-T02	Vollst.	7.091,9	21,3	1,2
	Bis NEU	-	1,6	-
	Ab HEU	-	0,0	-
9-T01	Vollst.	4.677,6	14,7	3,3
	Bis NEU	-	2,3	-
	Ab HEU	-	4,8	-
9-T02	Vollst.	6.220,9	21,5	3,2
	Bis HEU	-	5,2	-
	Ab NEU	-	0,6	-
12-T01		4,6	4,6	4,7
12-T02		2,0	2,0	2,0
13-T01		8,1	5,1	4,6
13-T02		4,4	3,6	3,6
15-T01		8,6	8,6	8,7
15-T02		18,3	18,3	20,5
16-T01		5,0	5,0	5,1
16-T02		3,3	3,3	3,3
17-T01		16,4	16,1	17,3
17-T02		1,7	1,7	1,6
18-T01		2,5	2,5	2,5
18-T02		4,2	4,2	4,5
19-T01		1,8	1,8	1,8
19-T02		3,4	3,4	3,4
∅		2.738,1	9,6	4,5

Tabelle 7.9: Durchschnittliche Verspätung pro Linie, mit und ohne Störfallmanagement (SFM), KVB-T 2001

Neumarkt (NEU) und Heumarkt (HEU) im Stadtzentrum zwischen neun und zehn Uhr morgens. Als Resultat, kann in diesem Zeitraum kein Betrieb auf der Verbindung stattfinden. Die Wahl zur Blockade genau dieser Verbindung erfolgt aus zwei Gründen:

1. Trotz ihrer zentralen Lage im Stadtbahnnetz, beeinträchtigt eine Störung an diesem Ort nur die Liniengruppen 1, 7, 8 und 9, die wiederum relativ isoliert auf der West-Ost-Achse des Netzes

der Haltestrategie, halbiert werden, während die durchschnittliche (maximale) Frühzeitigkeit, verglichen mit der alleinigen Linienmodifikation, um bis zu 68,5 Prozent (91,6 Prozent) reduziert werden kann.

Die stärkste Reduktion von Bunching-Effekten resultiert von einer Kombination, bei der bei erwarteter Verspätung die schnellste Route gewählt wird und auf Basis lokaler Informationen eine exakt pünktliche Abfahrt erzwungen wird. In diesem Fall beträgt der Gesamtanteil reduzierter Sicherheitsabstände nur 17,36 Prozent, nahezu kein Sicherheitsabstand weist eine Reduktion von über zehn Prozent auf und kein einziger Sicherheitsabstand ist mehr als 30 Prozent reduziert. Gleichzeitig unterscheidet sich die durchschnittliche Verspätung in diesem Fall nicht signifikant von der durchschnittlichen Verspätung im unkontrollierten Fall. Allerdings fällt die maximale Verspätung 46,0 Prozent geringer aus. Diese Effekte werden dabei durch eine deutlich erhöhte (maximale) Wartezeit erkauft.

Es lässt sich festhalten, dass die dem Dispatcher zur Verfügung stehenden Verfahren der Haltepunkt-Kontrolle die erwartete Funktionalität und plausibles Verhalten aufweisen. Sowohl bei einem Störfall als auch im Regelbetrieb kann – die entsprechenden Netzressourcen vorausgesetzt – durch die (kombinierte) Anwendung von Haltestrategien und Linienmodifikationen die Servicequalität, im Vergleich zum unkontrollierten Fall, z.T. deutlich erhöht werden.

### 7.3.2. Störfallmanagement

#### 7.3.2.1. Auftrennen

Um die Funktionalität und Plausibilität des Störfallmanagements zu untersuchen, wird ein Störfall im Kölner Stadtbahnnetz unter dem bereits in Abschnitt 5.3.2 besprochenen Optimalfahrplan KVBT2001<sup>+</sup> simuliert. Alle im vorhergehenden Abschnitt beschriebenen Simulationsparameter werden ebenso beibehalten, wie die angewendete Haltestrategie, bei der frühzeitige Fahrzeuge an jedem Haltepunkt zu einer pünktlichen Abfahrt erzwungen werden.

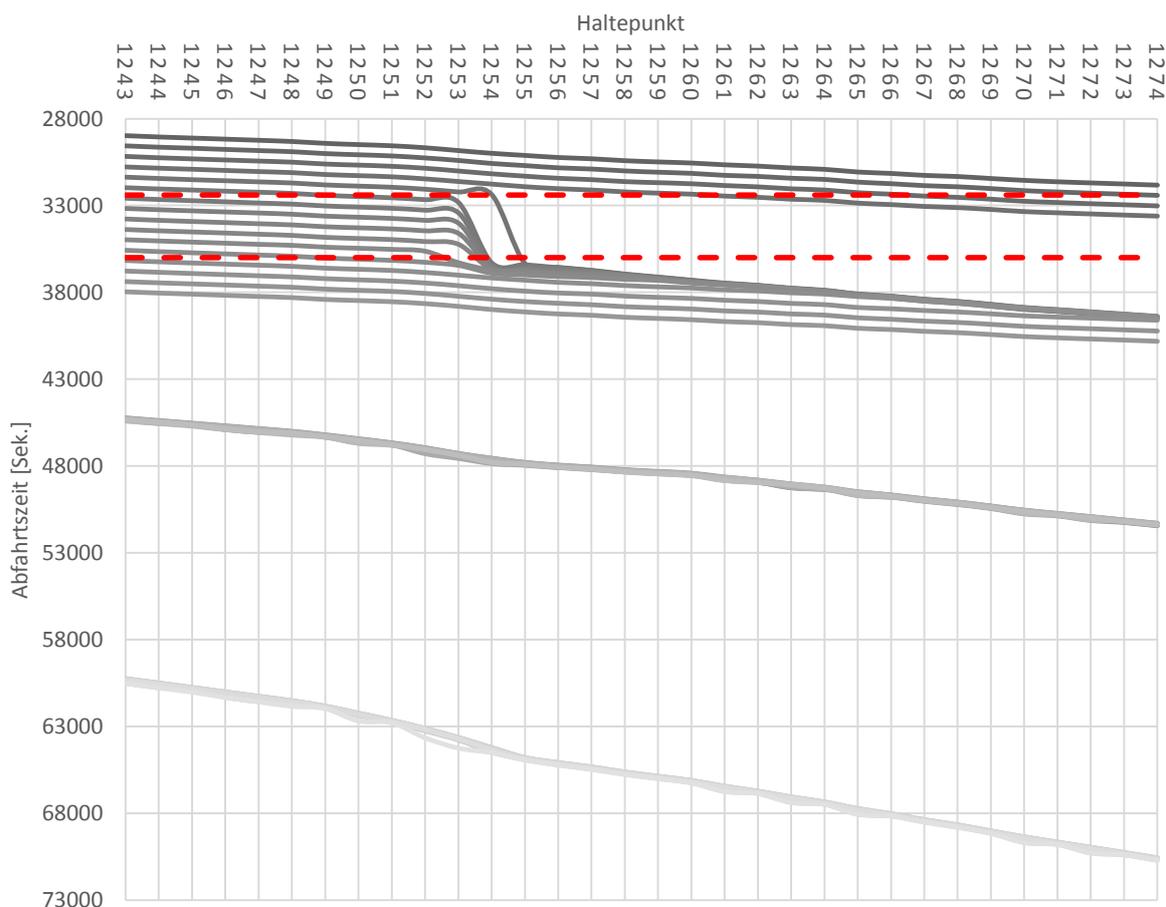
Das simulierte Störfallszenario umfasst auch in diesem Fall nur eine Störung: Die einseitige Blockade der Verbindung zwischen den Haltepunkten

verkehren. Die Effekte der Störung sollten sich entsprechend vornehmlich auf die direkt betroffenen Liniengruppen beschränken, wodurch eine einfachere Analyse möglich ist.

- Die Haltestellen an beiden Seiten der Verbindung bieten Wendemöglichkeiten, so dass die Linienverläufe der betroffenen Linien an diesen Positionen durch den Dispatcher aufgetrennt werden können.

Tabelle 7.9 listet die im Simulationszeitraum zwischen acht Uhr morgens und sechs Uhr abends erhobene durchschnittliche Linierverspätung mit und ohne Einsatz von Störfallmanagement auf und stellt sie dem ungestörten Basisfall (vgl. Tabelle 5.9) gegenüber. Wie erhofft, zeigen die Kennzahlen, dass sich die Effekte der Störung, auch ohne Störfallmanagement, weitestgehend auf die direkt betroffenen Liniengruppen beschränken. Diese werden dabei ohne explizite Adressierung der Störung, mit durchschnittlichen Verspätungen zwischen ca. einer und zwei Stunden, extrem in ihrem Betriebsablauf gestört. Wird hingegen durch das Auftrennen des Linienverlaufs auf die Störung reagiert können ihre Effekte zwar nicht vollständig eliminiert werden, schlagen sich aber mit Linierverspätungen von maximal ca. 44 Sekunden wesentlich moderater auf den Betriebsablauf nieder.

Dass der Dispatcher korrekt auf die Störung reagiert, zeigen Abbildung 7.3 und Abbildung 7.4. Dort sind beispielhaft für jeweils einen typischen Simulationslauf die Abfahrtszeiten aller im Messzeitraum simulierter Fahrten der Linie 1-T01 an allen Haltepunkten<sup>25</sup> des Linienverlaufs abgetragen. Zunächst zeigt sich vor Eintritt der Störung (obere gestrichelte Linie) sowohl mit als auch ohne Störfallmanage-



**Abbildung 7.3:** Abfahrtszeiten aller im Messzeitraum simulierter Fahrten der Linie 1-T01 (Richtung Bensberg) an den Haltepunkten der Linie, für einen typischen Simulationslauf ohne Störfallmanagement, KVB-T 2001. Gestrichelt: Start- bzw. Endzeitpunkt der Störung

<sup>25</sup> Zur Darstellung der Linienverläufe müssen die Haltepunkte hier anhand ihres eindeutigen numerischen Identifikators anstatt ihrer Abkürzung referenziert werden.

ment das gleiche Verhalten: Fahrzeuge sind ohne weitere Komplikationen in der Lage von ihrem Starthaltepunkt Junkersdorf (1243) bis zum Endhaltepunkt Bensberg (1274) zu fahren. Sobald die Störung aktiv wird unterscheidet sich das resultierende Systemverhalten jedoch signifikant.

Ohne Störfallmanagement werden alle Fahrten, die während der Störung den Haltepunkt Neumarkt (1254) erreichen, dort so lange verzögert bis die Störung aufgehoben wurde (untere gestrichelte Linie). Anschließend setzen die Fahrzeuge ihre Fahrten als Kolonne, ohne gleichmäßigen Abstand, fort. Auf diese Kolonne fahren dabei auch die erst nach der Störung startenden Fahrten auf, wodurch auch sie beeinträchtigt werden. Da die Fahrzeuge, auch durch die ungewollte Kolonnenfahrt, ihre Verspätung nicht mehr aufholen können, überträgt sie sich auf die Folgefahrten der Gegenrichtung, auf der das gleiche Phänomen zu beobachten ist. Ohne Eingriff von außen kann dieser Effekt nicht gebrochen werden, wodurch der Regelbetrieb zusammenbricht und nur noch sporadisch eine einzelne Kolonne von verspäteten Bahnen die Haltepunkte bedient.

Mit Störfallmanagement kann der Regelbetrieb hingegen zu großen Teilen aufrechterhalten werden. Dabei werden, wie Abbildung 7.4 zeigt, die Fahrten die den Haltepunkt Neumarkt während der Störung erreichen würden an diesem verkürzt. Die Bedienung der Haltepunkte hinter der Störung wird hingegen von den Fahrzeugen übernommen, die auf der entgegengesetzten Richtung nur bis zum Haltepunkt Heumarkt (1255) fahren. Zu sehen ist außerdem, dass auch mit Störfallmanagement kein durchgehender Betrieb möglich ist. Für ca. 1,5 Stunden starten keine Fahrten am Haltepunkt Junkersdorf, da die eigentlich diese Fahrten ausführenden Fahrzeuge auf der anderen Seite der Störung verkürzte Fahrten zwischen Bensberg und Heumarkt ausführen. Nach Ende der Störung und nachdem die Fahrzeuge mittels einer Regelfahrt der Gegenrichtung an der Haltestelle Junkersdorf angekommen sind, wird der Regel-

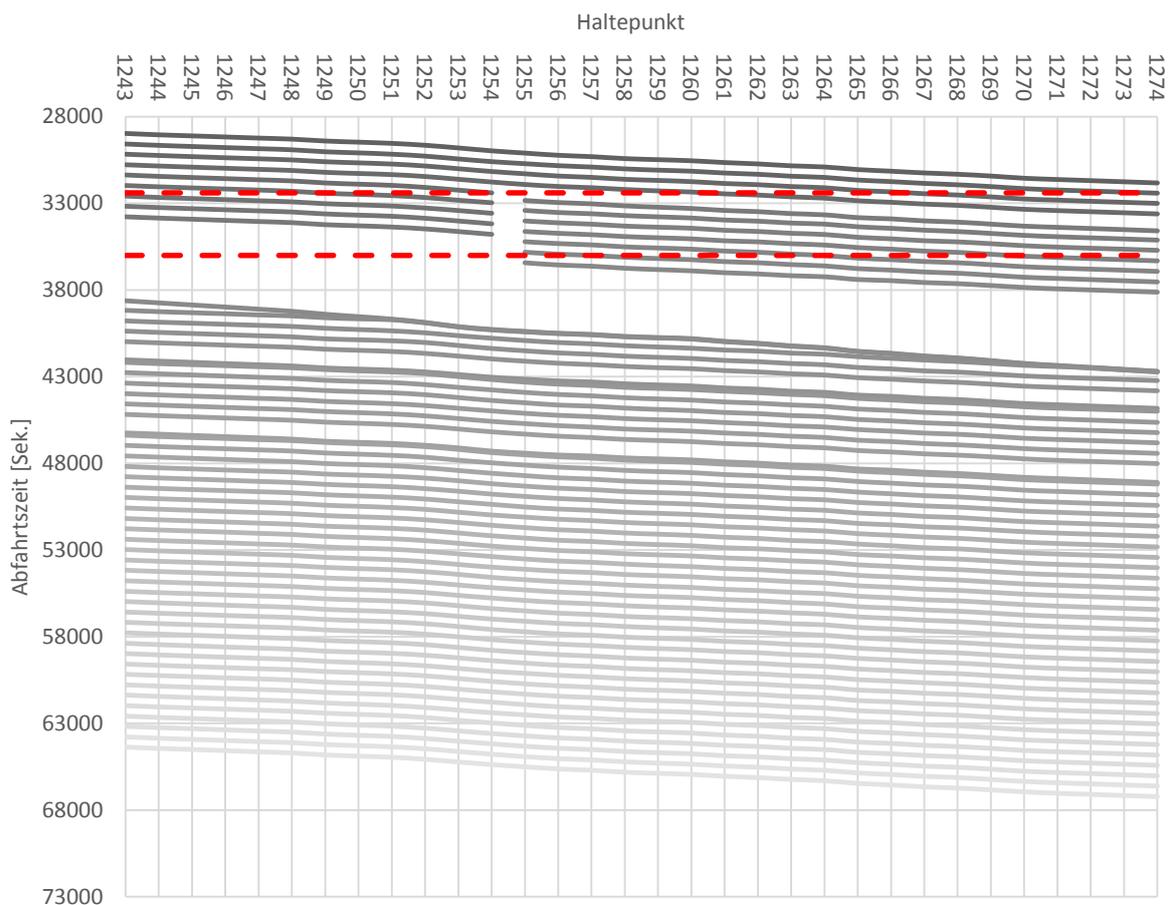


Abbildung 7.4: Abfahrtszeiten aller im Messzeitraum simulierter Fahrten der Linie 1-T01 (Richtung Bensberg) an den Haltepunkten der Linie, für einen typischen Simulationslauf mit Störfallmanagement, KVB-T 2001. Gestrichelt: Start- bzw. Endzeitpunkt der Störung

betrieb jedoch wieder aufgenommen. Dabei kommt es innerhalb der ersten drei Fahrten nach Wiederaufnahme, aufgrund der erhöhten Haltezeiten, zu einer Kolonnenbildung, deren Effekte sich ab ca. 13 Uhr (Simulationszeit 46970 in Abbildung 7.4) aufgelöst haben.

Es kann abschließend festgehalten werden, dass auch die Verfahren des Störfallmanagements die gewünschte Funktionalität erfüllen. Mit ihnen ist der Dispatcher in der Lage auf Störungen zu reagieren die den Regelbetrieb gefährden würden.

## 8. Anwendung auf das Gesamt-ÖPNV-Netz der Stadt Köln

Die bisherigen in dieser Arbeit durchgeführten Untersuchungen haben sich weitestgehend auf die Validierung der einzelnen Softwareanwendungen sowie deren Erweiterungen konzentriert und dabei z.T. spezifische, künstliche Beispiele für einzelne Verkehrsmodi herangezogen. Abschließend soll deswegen noch ein typischer Anwendungsfall für das gesamte Programmpaket für ein bi-modales Verkehrsnetz besprochen werden: Für das kombinierte Bus- und Stadtbahnnetz der Stadt Köln aus dem Jahr 2001 – bezeichnet mit KVB-BT 2001 – sollen zwei unterschiedliche Fahrpläne erzeugt und hinsichtlich ihrer Eignung zur Umsetzung gerichteter Umsteigeverbindungen zwischen den beiden Verkehrsmodi untersucht werden. Abfahrende Busfahrzeuge sollen dabei an ausgewählten Umsteigeknoten auf ankommende Bahnfahrzeuge warten, um umsteigenden Bahnpassagieren eine zügige Weiterfahrt zu gestatten.

Die beiden zu erzeugenden Fahrpläne unterscheiden sich dabei wie folgt: Der eine Fahrplan, bezeichnet mit KVB-BT<sup>+</sup>, wird ohne Berücksichtigung von verkehrsplanerischen Vorgaben erzeugt und stellt entsprechend die Optimallösung für die Serviceregularität des Gesamtsystems dar. Der andere Fahrplan, bezeichnet mit KVB-BT-C<sup>+</sup>, wird unter Berücksichtigung der in Tabelle 8.1 angegebenen Umsteigevorgaben und einer ausgeglichenen Gewichtung zwischen Serviceregularität und Erfüllung der Vorgaben – d.h.  $\alpha = 0,5$  – erzeugt.

Halte- stelle	Von	An	Zu	An	Abstands-Priorität			
					HIGH	MEDIUM	LOW	
Chorweiler	18-T01	1739	120-B01	9	0	1	2	3
	18-T01	1739	120-B02	49	0	1	2	3
	18-T01	1739	121-B01	85	0	1	2	3
	18-T01	1739	121-B02	123	0	1	2	3
	18-T01	1739	125-B01	186	0	1	2	3
	18-T01	1739	125-B02	193	0	1	2	3
	18-T01	1739	126-B02	231	0	1	2	3
Ostheim	9-T01	1549	9-T02	1562	0	-	-	-
	9-T01	1549	152-B01	840	0	-	-	-
	9-T02	1562	152-B01	840	0	-	-	-
	9-T01	1549	152-B02	891	0	-	-	-
	9-T02	1562	152-B02	891	0	-	-	-
	9-T01	1549	157-B01	1145	0	-	-	-
	9-T02	1562	157-B01	1145	0	-	-	-
Porz Markt	7-T01	1505	7-T02	1510	0	-	-	-
	7-T01	1505	152-B01	825	0	1	2	3
	7-T02	1510	152-B01	825	0	1	2	3
	7-T01	1505	154-B01	907	0	1	2	3
	7-T02	1510	154-B01	907	0	1	2	3
	7-T01	1505	160-B01	979	0	1	2	3
	7-T02	1510	160-B01	979	0	1	2	3
	7-T01	1505	161-B01	1023	0	1	2	3
	7-T02	1510	161-B01	1023	0	1	2	3
	7-T01	1505	162-B01	1036	0	1	2	3
7-T02	1510	162-B01	1036	0	1	2	3	

*Tabelle 8.1: Verkehrsplanerische Vorgaben zur Modellierung von gerichteten Umsteigeverbindungen an ausgewählten Haltestellen, Gesamt-ÖPNV-Netz Köln, 2001. Jede Zeile steht für bis zu drei Vorgaben, jeweils eine pro Priorität.*

Das Kölner Nahverkehrsnetz umfasst 1.770 Haltepunkte (1.242 für Busse, 528 für Bahnen), die von 68 Bus- bzw. 30 Bahnlinien bedient werden. Entsprechend bietet es zahlreiche potentielle Umsteigeknoten (vgl. Abbildung 3.3 bis Abbildung 3.6). Das Einrichten von Umsteigeverbindungen an allen diesen Punkten ist allerdings weder möglich noch zweckdienlich. Idealerweise sollten stattdessen anhand von Informationen über Passagierreisen – etwa in Form von Quell/Ziel-Matrizen – sowie betrieblichen und politischen Überlegungen die relevantesten Umsteigeknoten identifiziert und der Gesamtfahrplan unter Berücksichtigung dieser Informationen erstellt werden. In Ermangelung dieser Daten wird sich auf solche Orte des Netzes konzentriert, an denen die Einrichtung von Umsteigeverbindungen nicht oder zumindest nicht signifikant von Annahmen über die fehlenden Informationen abhängt. Für die Einrichtung von Umsteigeverbindungen kommen deswegen nur Haltestellen in Frage, an denen sich die Fahrzeuge der abfahrenden Linien wenn möglich keine gemeinsamen Ressourcen teilen. So wird vermieden, dass eine Wahl der Abfahrtsreihenfolge erfolgen muss, die mögliche Umsteigeverbindungen beeinflussen würde.

Aufgrund dieser Überlegungen, sind die interessantesten potentiellen Umsteigeknoten des Kölner Nahverkehrsnetzes die Busbahnhöfe Chorweiler, Ostheim und Porz Markt. Weitere prinzipiell relevante Umsteigeknoten, wie bspw. die Haltestellen Bocklemünd, Wiener Platz, Ebertplatz oder Bahnhof Deutz, werden hier hingegen nicht berücksichtigt, da an ihnen (mindestens) Annahmen über die Abfahrtsreihenfolge der abfahrenden Bahnlinien getätigt werden müssten. Diese Annahmen würden aber Informationen benötigen, die nicht von der zur Verfügung stehenden Datengrundlage gedeckt sind. Die drei ausgewählten Umsteigeknoten besitzen hingegen dedizierte Haltepunkte für alle abfahrenden Buslinien und – mit Ausnahme der Haltestelle Porz Markt – alle abfahrenden Bahnlinien. An der Haltestelle Porz Markt teilen sich zwar die Linien 7-T01 und 8-T01 sowie 7-T02 und 8-T02 gemeinsame Haltepunkte, da sich die Varianten der Liniengruppe 8 aber einen signifikanten Teil ihres Linienverlaufs mit den entsprechenden Varianten der Liniengruppe 7 teilen und letztere wesentlich längere Linienverläufe aufweisen, kann hier angenommen werden, dass Linien 7-T01 und 7-T02 für die Einrichtung von Umsteigeverbindungen relevanter sind.

Die drei ausgewählten Haltestellen bilden darüber hinaus einen interessanten Querschnitt über verschiedene mögliche Konstellationen von Start-/End- bzw. Durchfahrtshaltestellen: Die Haltestelle Chorweiler ist die End-/Starthaltestelle der Bahnlinien 18-T01/T02 sowie der Buslinie 126-B01/B02. Außerdem fahren an ihr die Buslinien 120-B01/B02, 121-B01/B02 und 125-B01/B02 ab. Die Haltestelle Ostheim ist hingegen eine Durchfahrtshaltestelle für die Bahnlinien 9-T01/T02 und die Buslinien 152-B01/B02, sowie die Start-/Endhaltestelle für die Buslinien 157-B01/B02. Die Haltestelle Porz Markt ist schließlich die Start-/Endhaltestelle der Buslinien 152-B01/B02, 154-B01/B02, 160-B01/B02, 161-B01/B02, 162-B01/B02 und der Bahnlinien 8-T02/T01, sowie eine Durchfahrtshaltestelle der Bahnlinien 7-T01/7-T02.

Zur Umsetzung der gewünschten gerichteten Umsteigeverbindungen sind mindestens 25 verkehrsplannerische Vorgaben notwendig (vgl. Spalte „HIGH“ in Tabelle 8.1). Idealerweise sollten diese Vorgaben eine zeitgleiche Abfahrt der betroffenen Linien erzwingen, wie dies für die Haltestelle Ostheim auch möglich ist. An den Haltestellen Chorweiler und Porz Markt ist dies jedoch nicht möglich, da die ankommenden Buslinien z.T. vor- und/oder nachgelagerte Haltepunkte gemeinsam haben, an denen die Optimierungsanwendung entsprechend keine zeitgleichen Abfahrten terminieren kann/darf. Aus diesem Grund muss der Optimierungsanwendung an diesen Haltestellen mehr Spielraum eingeräumt werden. Dies geschieht, indem 34 weitere Vorgaben mit entsprechend niedrigerer Priorität definiert werden, die Busfahrzeugen auch Abfahrten erlauben, die bis zu drei Minuten vor den Zubringerfahrzeugen des Bahnverkehrs stattfinden (vgl. Spalten „MEDIUM“ und „LOW“ in Tabelle 8.1).<sup>26</sup> An den Haltestellen Ostheim und Porz Markt wird außerdem die zeitgleiche Abfahrt der Hin- und Rückrichtung der dort abfahrenden Liniengruppen 9 bzw. 7 erzwungen, um Passagieren beider Richtungen den Umstieg in wartende Busse zu ermöglichen. Nicht berücksichtigt werden bei den Umsteigevorgaben Bus- und Bahnlinien deren Linienverlauf an der betroffenen Haltestelle enden resp. starten. So werden unnötige

<sup>26</sup> Der Grund für diese Art der Modellierung (anstatt einer Erzwingung späterer Abfahrten) wird in Kürze erläutert.

Umsteigeverbindungen zwischen ankommenden/abfahrenden Bahnen und ihre Fahrt beendenden Bussen vermieden.

Tabelle 8.2 listet die für beide Fahrpläne resultierenden Zielfunktionswerte sowie die durch die Optimierungsanwendung benötigte Laufzeiten auf. Bei beiden Fahrplänen handelt es sich um 12-Stunden-Fahrpläne von sieben Uhr morgens bis sieben Uhr abends, mit einem globalen Takt von 10 Minuten. Die Optimierungsläufe werden auf einem Intel Core i7-740QM Quad Core mit 1,73 GHz und 6 GB RAM ausgeführt.

Fahrplan	Zielfunktionswert	Regularität	Vorgaben	Laufzeit [Sek.]			Σ	Gap [%]
				Preprocessing	CPLEX	Postprocessing		
<b>KVB-BT<sup>+</sup></b>	45.026	45.026	-	11,1	604,1	72,1	687,3	-
<b>KVB-BT-C<sup>+</sup></b>	19.411	34.376	4.446	8,3	6.299,5	64,1	6.371,8	1,00

*Tabelle 8.2: Zielfunktionswerte und benötigte Laufzeiten mit und ohne Berücksichtigung von Umsteigevorgaben, KVB-BT 2001*

Ohne Berücksichtigung verkehrsplanerischer Vorgaben wird unter Anwendung des in Abschnitt 3.3.1 beschriebenen Zerlegungsverfahrens innerhalb von ca. 11,5 Minuten ein Optimalfahrplan mit einem netzwerkweiten Sicherheitsabstand von zwei Minuten und einem Zielfunktionswert von 45.026 erzeugt. Ohne Definition verkehrsplanerischer Vorgaben entfällt dieser Wert dabei vollständig auf den Service-regularitätsanteil der Zielfunktion.

Im Gegensatz dazu weist Fahrplan KVB-BT-C<sup>+</sup>, wie erwartet, nur einen netzwerkweiten Sicherheitsabstand von einer Minute und einen entsprechend reduzierten Zielfunktionswert von  $19.411 = 0,5 * 34.376 + 0,5 * 4.446$  auf. Die Erfüllung der Umsteigevorgaben hat also eine Reduzierung der Service-regularität von ca. 23,7 Prozent zur Folge. Dies ist zu erwarten, da durch die Umsteigevorgaben die Sicherheitsabstände zwischen den die Haltestellen Chorweiler und Porz Markt bedienenden Buslinien künstlich reduziert werden. Dies ist auch in Tabelle 8.3 zu sehen, die die Abfahrtszeiten der durch die Umsteigevorgaben betroffenen Linien an den drei ausgewählten Haltestellen auflistet. Unter Fahrplan KVB-BT-C<sup>+</sup> fahren alle Buslinien innerhalb des Taktes, wie gewünscht, vor den Bahnlinien ab, während sich die Abfahrten unter Fahrplan KVB-BT<sup>+</sup> sowohl vor als auch nach den Abfahrten der Bahnlinien verteilen. Zu sehen ist darüber hinaus, dass an den Haltestellen Chorweiler und Porz Markt jeweils zwei der verkehrsmittelübergreifenden Umsteigevorgaben optimal erfüllt sind, während vier bzw. zwei nur mit einer mittleren Priorität erfüllt werden können. Schließlich werden weitere zwei bzw. eine nur mit niedriger Priorität erfüllt.

Haltestelle	Chorweiler							Ostheim					Porz Markt							
	Fahrplan/ Linie	18-T01	120-B01	120-B02	121-B01	121-B02	125-B01	125-B02	126-B01	9-T01	9-T02	152-B01	152-B02	157-B01	7-T01	7-T02	152-B01	154-B01	160-B01	161-B01
<b>KVB-BT<sup>+</sup></b>	1	4	2	7	7	9	4	1	3	6	6	9	4	2	7	7	9	1	3	9
<b>KVB-BT-C<sup>+</sup></b>	9	8	9	9	8	6	7	7	7	7	7	7	7	9	9	8	9	7	9	6

*Tabelle 8.3: Abfahrtszeiten der durch Umsteigeverbindungen betroffenen Linien unter den zu vergleichenden Fahrplänen, gruppiert nach Haltestellen, KVB-BT 2001*

Die Erzeugung von Fahrplan KVB-BT-C<sup>+</sup> benötigt im Vergleich zur Erzeugung von Fahrplan KVB-BT<sup>+</sup> nahezu die zehnfache Laufzeit. Grund dafür ist, dass die definierten Umsteigevorgaben Linien unterschiedlicher Zusammenhangskomponenten des Streckennetzes umfassen, weswegen keine getrennte Optimierung dieser Teilnetze mehr erfolgen kann. Eine weitere Folge dieses Umstandes ist, dass die Optimierungsanwendung die Optimalität des erzeugten Fahrplans nicht mehr in vertretbarer Laufzeit sicherstellen kann. Die Optimierung wurde deswegen frühzeitig abgebrochen, sobald die Optimierungsanwendung sicherstellen konnte, dass zwischen der besten bis dato berechneten ganzzahligen Lösung und der besten berechneten oberen Schranke nur noch ein relativer Abstand von einem Prozent liegt.

Beobachtungen mehrerer weiterer Optimierungsläufe lassen aber vermuten, dass es sich bei der ausgewiesenen Lösung um die Optimallösung handelt, da CPLEX nach Berechnung dieser Lösung nur noch Verbesserungen der oberen Schranke verzeichnet.<sup>27</sup>

Die Ergebnisse der Optimierung lassen vermuten, dass Fahrplan KVB-BT-C<sup>+</sup> besser zur Umsetzung der definierten Umsteigeverbindungen geeignet ist als Fahrplan KVB-BT<sup>+</sup>. Die Abfahrtszeiten der relevanten Bus- und Bahnlinien unter beiden Fahrpläne lassen erwarten, dass ohne Eingriff des Dispatchers, d.h. ohne künstliches Verzögern von Busfahrzeugen, unter Fahrplan KVB-BT<sup>+</sup> durchschnittlich drei von 23 Umsteigeverbindungen (ca. 13%) gehalten werden können, während unter Fahrplan KVB-BT-C<sup>+</sup> durchschnittlich 12 von 23 (ca. 52,2%) Abfahrten innerhalb des Taktintervalls zeitgleich stattfinden. Ob dies tatsächlich der Fall ist, insbesondere unter dynamischen Bedingungen, soll im Folgenden mit Hilfe der Simulationsanwendung überprüft werden.

Für das Stadtbahnnetz werden die gleichen Simulationsparameter wie in Abschnitt 5.3.2 herangezogen, d.h. bei einer angenommenen Zustiegszeit von drei Sekunden pro Passagier und einem Takt von zehn Minuten, werden die Ankunftsdaten aller Haltepunkte so gesetzt, dass  $T_{f,s}^{Bahn} = 20$  Sekunden gilt. Die zur Berechnung der Fahrzeiten verwendeten Parameter werden wiederum auf  $\gamma^{Bahn} = 0,75$ ,  $\eta^{Bahn} = 0,1$  und  $\iota^{Bahn} = 0$  gesetzt. Darüber hinaus fungieren alle Haltepunkte des Bahnnetzes als Kontrollpunkte an denen eine pünktliche Abfahrt erzwungen wird.

Auch für das Busnetz wird eine Zustiegszeit von drei Sekunden pro Passagier angenommen und die Ankunftsdaten aller Haltepunkte so gesetzt, dass  $T_{f,s}^{Bus} = 20$  Sekunden gilt. Die in der Datenbasis vorhandenen Verbindungen des Kölner Busnetzes weisen eine durchschnittliche geplante Fahrzeit von ca. 94 Sekunden auf. Entsprechend wird  $\gamma^{Bus} = 1 - \left(\frac{20}{94}\right) \approx 0,79$  gesetzt. Da der Individualverkehr stärkeren Schwankungen der Fahrzeit als der Bahnverkehr unterliegt wird nach einer stichpunktartigen Sensitivitätsanalyse  $\eta^{Bus} = 0,21$  gesetzt. Außerdem werden leichte Fahrzeitschwankungen aufgrund von Lichtsignalanlagen – ebenfalls nach einer stichpunktartigen Sensitivitätsanalyse – abgebildet, indem  $\iota = 0,01$  gesetzt wird, d.h. pro Signalanlage auf einer Verbindung wird die Standardabweichung der Fahrzeit um ein Prozent erhöht. Schließlich werden noch die Starthaltepunkte aller Linien sowie die Haltepunkte, die Verbindungspunkte zum Fernverkehr darstellen (d.h. 128 der 1.242 Haltepunkte des Busnetzwerks, vgl. Abbildung 3.3 bis Abbildung 3.6) als Kontrollpunkte definiert. An diesen Kontrollpunkten wird die in Abschnitt 7.3.1.1 als am geeignetsten identifizierte Haltestrategie implementiert: Fahrzeuge dürfen, auf Basis lokaler Informationen, maximal zehn Sekunden vor ihrer planmäßigen Abfahrtszeit losfahren.

Unter diesen Parametern wird anhand von jeweils sechs Experimenten mit je 100 Simulationsläufen die Eignung beider Fahrpläne zur Umsetzung von Umsteigeverbindungen überprüft. Dazu wird die Pro-Synchronisierung Strategie aus Algorithmus 7.2 unter systematischer Variation der Parameter  $w_{t_c}$  und  $t_t$  zur Einhaltung der definierten Umsteigeverbindungen verwendet und die resultierenden Ergebnisse, soweit möglich, mit Kennzahlen des Basisfalls ohne Umsteigeverbindungen (gekennzeichnet durch die Parameterwerte „-,“) verglichen. Zur Vermeidung von Verzerrungen aufgrund von Ein- bzw. Ausschwingphasen, werden Simulationskennzahlen abermals nur zwischen den Simulationszeitpunkten acht Uhr morgens und sechs Uhr abends erfasst.

Die Wartezeit  $w_{t_c}$  wird in 60-Sekunden-Schritten von 0 bis 180 variiert, da die verkehrsplanerischen Vorgaben, unter denen Fahrplan KVB-BT-C<sup>+</sup> erzeugt wurde, Busfahrzeuge planmäßig bis zu drei Minuten vor dem Zubringerfahrzeug des Bahnverkehrs abfahren lassen. Um die Umsteigeverbindung zu halten, müssen sie bei pünktlicher Ankunft also bis zu drei Minuten auf das Bahnfahrzeug warten. Zusätzlich wird auch noch eine kürzere Wartezeit von 30 Sekunden getestet. Für die Umsteigezeit  $t_t$ , die Fahrzeuge nach Erfüllung einer Umsteigeverbindung bis zur eigentlichen Abfahrt warten, werden Werte von 0 sowie 30 Sekunden verwendet. Ersterer wird dabei nur zusammen mit einer Wartezeit von  $w_{t_c} = 0$  verwendet, um die inhärente Eignung der Fahrpläne zur Umsetzung von Umsteigeverbindungen, d.h. ohne (signifikanten) Eingriff des Dispatchers, zu erfassen. Der Wert von 30 Sekunden für  $t_t$  wurde

<sup>27</sup> Im Zuge dieser Beobachtungen konnte CPLEX durch die Verbesserung der oberen Schranke den relativen Abstand auf bis zu 0,5 Prozent reduzieren, benötigte dazu aber weitere ca. vier Stunden Laufzeit.

gewählt, damit eventuelle (künstliche) Verspätungen aufgrund dieses Wertes die Simulationskennzahlen nicht zu stark beeinflussen. Im Realsystem würden wartende Busse losfahren, sobald alle umsteigenden Passagiere das Fahrzeug betreten haben. In Ermangelung von Passagierzahlen, wird ein plausibler (Durchschnitts-)Wert gewählt, der der Mehrheit der Passagiere einen komfortablen Umstieg ermöglichen sollte. Die Zeitdauer  $\tau_q$ , die Busfahrzeuge bis zur erneuten Abfrage der Erfüllung der Umsteige Verbindung warten, wird für alle Experimente einheitlich auf 30 Sekunden gesetzt.

Tabelle 8.4 listet den im Messzeitraum von den unterschiedlichen Buslinien an den drei Haltestellen unter den unterschiedlichen Fahrplänen und Parametern erreichten prozentualen Anteil erfüllter Umsteige Verbindungen auf. Dies sind die Abfahrten der betroffenen Buslinien während denen die relevanten Zubringerfahrzeuge der Bahnlinien innerhalb des angegebenen Zeitintervalls eintreffen.

Fahrplan	KVB-BT <sup>+</sup>					KVB-BT-C <sup>+</sup>					
	wt <sub>c</sub> [Sek.]	0	30	60	120	180	0	30	60	120	180
$\tau_t$ [Sek.]	0	30	30	30	30	30	0	30	30	30	30
Chorweiler	120-B01	0,00	0,00	0,00	0,00	0,00	0,00	97,38	<b>98,33</b>	<b>98,33</b>	<b>98,33</b>
	120-B02	0,77	0,72	0,92	0,82	0,73	63,37	63,48	63,78	63,48	<b>64,32</b>
	121-B01	0,00	0,00	0,00	0,00	0,00	69,97	<b>70,53</b>	68,56	68,30	68,23
	121-B02	0,00	0,00	0,00	0,00	0,00	0,07	97,38	<b>98,33</b>	<b>98,33</b>	<b>98,33</b>
	125-B01	13,44	12,57	12,05	98,28	98,06	0,00	0,00	0,00	0,00	<b>98,32</b>
	125-B02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	<b>98,33</b>	<b>98,33</b>
Ostheim	126-B02	96,67	96,67	96,67	96,67	96,67	0,00	0,00	0,02	<b>98,33</b>	<b>98,33</b>
	152-B01	49,88	48,91	37,50	0,88	0,00	<b>99,95</b>	98,78	49,13	46,86	5,22
	152-B02	0,00	0,00	0,00	0,00	0,00	99,98	<b>100,00</b>	99,99	<b>100,00</b>	<b>100,00</b>
Porz	157-B01	0,00	0,00	0,00	49,92	49,03	99,70	<b>99,80</b>	99,73	95,96	83,00
	152-B01	50,00	50,00	50,00	50,00	50,00	0,00	26,26	<b>100,00</b>	<b>100,00</b>	<b>100,00</b>
	154-B01	0,00	0,00	0,00	0,00	50,00	<b>100,00</b>	<b>100,00</b>	<b>100,00</b>	<b>100,00</b>	<b>100,00</b>
	160-B01	0,00	30,36	50,00	50,00	50,00	0,00	0,00	0,00	<b>100,00</b>	<b>100,00</b>
	161-B01	0,00	0,00	0,00	0,00	0,00	<b>100,00</b>	<b>100,00</b>	<b>100,00</b>	<b>100,00</b>	<b>100,00</b>
$\emptyset$	14,05	15,95	16,48	23,10	29,63	42,20	56,91	58,53	77,86	<b>87,49</b>	

Tabelle 8.4: Durchschnittlicher prozentualer Anteil erfüllter Umsteige Verbindungen an den unterschiedlichen Haltestellen in Abhängigkeit des Fahrplans und der Warte-/Umsteigezeit, KVB-BT 2001

Zunächst wird sichtbar, dass Fahrplan KVB-BT-C<sup>+</sup>, wie erwartet, bereits ohne signifikanten Eingriff des Dispatchers besser zur Umsetzung der Umsteige Verbindungen geeignet ist als Fahrplan KVB-BT<sup>+</sup>. Unter letzterem werden inhärent nur durchschnittlich 14,05 Prozent aller potentieller Umsteige Verbindungen erfüllt, während unter Fahrplan KVB-BT-C<sup>+</sup> durchschnittlich 42,20 Prozent aller Umsteige Verbindungen erfüllt werden. Die konkreten Durchschnittswerte überraschen dabei zunächst leicht: Während Fahrplan KVB-BT<sup>+</sup> den für ihn theoretisch erwarteten Wert um ca. ein Prozent übertrifft, fällt der Anteil erfüllter Umsteige Verbindungen unter Fahrplan KVB-BT-C<sup>+</sup> zehn Prozent geringer aus als der theoretisch erwartete Wert. Hier zeigen sich die Auswirkungen der dynamischen Bedingungen, die durch die Optimierungsanwendung nicht berücksichtigt werden. Wie Tabelle 8.5 zeigt, weist Linie 125-B01 bereits ohne Berücksichtigung von Umsteige Verbindungen an der Haltestelle Chorweiler eine durchschnittliche Verspätung von 50 bzw. ca. 63 Sekunden auf. Da diese Linie unter Fahrplan KVB-BT<sup>+</sup> planmäßig zwei Minuten vor Bahnlinie 18-T01 in Chorweiler abfährt, ist zu vermuten, dass einige ihrer im Simulationsverlauf stattfindenden Abfahrten – nämlich durchschnittlich 13,44 Prozent – sogar so stark verspätet stattfinden dass sie unbeabsichtigt dazu in der Lage sind die Umsteige Verbindung zu erfüllen. Unter Fahrplan KVB-BT-C<sup>+</sup> finden die Abfahrten von Linie 125-B01 hingegen drei Minuten vor den Abfahrten der Linie 18-T01 statt, sodass die starken Verspätungen keine unerwarteten Auswirkungen auf diese Umsteige Verbindungen haben. Unter diesem Fahrplan schlägt sich das gleiche Phänomen stattdessen negativ auf den Anteil der erfüllten Umsteige Verbindungen der Linien 120-B01/B02 nieder: Ihre Fahrzeuge sollen unter Fahrplan KVB-BT-C<sup>+</sup> zwar zeitgleich mit denen der Linie 18-T01 an der Haltestelle Chorweiler abfahren, sind durchschnittlich aber minimal verzögert (vgl. Tabelle 8.5). Da

die Zubringerfahrzeuge der Linie 18 ihre Fahrt allerdings im Durchschnitt pünktlich beenden, werden nur ca. 63 bis 70 Prozent der Umsteigeverbindungen gehalten.<sup>28</sup> Wie Tabelle 8.4 zeigt, hat in diesem Fall erwartungsgemäß auch eine Erhöhung der maximalen Wartezeit keinen Effekt.

Wird der Dispatcher durch eine Erhöhung der maximalen Wartezeit  $w_{t_c}$  zu stärkeren Eingriffen veranlasst, kann der Anteil erfüllter Umsteigeverbindungen unter Fahrplan KVB-BT<sup>+</sup> auf durchschnittlich bis zu 29,63 Prozent erhöht werden. Unter Fahrplan KVB-BT-C<sup>+</sup> kann der durchschnittliche Anteil erfüllter Umsteigeverbindungen hingegen auf bis zu 87,49 Prozent erhöht werden. Dieser Effekt ist zu erwarten, da unter Fahrplan KVB-BT-C<sup>+</sup> erzwungen wird, dass die Abfahrten der Buslinien an den drei Umsteigeknoten innerhalb der drei Minuten vor den Abfahrtszeiten der betroffenen Bahnlinien stattfinden. Durch eine bewusste Verzögerung der Fahrzeuge der Buslinien um bis zu drei Minuten während der Simulation muss folglich die Wahrscheinlichkeit erhöht werden, dass die Umsteigeverbindungen im Regelbetrieb erfüllt werden. Unter Fahrplan KVB-BT<sup>+</sup> hat die Anwendung dieser Strategie hingegen keinen so ausgeprägten Effekt, da die Synchronisationen zwischen den Abfahrten der Bus- und Bahnlinien ohne die Berücksichtigung verkehrsplanerischer Vorgaben weitestgehend zufällig erfolgt. Dass auch unter Fahrplan KVB-BT-C<sup>+</sup> dennoch nicht alle Umsteigeverbindungen erfüllt werden können, ist auch hier mehrheitlich durch die aufgrund der dynamischer Bedingungen schwankenden Fahr- und Haltezeiten zu erklären.

Fahrplanunabhängig kann in Tabelle 8.4 außerdem beobachtet werden, dass der Anteil erfüllter Umsteigeverbindungen für Linie 152-B01 an der Haltestelle Ostheim mit zunehmender maximaler Wartezeit deutlich sinkt. Der Grund hierfür ist an der Haltestelle Porz Markt zu finden: Sowohl unter Fahrplan KVB-BT<sup>+</sup> als auch KVB-BT-C<sup>+</sup> starten Fahrzeuge von Linie 152-B01 dort nicht zeitgleich mit allen relevanten Bahnfahrzeugen der Liniengruppe 7 (vgl. Tabelle 8.3). Unter Fahrplan KVB-BT<sup>+</sup> starten sie zeitgleich mit Fahrzeugen der Linie 7-T02 und fünf Minuten vor Fahrzeugen der Linie 7-T01, sodass sie unter ihm stets nur 50 Prozent aller Umsteigeverbindungen halten können. Im Versuch auch die Umsteige Verbindung zu Linie 7-T01 zu halten, warten sie vergeblich bis die zulässige Höchstwartezeit erreicht wurde (vgl. Tabelle 8.5 und Tabelle 8.6). Unter Fahrplan KVB-BT-C<sup>+</sup> starten sie hingegen eine Minute vor Fahrzeugen der Liniengruppe 7. Entsprechend müssen sie unter ihm mindestens eine Minute auf die Zubringerfahrzeuge warten. Da sich Linie 152-B01 zugleich bis zur Haltestelle Gremberghoven, S-Bahn (vgl. Abbildung 3.6) einen Großteil ihrer Route abwechselnd mit den Linien 154-B01, 160-B01, 161-B02 und 162-B02 teilt, kommt es an den gemeinsam genutzten Ressourcen zu Kapazitätsengpässen und Bunching-Effekten. Dieser Effekt wird stärker, je stärker auch die Fahrzeuge der anderen Linien an der Haltestelle Porz Markt verzögert werden und resultiert in einer Verspätung der Fahrzeuge der Linie 152-B01 an der Haltestelle Ostheim. In der Folge können die Umsteigeverbindungen dort wiederum nicht gehalten werden.

Ein ähnlicher Effekt mit anderer Ursache ist unter Fahrplan KVB-BT-C<sup>+</sup> für Linie 157-B01 zu beobachten: Vereinzelt Fahrten dieser Linie starten aufgrund von Fahrzeitschwankungen der Rückrichtung leicht verzögert an der Haltestelle Ostheim und verpassen dort ihr Zubringerfahrzeug. In der Folge warten die betroffenen Fahrzeuge vergeblich auf das nächste Zubringerfahrzeug und verzögern sich um den Wert von  $w_{t_c}$ . Bis zu einer Wartezeit von 60 Sekunden hat dies keine signifikanten Auswirkungen auf den restlichen Verlauf dieser Fahrten. Ab einem Wert von  $w_{t_c} = 120$  Sekunden sind die betroffenen Fahrzeuge jedoch nicht mehr in der Lage die Eingangsverspätung im Verlauf ihrer Fahrt aufzuholen, sodass sich die Verspätung auf ihre Folgefahrt überträgt. Auch auf dieser Fahrt sind sie nicht in der Lage

---

<sup>28</sup> Es handelt sich hierbei zugegebenermaßen um einen Grenzfall. Es kann begründet argumentiert werden, dass eine Umsteige Verbindung auch als gehalten gelten kann, wenn das Zubringerfahrzeug die Haltestelle kurz vor dem Zielfahrzeug verlässt. Da es nach Kenntnis des Autors aber keine Veröffentlichung darüber gibt, wieviel Zeit verstreichen muss bis ein Passagier eine Umsteige Verbindung als gescheitert betrachtet und die zugrunde liegende Datenbasis keine Informationen diesbezüglich beinhaltet, wird der Einfachheit halber angenommen, dass jegliches Verpassen des Zubringerfahrzeugs in einer nicht gehaltenen Umsteige Verbindung mündet. Eine zukünftige Modifikation der Pro-Synchronisierung Strategie bzw. der zugehörigen Datenerfassung sollte jedoch ohne weiteres möglich sein.

die Verspätung gutzumachen, weswegen sie mit erhöhter Wahrscheinlichkeit auch ihre nächste Umsteige-  
 ge Verbindung verpassen. Während sich dieser Effekt bei einer maximalen Wartezeit von  $w_{t_c} = 120$   
 Sekunden noch nicht signifikant in den Durchschnittswerten für den Anteil Erfüllter Umsteige-  
 ge Verbindungen, Verspätung und Wartezeit niederschlägt, wird ab einer Verzögerung von 180 Sekunden ein deutlicher  
 Einbruch des Anteils der erfüllten Umsteige-  
 ge Verbindungen und ein Anstieg in der durchschnittlichen  
 Verspätung und Wartezeit sichtbar. Der Anstieg der durchschnittlichen Verspätung und Wartezeit fällt  
 dabei vergleichsweise moderat aus, da nur wenige Fahrten von diesem Phänomen betroffen sind.

Fahrplan	KVB-BT <sup>+</sup>						KVB-BT-C <sup>+</sup>						
	$w_{t_c}$ [Sek.]	-	0	30	60	120	180	-	0	30	60	120	180
	$t_t$ [Sek.]	-	0	30	30	30	30	-	0	30	30	30	30
Chorweiler	120-B01	0,0	0,0	30,0	60,0	120,0	180,0	0,0	0,0	46,5	47,8	48,6	49,5
	120-B02	8,7	9,1	30,8	59,5	119,0	178,7	5,3	5,4	17,8	28,5	50,3	70,8
	121-B01	6,1	6,2	31,0	60,1	77,1	78,1	5,1	5,4	15,4	24,5	43,6	62,7
	121-B02	0,0	0,0	30,0	60,0	77,3	78,2	0,0	0,0	46,5	47,8	48,6	49,5
	125-B01	50,0	50,0	53,0	51,9	108,6	110,1	62,8	62,9	60,1	60,5	60,3	167,5
	125-B02	1,1	1,2	30,0	60,0	120,0	180,0	0,8	0,7	1,2	1,7	107,6	108,5
Ostheim	126-B02	0,0	0,0	20,3	21,3	23,3	25,3	0,0	0,0	0,5	1,0	107,6	108,5
	152-B01	0,0	0,0	30,0	60,0	120,0	180,1	0,0	0,0	12,0	42,2	78,2	172,7
	152-B02	0,0	0,0	30,0	60,0	120,0	180,0	0,0	0,0	9,8	9,9	10,0	10,1
Porz	157-B01	0,0	0,0	20,3	20,8	119,9	180,0	0,0	0,0	20,2	20,4	24,9	49,3
	152-B01	0,0	0,0	30,0	60,0	120,0	180,0	0,0	0,0	14,7	66,3	66,3	66,3
	154-B01	0,0	0,0	30,0	60,0	120,0	180,0	0,0	0,0	20,0	20,0	20,0	20,0
	160-B01	0,0	0,0	30,0	60,0	120,0	180,0	0,0	0,0	20,0	20,0	126,3	126,3
	161-B01	0,0	0,0	30,0	60,0	120,0	140,0	0,0	0,0	20,0	20,0	20,0	20,0
	162-B01	0,0	0,0	30,0	60,0	120,0	180,0	0,0	0,0	30,0	60,0	80,0	186,3
	∅	4,4	4,4	30,4	54,2	107,0	148,7	4,9	5,0	22,3	31,4	59,5	84,5

Tabelle 8.5: Durchschnittliche Verspätung an den unterschiedlichen Haltestellen in Abhängigkeit des Fahrplans und der Warte-/Umsteigezeit, KVB-BT 2001

Fahrplan	KVB-BT <sup>+</sup>						KVB-BT-C <sup>+</sup>						
	$w_{t_c}$ [Sek.]	-	0	30	60	120	180	-	0	30	60	120	180
	$t_t$ [Sek.]	-	0	30	30	30	30	-	0	30	30	30	30
Chorweiler	120-B01	54,3	63,5	93,9	123,2	183,2	242,9	40,3	49,6	97,5	98,1	99,3	100,0
	120-B02	6,3	10,8	32,8	61,8	120,6	161,1	10,1	15,7	28,1	38,4	60,8	81,5
	121-B01	12,9	18,8	43,9	73,0	89,6	90,5	14,3	20,2	30,5	38,6	57,6	76,9
	121-B02	46,1	55,4	85,2	115,5	132,3	133,9	44,2	53,2	99,9	101,1	102,3	102,9
	125-B01	0,0	0,1	3,9	4,1	61,9	64,1	0,0	0,0	0,1	0,3	1,2	108,6
	125-B02	6,8	13,8	42,5	72,8	133,1	193,3	8,6	16,0	16,4	17,4	123,0	123,9
Ostheim	126-B02	0,0	10,0	30,3	31,3	33,3	35,3	0,0	10,0	10,5	11,0	117,6	118,5
	152-B01	69,4	76,8	76,1	73,8	68,7	59,8	47,4	56,7	59,0	37,1	70,9	126,9
	152-B02	47,8	56,7	86,5	116,5	176,6	236,6	48,9	58,1	68,3	68,1	68,3	68,4
Porz	157-B01	0,0	9,9	30,2	30,7	120,7	142,3	0,0	9,9	30,1	30,3	32,5	40,1
	152-B01	0,0	10,0	40,0	70,0	130,0	190,0	0,0	10,0	24,7	76,3	76,3	76,3
	154-B01	0,0	10,0	40,0	70,0	130,0	190,0	0,0	10,0	30,0	30,0	30,0	30,0
	160-B01	0,0	10,0	40,0	70,0	130,0	190,0	0,0	10,0	30,0	30,0	136,3	136,3
	161-B01	0,0	10,0	40,0	70,0	130,0	150,0	0,0	10,0	30,0	30,0	30,0	30,0
	162-B01	0,0	10,0	40,0	70,0	130,0	190,0	0,0	10,0	40,0	70,0	90,0	196,3
	∅	16,2	24,4	48,3	70,2	118,0	151,3	14,3	22,6	39,7	45,1	73,1	94,4

Tabelle 8.6: Durchschnittliche Wartezeit an den unterschiedlichen Haltestellen in Abhängigkeit des Fahrplans und der Warte-/Umsteigezeit, KVB-BT 2001

Den Preis für den erhöhten Anteil erfüllter Umsteige-  
 ge Verbindungen zeigen Tabelle 8.5 bis Tabelle 8.7.  
 Wie zu erwarten steigt die durchschnittliche Verspätung und Wartezeit an den drei Umsteigepunkten  
 mit zunehmendem Wert von  $w_{t_c}$ . Unter Fahrplan KVB-BT<sup>+</sup> fällt der Effekt dabei wesentlich deutlicher

aus, da unter ihm mehr Fahrzeuge vergeblich versuchen die Umsteigeverbindungen durch zusätzliches Warten zu halten. Entsprechend wird die durchschnittliche Verspätung an den Umsteigeknoten, d.h. die Zeitdauer um die der geplante Abfahrtszeitpunkt überschritten wird, weitestgehend vom Wert von  $w_{t_c}$  bestimmt. Die durchschnittliche Wartezeit, d.h. der Zeitabschnitt zwischen Ende des Passagierwechsels und Abfahrt, fällt insbesondere für die Linien für die die Umsteigeknoten Starthaltestellen darstellen häufig noch etwas höher aus. Dies ist unter anderem dadurch zu erklären, dass Fahrzeuge durch den Dispatcher etwas früher als unbedingt notwendig an ihrem Starthaltepunkt eingesetzt werden, um eine künstliche Verspätung zu vermeiden (vgl. Abschnitt 5.1.4.3).

Die durch die Pro-Synchronisierung Strategie induzierten Effekte begrenzen sich jedoch nicht nur auf die Umsteigeknoten und die direkt betroffenen Linien. Wie Tabelle 8.7 bis Tabelle 8.9 zeigen, werden die Verspätungen und damit einhergehende Bunching-Effekte häufig über den gesamten restlichen Linienverlauf weitergetragen und übertragen sich z.T. in Form von erhöhter Verspätung und/oder Frühzeitigkeit auch auf die Rückrichtungen der Liniengruppen. Dieser Effekt ist, wie zu erwarten, besonders bei den Liniengruppen zu beobachten, deren Abfahrtszeit am Umsteigeknoten stark von der Abfahrtszeit der Zubringerfahrzeuge abweicht und dort entsprechend stark verzögert werden, wie bspw. bei Liniengruppe 162. Interessant ist dabei auch, dass diese Effekte unter den beiden Fahrplänen in unterschiedliche Richtungen ausschlagen. Während unter Fahrplan KVB-BT<sup>+</sup> eine Erhöhung der durchschnittlichen Linienverspätung der Rückrichtungen und eine damit einhergehende Verringerung der durchschnittlichen Frühzeitigkeit verzeichnet werden kann, steigen beide Werte unter Fahrplan KVB-BT-C<sup>+</sup> (sehr) leicht an. Dies lässt darauf schließen, dass es unter Fahrplan KVB-BT-C<sup>+</sup> zu stärkeren Bunching-Effekten kommt.

Die Auswirkungen der Pro-Synchronisierung Strategie sind darüber hinaus, in abgeschwächter Form, auch sichtbar betrachtet man die durchschnittliche Linienverspätung des gesamten Busnetzes (vgl. Tabelle 8.7). Der Effekt wird dabei allerdings erst ab einer maximalen Wartezeit von  $w_{t_c} = 120$  Sekunden deutlich, während er bei ausschließlicher Betrachtung der durch Umsteigeverbindungen betroffenen Buslinien auch schon bei kleinen Verzögerungen von 30 Sekunden sichtbar wird.

Fahrplan	KVB-BT <sup>+</sup>						KVB-BT-C <sup>+</sup>					
$w_{t_c}$ [Sek.]	-	0	30	60	120	180	-	0	30	60	120	180
$t_t$ [Sek.]	-	0	30	30	30	30	-	0	30	30	30	30
120-B01	0,6	1,5	12,2	33,4	78,9	124,5	<b>0,3</b>	0,6	18,2	18,6	18,6	38,2
120-B02	6,6	6,8	9,0	15,0	33,7	64,0	<b>5,6</b>	5,7	6,5	8,3	14,9	21,6
121-B01	<b>6,7</b>	6,8	7,9	14,7	20,1	18,6	6,8	7,0	7,2	9,1	17,7	26,2
121-B02	<b>4,2</b>	4,8	14,6	34,5	44,3	44,2	4,5	5,7	29,1	29,4	29,3	27,2
125-B01	<b>27,5</b>	27,3	27,8	27,1	39,1	39,8	34,4	34,7	33,0	33,2	32,5	60,0
125-B02	<b>4,8</b>	4,9	12,9	34,5	84,4	133,9	5,4	5,9	5,8	5,8	83,8	83,2
126-B02	<b>0,1</b>	0,4	3,6	4,3	5,6	7,6	2,3	4,5	4,4	4,5	72,1	95,5
152-B01	<b>0,2</b>	<b>0,2</b>	3,9	15,9	70,8	138,5	0,3	0,3	2,2	17,5	35,2	102,9
152-B02	0,7	0,8	3,2	10,4	32,7	55,7	<b>0,3</b>	<b>0,3</b>	0,8	0,8	0,8	0,8
154-B01	4,2	4,2	8,0	21,4	70,2	129,9	<b>3,8</b>	4,0	6,5	6,4	6,5	6,5
157-B01	<b>0,3</b>	1,2	9,3	9,7	101,4	161,4	<b>0,3</b>	1,2	9,2	9,5	14,0	39,3
160-B01	1,3	1,3	8,3	30,4	88,6	146,9	<b>0,6</b>	1,0	5,7	20,0	84,3	131,6
161-B01	0,9	0,9	9,8	34,0	92,8	106,4	<b>0,1</b>	<b>0,1</b>	2,1	5,8	5,0	24,4
162-B01	19,7	20,0	40,1	73,1	134,8	196,4	<b>16,9</b>	23,4	55,9	90,9	111,4	232,8
∅	<b>5,6</b>	5,8	12,2	25,6	64,1	97,7	5,8	6,7	13,3	18,6	37,6	63,6
Busnetz gesamt	<b>5,5</b>	<b>5,5</b>	6,9	9,6	18,3	27,0	5,7	5,8	7,2	8,3	12,2	17,8

Tabelle 8.7: Durchschnittliche Linienverspätung der durch Umsteigeverbindungen betroffenen Linien, KVB-BT 2001

Fahrplan	KVB-BT <sup>+</sup>						KVB-BT-C <sup>+</sup>					
wt <sub>c</sub> [Sek.]	-	<b>0</b>	<b>30</b>	<b>60</b>	<b>120</b>	<b>180</b>	-	<b>0</b>	<b>30</b>	<b>60</b>	<b>120</b>	<b>180</b>
t <sub>t</sub> [Sek.]	-	<b>0</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	-	<b>0</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
<b>126-B01</b>	3,5	3,5	3,6	<b>3,4</b>	3,5	3,5	4,8	4,9	4,8	4,8	4,8	4,8
<b>154-B02</b>	10,8	10,9	10,6	10,7	10,4	10,3	9,1	9,2	9,1	9,1	9,1	<b>8,7</b>
<b>157-B02</b>	2,2	2,1	2,1	2,2	31,6	89,8	2,1	<b>2,0</b>	2,2	2,2	5,6	26,2
<b>160-B02</b>	<b>0,1</b>	<b>0,1</b>	0,2	1,5	22,4	72,2	<b>0,1</b>	<b>0,1</b>	<b>0,1</b>	<b>0,1</b>	0,2	0,2
<b>161-B02</b>	0,2	0,2	0,2	0,2	<b>0,1</b>	<b>0,1</b>	0,5	0,5	0,4	0,5	0,4	0,5
<b>162-B02</b>	4,4	4,5	3,9	<b>3,0</b>	4,4	14,5	10,8	10,7	10,9	10,6	7,7	5,1
∅	3,5	3,6	<b>3,4</b>	3,5	12,1	31,7	4,6	4,6	4,6	4,6	4,6	7,6

*Tabelle 8.9: Durchschnittliche Linienverspätung der Rückrichtungen der durch Umsteigeverbindungen betroffenen Linien, KVB-BT 2001*

Fahrplan	KVB-BT <sup>+</sup>						KVB-BT-C <sup>+</sup>					
wt <sub>c</sub> [Sek.]	-	<b>0</b>	<b>30</b>	<b>60</b>	<b>120</b>	<b>180</b>	-	<b>0</b>	<b>30</b>	<b>60</b>	<b>120</b>	<b>180</b>
t <sub>t</sub> [Sek.]	-	<b>0</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	-	<b>0</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
<b>126-B01</b>	18,1	18,0	18,0	18,1	18,0	18,0	<b>16,1</b>	<b>16,1</b>	16,2	16,2	<b>16,1</b>	<b>16,1</b>
<b>154-B02</b>	7,9	<b>7,5</b>	7,6	7,7	8,0	8,3	8,6	8,8	8,7	9,0	9,0	9,1
<b>157-B02</b>	11,4	11,4	11,4	11,4	3,1	<b>0,1</b>	11,4	11,5	11,4	11,4	11,1	10,7
<b>160-B02</b>	34,6	34,4	34,9	31,3	10,3	<b>1,2</b>	34,1	33,9	34,0	34,4	33,4	32,9
<b>161-B02</b>	15,9	15,9	16,2	16,5	18,0	17,6	14,5	14,6	14,7	14,7	14,6	14,7
<b>162-B02</b>	15,2	15,1	16,3	18,3	16,5	9,1	<b>6,3</b>	6,4	6,2	6,4	9,5	14,0
∅	17,2	17,1	17,4	17,2	12,3	<b>9,1</b>	15,2	15,2	15,2	15,4	15,6	16,3

*Tabelle 8.8: Durchschnittliche Frühzeitigkeit der Rückrichtungen der durch Umsteigeverbindungen betroffenen Linien, KVB-BT 2001*

Abschließend lässt sich festhalten, dass Fahrplan KVB-BT-C<sup>+</sup> auch unter dynamischen Bedingungen besser zur Umsetzung von gerichteten Umsteigeverbindungen geeignet ist. Er ermöglicht bereits ohne Eingriff des Dispatchers die Erfüllung eines signifikant größeren Anteils der Umsteigeverbindungen als der nur auf gleichmäßige Sicherheitsabstände ausgelegte Fahrplan KVB-BT<sup>+</sup>. Wird der Dispatcher zu Eingriffen veranlasst, kann unter Fahrplan KVB-BT-C<sup>+</sup> ein Großteil der Umsteigeverbindungen gehalten werden. Dabei ist allerdings zu beachten, dass die Notwendigkeit für die Mehrheit dieser Eingriffe in der durch die Optimierungsanwendung erzwungenen Struktur der Fahrpläne liegt. Wäre es – etwa durch die Hinzunahme von Kapazitätsbeschränkungen während der Optimierung – zulässig an gemeinsamen Haltepunkten zeitgleiche Abfahrten von Busfahrzeugen zu terminieren, könnte bereits ohne Eingriff des Dispatchers ein größerer Anteil von Umsteigeverbindungen gehalten werden, wie es die Kennzahlen für die Haltestelle Ostheim bereits vermuten lassen.

Der simulative Vergleich der beiden Fahrpläne hat darüber hinaus gezeigt, dass die dynamischen Bedingungen des Nahverkehrsnetzes, abhängig vom Fahrplan, sowohl ungewollt positive als auch negative Auswirkungen auf den Anteil der erfüllten Umsteigeverbindungen haben kann und die rein statische Betrachtung aufgrund von Zielfunktionswerten mit Vorsicht zu genießen ist.

Die Simulation kann darüber hinaus auch dazu herangezogen werden Empfehlungen bzgl. der Abfahrtsreihenfolge an den Umsteigeknoten zu tätigen. So erscheint es sinnvoll Linien 120-B01, 121-B01 und 125-B01 jeweils eine Minute vor Linie 18-T01 an der Haltestelle Chorweiler abfahren zu lassen, um für die Verspätungen dieser Linien zu kompensieren. Gleichzeitig sollten die Abfahrten der Linie 152-B01 an der Haltestelle Porz Markt zeitgleich mit den Abfahrten der Linien 7-T01/T02 und an der Haltestelle Ostheim zeitgleich mit den Abfahrten der Linien 9-T01/T02 erfolgen.

## 9. Zusammenfassung und Ausblick

### 9.1. Zusammenfassung

Nach einer Einführung in Kontext und Thema der Arbeit wurden bestehende Optimierungsverfahren zur Erstellung von Fahrplänen für den ÖPNV betrachtet. Dabei wurde zwischen Verfahren zur Erstellung von Fahrplänen mit hoher Serviceregularität, hoher Synchronisation sowie hoher Robustheit unterschieden und die für die Erstellung von Fahrplänen für multi-modale ÖPNV-Systeme als besonders vielversprechend identifizierten Ansätze von Genç in [39] und Ullrich et al. in [90] detailliert besprochen.

Anschließend wurde ein neuer Optimierungsansatz vorgestellt, der die Vorteile der Modelle von Genç und Ullrich et al. kombiniert, um Fahrpläne für multi-modale ÖPNV-Systeme mit hoher Serviceregularität und Synchronisation zu erzeugen, der darüber hinaus auch in der Lage ist weitergehende, für die Anwendbarkeit im täglichen Betrieb essentielle, planerische Ansprüche zu berücksichtigen. Als Indikator für die Serviceregularität wurden dabei möglichst große Abstände zwischen den einzelnen Abfahrten an Haltepunkten genutzt. Sie sollen zum einen verhindern, dass sich im täglichen Betrieb unweigerlich auftretende kleinere Verspätungen, in Verkehrssystemen in denen sich Fahrzeuge gemeinsame Ressourcen mit knapper Kapazität und wenigen Ausweichmöglichkeiten teilen, von einem Fahrzeug auf sein Nachfolgerfahrzeug übertragen und in der Folge die Performanz des Gesamtsystems beeinträchtigen. Zum anderen soll durch sie eine gleichmäßigere Verteilung der Passagiere auf die im Einsatz befindlichen Fahrzeuge erfolgen und damit eine effizientere Fahrzeugauslastung mit weniger starken Auslastungsspitzen sichergestellt werden. Für die Serviceregularität wurde mit der Erfüllung entsprechend formulierter verkehrsplanerischer Bedingungen über die Abfahrtszeitabweichungen betroffener Linien hingegen ein indirekter Indikator herangezogen. Über ihn soll sichergestellt werden, dass für Passagiere wichtige Umsteigeverbindungen innerhalb und zwischen Verkehrsmodi durch den Fahrplan eingehalten werden.

Das resultierende Optimierungsproblem wurde als disjunktives Programm formuliert und zu seiner Lösung wurde, unter Verwendung der proprietären Programmbibliothek IBM ILOG CPLEX, eine Java-Anwendung realisiert. Die Korrektheit und Leistungsfähigkeit dieses Verfahrens wurde durch beispielhafte Anwendungen auf Modelle zweier künstliche Netze sowie des Kölner Busnetzes aus dem Jahr 2001 demonstriert. Dabei konnte gezeigt werden, dass das Verfahren dazu in der Lage ist in akzeptabler Laufzeit gültige Fahrpläne mit hoher Güte zu erzeugen. Dies gilt insbesondere für Verkehrssysteme deren Streckennetze aus mehreren Zusammenhangskomponenten bestehen und für die keine verkehrsplanerischen Bedingungen zu berücksichtigen sind, die Linien unterschiedlicher Zusammenhangskomponenten betreffen. Für diesen Fall wurde ein von Genç vorgestellter Ansatz realisiert, der zunächst Teilfahrpläne für die einzelnen Zusammenhangskomponenten generiert und diese anschließend zu einem Gesamtfahrplan zusammensetzt. Unter Verwendung dieses Ansatzes konnte die zur Erstellung eines Fahrplans für das Kölner Busnetz notwendige Zeit um mindestens 80 Prozent reduziert werden.

Um die Eignung der unter statischen Bedingungen erzeugten Fahrpläne für den täglichen Betrieb – d.h. unter dynamischen Bedingungen – zu evaluieren, wurde ein Simulationsverfahren herangezogen. Nach einer Einführung in allgemeine und spezifisch auf (Nah-)Verkehrssysteme fokussierte Simulationsmodelle wurde dazu ein ereignisbasiertes, mesoskopisches Simulationsmodell für (multi-modalen) ÖPNV entworfen. Dieses Modell zeichnet sich dadurch aus, dass – anders als bei einem Großteil existierender Modelle – nicht der Individualverkehr sondern die abzubildenden Nahverkehrssysteme und die Besonderheiten ihrer Streckennetze im Mittelpunkt aller Modellierungsentscheidungen stehen. Während das Modell in der in dieser Arbeit vorgestellten Version nur Bus- und Stadtbahnverkehr beinhaltet, kann es aufgrund seiner modularen Struktur zukünftig auf einfache Art und Weise um weitere Verkehrsmodi erweitert werden. Darüber hinaus ist das Modell so ausgelegt, dass es mit statistischen Eingangsdaten nahezu beliebiger Granularität arbeiten kann. Stehen detaillierte statistische Daten nicht zur Verfügung, arbeitet das Modell mit durch die Nutzerin festzulegenden Parametern und allgemeinen Verteilungsfunktionen.

Auf Basis des formulierten Simulationsmodells wurde mit Hilfe eines eigens entwickelten ereignisbasierten Java-Frameworks das gewünschte neue Simulationsmodul entwickelt. Seine Funktionalität und Plausibilität wurde durch theorie- und funktionsbezogene Validierungen anhand von Modellen eines künstlichen Busnetzes sowie des Kölner Stadtbahnnetzes aus dem Jahr 2001 gezeigt. Dazu wurden jeweils durch die Simulationsanwendung erhobene Kennzahlen für den durch die Optimierungsanwendung ausgewiesenen Optimalfahrplan ohne verkehrsplanerische Bedingungen und für den schlechtesten noch gültigen Fahrplan verglichen. Die Ergebnisse zeigen, dass das Simulationsverhalten, trotz kleinerer, aufgrund der spärlichen Datenbasis resultierender Simulationsartefakte, sowohl für Bus- als auch für Stadtbahnssysteme plausibel ist. Insbesondere konnte das in Bussystemen häufig anzutreffende Phänomen der ungewollten Kolonnenfahrten abgebildet werden. Weiterhin konnte durch die Ergebnisse gezeigt werden, dass der durch die Optimierungsanwendung als optimal ausgewiesene Fahrplan, unter den gewählten Simulationsparametern, eine bessere Systemperformanz aufweist, die Simulationsanwendung also zur Evaluation der Güte von Fahrplänen herangezogen werden kann.

Zur Evaluation der Leistungsfähigkeit der Simulationsanwendung wurden anschließend Laufzeitvergleiche auf Basis von Modellen eines künstlichen Busnetzes sowie des Kölner Bus- und Stadtbahnnetzes durchgeführt. Die Ergebnisse zeigen, dass die Anwendung für die reine Simulation eine sehr gute Laufzeit aufweist, die in Abhängigkeit der Größe des simulierten Netzes, des Umfangs des Fahr- und Umlaufplans sowie der Zahl der durchzuführenden Läufe steigt. Die Untersuchungen zur Laufzeit zeigen aber auch, dass die Simulationsanwendung noch erhebliches Verbesserungspotential im Bereich des Postprocessings besitzt: Bei Experimenten mit mehreren Simulationsläufen benötigt die Deserialisierung statistischer Daten einzelner Simulationsläufe und deren Aggregation zu statistischen Daten des Gesamtexperiments i.d.R. genauso lange wie die Durchführung aller Simulationsläufe des Experiments. Als Grund konnte dabei das extrem unbefriedigende Laufzeitverhalten der Standard-Java-Schnittstelle für die Deserialisierung abstrakter Datentypen identifiziert werden. Sie ist im Falle des künstlichen Busnetzes für nahezu 97 Prozent der Postprocessinglaufzeit verantwortlich.

In einem weiteren Schritt wurde das entworfene Simulationsmodell um Komponenten und Methoden erweitert, die es dazu in die Lage versetzen selbstständig einfache regelbasierte Maßnahmen des Verkehrsmanagements einzuleiten. Dazu erfolgte zunächst eine kurze Einführung in den Bereich des Verkehrsmanagements, die dort vorzufindenden unterschiedlichen Strategietypen, sowie eine Besprechung existierender Optimierungs- und Simulationsverfahren.

Anschließend wurde das Simulationsmodell um die Möglichkeit erweitert, größere Störungsszenarien in Form von exogenen Ereignissen zu berücksichtigen. Darüber hinaus wurde das im Simulationsmodell für das Verkehrsmanagement zuständige Verwaltungsmodul um diverse Verfahren erweitert, mit denen es eigenständig sowohl auf sich spontan entwickelnde kleinere Störungen als auch umfangreichere Störfälle reagieren kann. Zudem wurde der Dispatcher auch dazu in die Lage versetzt durch die Nutzerin zu definierende Rendezvous- und Umsteigeverbindungen zu berücksichtigen und auf Wunsch Maßnahmen zu veranlassen, um diese Verbindungen mit höherer Wahrscheinlichkeit einzuhalten. Die realisierten Verkehrsmanagementmaßnahmen können damit in die Bereiche Haltepunkt-Kontrolle und Störfallmanagement eingeteilt werden. Erstere beinhalten Haltestrategien zur Reduzierung von Bunching-Effekten und Erhöhung der Fahrzeugsynchronisation sowie Verfahren zur dynamischen Modifikation und Verkürzung von Linienverläufen, um die Abfahrtszeitabweichung von einzelnen Fahrzeugen zu reduzieren. Letztere beinhalten Verfahren zur Linienmodifikation ganzer Mengen von durch Störungen betroffenen Fahrten. Dabei können in der in dieser Arbeit vorgestellten Version nur Störfälle verarbeitet werden, die Haltepunkte, Verbindungen oder Segmente betreffen. Störungen von Fahrzeugen können hingegen aufgrund mangelnder Informationen über den Fahrzeugpool und seine Verteilung über die verschiedenen Depots des Streckennetzes nicht verarbeitet werden. Eine zukünftige Erweiterung des Modells um diesen Fall ist aber mit vergleichsweise geringem Aufwand möglich.

Die Validierung der implementierten Verfahren des Verkehrsmanagements erfolgte stichpunktartig anhand von Modellen eines künstlichen Busnetzes und des Kölner Stadtbahnnetzes aus dem Jahr 2001. Für das künstliche Busnetz konnte gezeigt werden, dass die Anti-Bunching Haltestrategien in einer Reduktion des Anteils reduzierter Sicherheitsabstände und damit in einer Reduktion der Effekte ungewoll-

ter Kolonnenfahren mündet. Darüber hinaus konnte anhand von Untersuchungen zur dynamischen Modifikation des Linienverlaufs am Modell des künstlichen Busnetzes gezeigt werden, dass der Dispatcher sich im Simulationsverlauf entwickelnde Störungen korrekt erkennt und – entsprechende Alternativrouten vorausgesetzt – wunschgemäß reagiert, indem er den Linienverlauf einzelner Fahrzeuge gemäß den Vorgaben der Nutzerin modifiziert. Schließlich konnte durch die Simulation eines größeren Störfalls im Kölner Stadtbahnnetz auch gezeigt werden, dass der Dispatcher korrekt und plausibel auf größere Störfälle reagiert.

Die Funktionalität des Gesamtprogrammpakets wurde abschließend noch anhand eines typischen Anwendungsfalls für das kombinierte Kölner Bus- und Stadtbahnnetz aus dem Jahr 2001 demonstriert. Dazu wurden mit Hilfe der Optimierungsanwendung zwei Fahrpläne erzeugt und mit Hilfe der Simulationsanwendung hinsichtlich ihrer Eignung zur Umsetzung gerichteter Umsteigeverbindungen vom Bahn- zum Busverkehr an ausgewählten Haltestellen untersucht. Die Ergebnisse zeigen, dass durch das Optimierungsmodell in der Tat nicht nur Fahrpläne mit maximaler Serviceregularität erzeugt werden können, sondern, bei entsprechender Definition verkehrsplanerischer Bedingungen, auch wie gewünscht eine hohe Synchronisation erreicht werden kann. Diese Synchronisation kann während des operativen Betriebs auf Kosten erhöhter Verspätungen und Wartezeiten dabei noch erhöht werden, wenn der Dispatcher angewiesen wird entsprechend in den Betrieb einzugreifen.

## 9.2. Ausblick

In weiteren Schritten sollte zunächst eine weitere Validierung der Simulationsanwendung erfolgen. Um die Anwendung tatsächlich für Empfehlungen an Verkehrsplaner einsetzen zu können ist eine ergebnisbezogene Validierung mit Hilfe umfangreicher Realdaten unabdingbar. Insbesondere sind hier Daten zu Fahrzeiten und Ankunftsdaten notwendig. Auch weitere Daten zu Signalanlagen und Verknüpfungspunkten zwischen den unterschiedlichen Verkehrsmodi könnten helfen die Anwendung valider zu gestalten, indem sie die Modellierung weiterer Interaktionen zwischen den Verkehrsmodi, wie bspw. auf sowohl von Bus- als auch Bahnfahrzeugen verwendeten Streckenabschnitten, ermöglichen. Die vorhandene Datenbasis reicht für solche Vorhaben nicht aus und ist weitestgehend ausgeschöpft.

Sind detailliertere Realdaten verfügbar sollte darüber hinaus der Dispatcher um das bisher noch fehlende Störfallmanagement für Fahrzeugstörungen erweitert werden. Gleichzeitig könnten dann die bereits vorhandenen Maßnahmen um die Berücksichtigung von Ersatzfahrzeugen und Ersatzverkehr erweitert werden.

Darüber hinaus sollten die im Dispatcher umgesetzten regelbasierten Verfahren des Verkehrsmanagements mit (Echtzeit-)Optimierungsverfahren verglichen werden, um die Güte der vorgeschlagenen Maßnahmen besser abschätzen zu können. Hierzu sollte das Programmpaket um entsprechende Optimierungsmodule erweitert werden, die anschließend mit dem Dispatcher gekoppelt und alternativ zu den regelbasierten Verfahren durch den Dispatcher aufgerufen werden.

Interessant wäre in diesem Zusammenhang auch eine Erweiterung der Simulationsanwendung zur Onlinesimulation, also das Weitersimulieren eines realen Betriebstags von einer gegebenen Momentaufnahme aus. Mit einer solchen Erweiterung könnten nach dem Auftreten einer größeren Störung unterschiedliche Optionen für das Verkehrsmanagement geprüft werden. Für diese Erweiterung wären nur wenige Anpassungen an der Simulationsanwendung notwendig, insbesondere würde aber eine Echtzeitanbindung zum Informationssystem des betroffenen Verkehrsunternehmens benötigt.

Erweitert werden könnte auch die Optimierungsanwendung zur Fahrplangenerierung. Wie in Kapitel 8 gesehen, limitiert die Erzwingung von mindestens einminütigen Abständen zwischen Abfahrten an gemeinsamen Haltepunkten künstlich die Möglichkeit zur Umsetzung von Umsteigeverbindungen. Die Berücksichtigung der Fahrzeugkapazität des Haltepunkts könnte hier Abhilfe schaffen, indem – bei entsprechender Kennzeichnung durch die Nutzerin – so viele zeitgleiche Abfahrten zugelassen werden, wie Kapazität am Haltepunkt existiert.

Eine weitere hilfreiche Erweiterung der Optimierungsanwendung wäre die Möglichkeit mehrerer Abfahrten einer Linie am gleichen Haltepunkt. Da jede Linie aktuell nur exakt eine Abfahrtszeit pro Haltepunkt besitzt, ist eine Abbildung von zirkulären Linien und Linien mit Schleifen in ihrer Route nur unzureichend möglich, wie in Abschnitt 3.4 gesehen.

## A. Beschreibung des Datenbankmodells

Daten für frühere Optimierungs- und Simulationsmodule des CATS-Projekts wurden in einer Datenbank nach erweitertem ÖPNV5-Standard gehalten (siehe [96] für einen Überblick des Basismodells sowie [91] für im Rahmen des CATS-Projekts hinzugefügte Erweiterungen). Während dieser Standard eine ausreichende Basis für einfache Informationssysteme von (Nah-)Verkehrsunternehmen bietet, ist er für Anwendungen mit detailliertem Informationsbedarf nur bedingt geeignet und erfordert dazu zum Teil umfangreiche Datentransformationen. Beispielhaft seien hier Verbindungen erwähnt. Sie werden im ÖPNV5-Standard nicht explizit gespeichert, sondern müssen anhand ihres Start- und Endhaltepunktes sowie der Menge aller Segmente zusammengesetzt werden. Existieren mehrere potentielle Wege zwischen zwei benachbarten Haltepunkten, kann auf Basis der Informationen im ÖPNV5-Standard nicht festgestellt werden, welcher Weg korrekt ist. Dazu müsste er um neue Relationen erweitert und die entsprechenden notwendigen Informationen dort gespeichert werden. Ähnliche Limitierungen im Umfang der abbildbaren Informationen finden sich in diversen Relationen des ÖPNV5-Standards. Aus diesem Grund wurde für die in dieser Arbeit entworfenen Module ein neues Datenbankmodell entworfen, das besser auf die Bedürfnisse der Optimierungs- und Simulationsmodule abgestimmt ist. Seine Relationen werden im Folgenden kurz beschrieben. Für sie alle gilt dabei: Ein Eintrag in der Spalte PK zeigt an, dass das entsprechende Attribut Bestandteil des Primärschlüssels der Relation ist.

### NET\_INDEX: Verwaltung unterschiedlicher Netzwerkinstanzen

PK	Name	Typ	Erläuterung
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	ID_NET	String	Kurzname der Netzwerkinstanz
	ID_VERSION	Integer	Versionsnummer
	IDENTIFIER	String	Langname der Netzwerkinstanz

*Tabelle A.1: Relation NET\_INDEX*

Die Relation NET\_INDEX (siehe Tabelle A.1) enthält für jedes verwaltete Streckennetz einen Eintrag mit einem identifizierenden numerischen Bezeichner, einem Kurz- sowie Langnamen sowie einer Versionsnummer, um unterschiedliche Ausbaustufen identischer Netzwerke unterscheiden zu können.

### TIME\_TABLE\_INDEX: Verwaltung unterschiedlicher Fahrpläne

PK	Name	Typ	Erläuterung
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
	ID_VERSION	Integer	Versionsnummer
	IDENTIFIER	String	Erläuternder Bezeichner
	OBJECTIVE_FVALUE	Integer	Gesamtzielfunktionswert des Fahrplans
	ADHERENCE_VALUE	Integer	Normalisierter Wert des Erfülltheitsgrades im Zielfunktionswert
	RELATIVE_GAP	Double	Durch CPLEX ausgegebene relative Abweichung zwischen dem Zielfunktionswert der berechneten Lösung und der besten berechneten Schranke

*Tabelle A.2: Relation TIME\_TABLE\_INDEX*

Die Relation TIME\_TABLE\_INDEX (siehe Tabelle A.2) enthält für jeden verwalteten Fahrplan einen Eintrag mit einem identifizierenden numerischen Bezeichner, einem Verweis auf den identifizierenden Bezeichner des zugehörigen Netzwerks, eine Versionsnummer, sowie einen erläuternden Bezeichner. Darüber hinaus können jedem Eintrag optional noch Angaben über den Gesamtzielfunktionswert, den normalisierten Erfülltheitsgrad und der durch CPLEX ausgewiesenen relativen Abweichung zwischen dem Gesamtzielfunktionswert der besten berechneten Schranke zugewiesen werden.

**BASIC\_INTERVAL: Verwaltung von Taktintervallen**

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	TIMETABLE_INDEX	Integer	Index des zugehörigen Fahrplans
<b>PK</b>	LINE_ID	Integer	Index der zugehörigen Linie
	BASIC_INTERVAL	Integer	Taktintervall

Tabelle A.3: Relation BASIC\_INTERVAL

Die Relation BASIC\_INTERVAL (siehe Tabelle A.3) verwaltet die zur Erstellung eines Fahrplans verwendeten Taktintervalle der einzelnen Linien.

**TRIPS: Verwaltung von Fahrten**

PK	Name	Typ	Erläuterung
<b>PK</b>	TIME_TABLE_INDEX	Integer	Index des zugehörigen Fahrplans
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	START_TIME	Integer	Startzeit in Sekunden nach 0 Uhr
	LINE_INDEX	Integer	Index der zugehörigen Linie

Tabelle A.4: Relation TRIPS

In der Relation TRIPS (siehe Tabelle A.4) werden die zu einem Fahrplan gehörenden einzelnen Fahrten verwaltet. Ihnen werden dazu ein eindeutiger numerischer Bezeichner, eine Startzeit sowie die zugehörige Linie zugewiesen.

**LOCATIONS: Verwaltung von Orten**

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	POSITION_X	Integer	X-Koordinate/Längengrad
	POSITION_Y	Integer	Y-Koordinate/Breitengrad

Tabelle A.5: Relation LOCATIONS

In der Relation LOCATIONS (siehe Tabelle A.5) werden die für ein Streckennetz relevanten Orte verwaltet. Dabei handelt es sich um Positionen im Netzwerk an denen für die Abbildung des Systemverhaltens wichtige Elemente zu finden sind. Beispiele für solche Elemente sind Weichen, Kreuzungen oder Signalanlagen. Ein Ort kann dabei als Position mehrerer unterschiedlicher Elemente fungieren.

**TRANSIT\_TYPE: Verwaltung von Verkehrsmitteltypen**

PK	Name	Typ	Erläuterung
<b>PK</b>	ID	Integer	Identifizierender Bezeichner
	IDENTIFIER	String	Erläuternder Bezeichner

Tabelle A.6: Relation TRANSIT\_TYPE

Die Relationen TRANSIT\_TYPE (siehe Tabelle A.6) verwaltet die im Datenbankmodell verwendeten Verkehrsmitteltypen und weist ihnen einen eindeutigen numerischen Bezeichner zu.

**STATIONS: Verwaltung von Haltestellen**

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	IDENTIFIER	String	Erläuternder Bezeichner
	LOCATION	Integer	Index des zugehörigen Orts
	TURNING_POINT	Boolean	True: Haltestelle besitzt einen Wendepunkt False: Haltestelle besitzt keinen Wendepunkt

Tabelle A.7: Relation STATIONS

In der Relation STATIONS (siehe Tabelle A.7) werden die Haltestellen eines Netzwerks verwaltet. Ihnen wird ein eindeutiger numerischer sowie ein erläuternder Bezeichner, und ein Referenzort zugewiesen. Darüber hinaus wird für jeden Eintrag noch festgelegt, ob an der zugehörigen Haltestelle ein Wendepunkt zu finden ist oder nicht.

### STOPS: Verwaltung von Haltepunkten

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	IDENTIFIER	String	Erläuternder Bezeichner
	LOCATION	Integer	Index des zugehörigen Orts
	STATION	Integer	Index der zugehörigen Haltestelle
	TRANSIT_TYPE	Integer	Index des zugehörigen Verkehrsmitteltyps
	CAPACITY	Integer	Maximale Fahrzeugkapazität
	IS_TIME_CONTROL_POINT	Boolean	True: Haltepunkt kann zur Umsetzung von Haltestrategien verwendet werden False: Haltepunkt kann nicht zur Umsetzung von Haltestrategien verwendet werden

*Tabelle A.8: Relation STOPS*

Zur Verwaltung der Haltepunkte eines Netzwerks wird die Relation STOPS verwendet (siehe Tabelle A.8). Neben einem eindeutigen numerischen sowie einem erläuternden Bezeichner wird jeder Haltepunkt mit seiner zugehörigen Haltestelle und einem Ort im Netzwerk assoziiert. Außerdem besitzt jeder Eintrag Angaben zum zulässigen Verkehrsmitteltyp und der maximale Fahrzeugkapazität, sowie einen Indikator zur Angabe ob der zugehörige Haltepunkt zur Umsetzung von Haltestrategien verwendet werden kann.

### CONNECTIONS: Verwaltung von Verbindungen

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	IDENTIFIER	String	Erläuternder Bezeichner
	STOP_START	Integer	Index des Starthaltepunktes
	STOP_END	Integer	Index des Zielhaltepunktes
	LENGTH	Integer	Länge der Verbindung in Metern
	TIME	Integer	Geplante Fahrzeit in Sekunden

*Tabelle A.9: Relation CONNECTIONS*

Die Verbindungen eines Netzwerks werden in der Relation CONNECTIONS verwaltet (siehe Tabelle A.9). Wie den Haltepunkten, werden auch den Verbindungen jeweils ein eindeutiger numerischer sowie ein erläuternder Bezeichner zugewiesen. Darüber hinaus wird jede Verbindung mit den numerischen Bezeichnern der Haltepunkte die sie verbindet, ihrer Länge in Metern und ihrer geplanten Fahrzeit in Sekunden assoziiert.

### SEGMENTS: Verwaltung von Segmenten

In der Relation SEGMENTS (siehe Tabelle A.10) werden die Segmente des Netzwerks verwaltet. Sie weisen die gleichen Attribute wie Verbindungen auf, werden aber mit zwei Orten anstatt zwei Haltepunkten assoziiert. Darüber hinaus werden für jedes Segment der zulässige Verkehrsmitteltyp und die erlaubte Höchstgeschwindigkeit in km/h gespeichert.

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	LOCATION_START	Integer	Index des Startorts
	LOCATION_END	Integer	Index des Zielorts
	LENGTH	Integer	Länger in Metern
	TIME	Integer	Geplante Fahrzeit in Sekunden
	TRANSIT_TYPE	Integer	Index des zugehörigen Verkehrstyps
	MAX_SPEED	Integer	Erlaubte Höchstgeschwindigkeit in km/h

*Tabelle A.10: Relation SEGMENTS*

### SEGMENT\_TO\_CONNECTION: Zuweisung von Segmenten zu Verbindungen

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	CONNECTION	Integer	Index der zugehörigen Verbindung
<b>PK</b>	RUNNING_NUMBER	Integer	Position des Segments innerhalb der Verbindung
	SEGMENT	Integer	Index des zugehörigen Segments

*Tabelle A.11: Relation SEGMENT TO CONNECTION*

Die Zuweisung von Segmenten zu Verbindungen erfolgt in der Relation SEGMENT\_TO\_CONNECTION (siehe Tabelle A.11). Sie assoziiert jede Verbindung mit einer fortlaufend nummerierten Menge ihrer zugehörigen Segmente.

### SWITCHES: Verwaltung von Weichen

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	IDENTIFIER	String	Erläuternder Bezeichner
	LOCATION	Integer	Index des zugehörigen Orts

*Tabelle A.12: Relation SWITCHES*

Weichen werden in der Relation SWITCHES verwaltet (siehe Tabelle A.12). Auch ihnen wird ein eindeutiger numerischer Bezeichner sowie ein erläuternder Bezeichner und ein Ort des Netzwerks zugewiesen.

### SIGNALS: Verwaltung von Signalen

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	TYPE	Integer	Signaltyp
	LOCATION	Integer	Index des zugehörigen Orts
	SIGNAL_GROUP	Integer	Index der zugehörigen Signalgruppe

*Tabelle A.13: Relation SIGNALS*

Die Relation SIGNALS (siehe Tabelle A.13) besitzt einen Eintrag für jede in einem Netzwerk zu findende Signalanlage. Ihnen wird neben einem Ort des Netzwerks auch ein Typ zugewiesen, um z.B. zwischen Signalanlagen unterschiedlicher Verkehrsmittel unterscheiden zu können. Darüber hinaus wird jeder Signalanlage eine Signalgruppe zugeordnet, um miteinander in Verbindung stehende Signale, wie bspw. an einer Kreuzung, erfassen zu können.

### LINE\_INDEX: Verwaltung von Linien

In der Relation LINE\_INDEX (siehe Tabelle A.14) werden die Linien der unterschiedlichen Netzwerke verwaltet. Ihnen werden ein eindeutiger numerischer Bezeichner sowie ein Kurz- und ein Langname zugewiesen. Darüber hinaus werden sie mit ihrer zugehörigen Liniengruppe, Verkehrsmitteltyp und Fahrzeugtyp (z.B. Niederflur- oder Hochflurbahn) assoziiert.

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	INDEX	Integer	Identifizierender Bezeichner
	VARIANT_NAME	String	Variantenbezeichner (kurz)
	VARIANT_LONG_NAME	String	Variantenbezeichner (lang)
	LINE	Integer	Zugehörige Nummer der Liniengruppe
	VEHICLE_TYPE	Integer	Index des zugehörigen Fahrzeugtyp
	TRANSIT_TYPE	Integer	Index des zugehörigen Verkehrstyps

Tabelle A.14: Relation *LINE INDEX*

### LINE\_ROUTES: Verwaltung von Linienverläufen

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	LINE_INDEX	Integer	Index der zugehörigen Linie
<b>PK</b>	RUNNING_NUMBER	Integer	Position des Haltepunkts im Linienverlauf
	STOP	Integer	Index des zugehörigen Haltepunkts

Tabelle A.15: Relation *LINE ROUTES*

Linienverläufe werden in der Relation *LINE\_ROUTES* verwaltet (siehe Tabelle A.15). Sie assoziiert den numerischen Bezeichner einer Linie mit einer fortlaufend nummerierten Menge abzufahrender Haltepunkte.

### OPPOSITE\_LINES: Verwaltung von gegenläufigen Linien

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	LINE_INDEX	Integer	Index der zugehörigen Linie
<b>PK</b>	OPPOSITE_LINE_INDEX	Integer	Index der gegenläufigen Linie
	RUNNING_NUMBER	Integer	Fortlaufende Nummerierung

Tabelle A.16: Relation *OPPOSITE\_LINES*

In der Relation *OPPOSITE\_LINES* (siehe Tabelle A.16) werden Linien mit ihren gegenläufigen Pendants assoziiert. Da mehrere gegenläufige Linien existieren können, erhalten sie zusätzlich eine fortlaufende Nummerierung.

### TRUNCATED\_LINES: Verwaltung von verkürzten Linien

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	LINE_INDEX	Integer	Index der zugehörigen Linie
<b>PK</b>	TRUNCATED_LINE_INDEX	Integer	Index der verkürzten Linie

Tabelle A.17: Relation *TRUNCATED\_LINES*

Die Relation *TRUNCATED\_LINES* (siehe Tabelle A.17) assoziiert Varianten einer Liniengruppe mit verkürzten Alternativen der gleichen Liniengruppe. Dies ist bspw. nützlich wenn Verstärkerfahrten eingesetzt werden sollen, um einen spezifischen Abschnitt des Linienverlaufs mit hoher Nachfrage zu entlasten.

### SCENARIO\_INDEX: Verwaltung von Szenarien

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	SCENARIO_INDEX	Integer	Identifizierender Bezeichner
	NO_PERIODS	Integer	Anzahl an Zeitperioden
	PERIOD_LENGTH	Integer	Länge pro Periode in Sekunden
	COMMENT	String	Szenariobeschreibung

Tabelle A.18: Relation *SCENARIO\_INDEX*

Die Relation *SCENARIO\_INDEX* (siehe Tabelle A.18) enthält einen Eintrag für jedes durch die Nutzerin spezifizierte Szenario. Ein Szenario muss einem spezifischen Streckennetz zugeordnet sein und

erhält einen eindeutigen numerischen Bezeichner. Darüber hinaus muss durch die Nutzerin angegeben werden, in wieviele zeitliche Perioden ein Szenario eingeteilt ist und welche Länge eine Periode hat. Über diese Angabe werden z.B. Veränderungen in den Ankunftsrate über den Simulationsverlauf abgebildet. Außerdem kann jedem Eintrag noch ein erklärender Kommentar hinzugefügt werden.

### SCENARIO\_ARRIVAL\_RATES: Verwaltung von Passagierankunftsrate

PK	Name	Typ	Erläuterung
PK	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
PK	SCENARIO_INDEX	Integer	Index des zugehörigen Szenarios
PK	STOP	Integer	Index des zugehörigen Haltepunkts
PK	PERIOD	Integer	Index der zugehörigen Periode
	ARRIVAL_RATE	Double	Ankunftsrate

Tabelle A.19: Relation SCENARIO\_ARRIVAL\_RATES

Die für einzelne Szenarien zu verwendenden Passagierankunftsrate werden in der Relation SCENARIO\_ARRIVAL\_RATES verwaltet (siehe Tabelle A.19). Jedem Haltepunkt des zugehörigen Netzwerks muss dazu für jede Periode eine Ankunftsrate zugewiesen werden.

### NODE\_TYPE: Verwaltung von Knotentypen

PK	Name	Typ	Erläuterung
PK	ID	Integer	Identifizierender Bezeichner
	DESCRIPTION	String	Erläuternder Bezeichner

Tabelle A.20: Relation NODE\_TYPE

Die Relation NODE\_TYPE (siehe Tabelle A.20) assoziiert die im Simulationsmodul verwendeten Knotentypen mit eindeutigen numerischen Bezeichnern und erläuternden Beschreibungen. So ist beim Einlesen von Störungen sofort ersichtlich ob sie eine Verbindung, ein Segment oder einen Haltepunkt betreffen.

### DISRUPTION\_TYPE: Verwaltung von Störungstypen

PK	Name	Typ	Erläuterung
PK	ID	Integer	Identifizierender Bezeichner
	DESCRIPTION	String	Erläuternder Bezeichner

Tabelle A.21: Relation DISRUPTION\_TYPE

Die im Simulationsmodul abbildbaren unterschiedlichen Störungstypen werden in der Relation DISRUPTION\_TYPE (siehe Tabelle A.21) mit einem eindeutigen numerischen Bezeichner und einer erläuternden Beschreibung assoziiert.

### SCENARIO\_DISRUPTIONS: Verwaltung von Störfallszenarien

PK	Name	Typ	Erläuterung
PK	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
PK	SCENARIO_INDEX	Integer	Index des zugehörigen Szenarios
PK	RUNNING_NUMBER	Integer	Fortlaufende Nummerierung zur Unterscheidung unterschiedlicher Störungen eines Szenarios
	START_TIME	Integer	Eintrittszeitpunkt der Störung in Sekunden nach 0 Uhr
	DURATION	Integer	Dauer der Störung in Sekunden
	TYPE	Integer	Index des zugehörigen Störungstyps
	EFFECT	Double	Relativer Störungseffekt
	NODE_TYPE	Integer	Index des zugehörigen Knotentyps
	NODE_ID	Integer	Index des zugehörigen Knotens
	AFFECTS_NODE	Boolean	True: Störung betrifft Knoten False: Störung betrifft Fahrzeug
	VEHICLE_ID	Integer	Index des zugehörigen Fahrzeugs

Tabelle A.22: Relation SCENARIO\_DISRUPTIONS

In der Relation SCENARIO\_DISRUPTIONS (siehe Tabelle A.22) werden fortlaufend nummerierte Störungen zu Störfallszenarios zusammengefasst. Für jede Störung eines Szenarios muss ein Eintrag mit dem Typ der Störung, ihrer Startzeit und der durch die Störung betroffene Komponente existieren. Für temporäre Störungen muss außerdem eine Dauer angegeben werden. Störungen die Halte- oder Fahrzeiten anpassen, müssen darüber hinaus den relativen Störungseffekt angeben.

#### SCENARIO\_RENDEZVOUS\_CONNECTIONS: Verwaltung von Rendezvousverbindungen

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	SCENARIO_INDEX	Integer	Index des zugehörigen Szenarios
<b>PK</b>	PERIOD	Integer	Index der zugehörigen Periode
<b>PK</b>	STATION_INDEX	Integer	Index der zugehörigen Haltestelle

*Tabelle A.23: Relation SCENARIO\_RENDEZVOUS\_CONNECTIONS*

In der Relation SCENARIO\_RENDEZVOUS\_CONNECTIONS (siehe Tabelle A.23) werden die während eines Szenarios durch den Dispatcher zu berücksichtigenden Rendezvousverbindungen verwaltet. Dazu wird das betroffene Netzwerk, das zugehörige Szenario, die Zeitperiode in der und die Haltestelle an der die Rendezvousverbindung stattfinden soll angegeben.

#### SCENARIO\_TRANSFER\_CONNECTIONS: Verwaltung von Umsteigeverbindungen

PK	Name	Typ	Erläuterung
<b>PK</b>	NET_INDEX	Integer	Index des zugehörigen Streckennetzes
<b>PK</b>	SCENARIO_INDEX	Integer	Index des zugehörigen Szenarios
<b>PK</b>	PERIOD	Integer	Index der zugehörigen Periode
<b>PK</b>	FIRST_STOP_INDEX	Integer	Index des Haltepunkts der Ziellinie
<b>PK</b>	FIRST_LINE_INDEX	Integer	Index der Ziellinie
<b>PK</b>	SECOND_STOP_INDEX	Integer	Index des Haltepunkts der Zubringerlinie
<b>PK</b>	SECOND_LINE_INDEX	Integer	Index der Zubringerlinie

*Tabelle A.24: Relation SCENARIO\_TRANSFER\_CONNECTIONS*

Gerichtete Umsteigeverbindungen werden in der Relation SCENARIO\_TRANSFER\_CONNECTIONS verwaltet (siehe Tabelle A.24). Dazu wird die eventuell zu verzögernde Ziellinie und ihr zugehöriger Haltepunkt mit der Zubringerlinie und ihrem zugehörigen Haltepunkt assoziiert. Dies geschieht für das jeweilige Netzwerk, Szenario und die Zeitperiode.

## B. Literaturverzeichnis

- [1] ABKOWITZ, M. ; TOZZI, J.: Research contributions to managing transit service reliability. In: *Journal of Advanced Transportation* Bd. 21 (1987), Nr. 1, S. 47–65
- [2] ADAMSKI, A.: Probabilistic models of passengers service processes at bus stops. In: *Transportation Research Part B: Methodological* Bd. 26B (1992), Nr. 4, S. 253–259
- [3] ANDERSSON, P. ; HERMANSSON, A. ; TENGVOLD, E. ; SCALIA-TOMBA, G.: Analysis and simulation of an urban bus route. In: *Transportation Research Part A: General* Bd. 13 (1979), Nr. 6, S. 439–466
- [4] APACHE COMMONS: *Apache Commons Lang*. URL <https://commons.apache.org/proper/commons-lang/>. - abgerufen am 2015-11-28
- [5] APACHE COMMONS: *Apache Commons Logging*. URL <https://commons.apache.org/proper/commons-logging/>. - abgerufen am 2015-11-28
- [6] BAMPAS, E. ; KAOURI, G. ; LAMPIS, M. ; PAGOURTZIS, A.: Periodic Metro Scheduling. In: JACOB, R. ; MÜLLER-HANNEMANN, M. (Hrsg.): *6th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'06), OpenAccess Series in Informatics (OASICS)*. Bd. 5. Dagstuhl, Germany : Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2006 — ISBN 978-3-939897-01-9
- [7] BANKS, J. ; CARSON II, J. ; NELSON, B. ; NICOL, D.: *Discrete-Event System Simulation*. 5. Aufl. Upper Saddle River : Pearson, 2010
- [8] BARNETT, A.: On Controlling Randomness in Transit Operations. In: *Transportation Science* Bd. 8 (1974), Nr. 2, S. 102–116
- [9] BAY, S.: Nearest neighbor classification from multiple feature subsets. In: *Intelligent Data Analysis* Bd. 3 (1999), Nr. 3, S. 191–209
- [10] BEHRISCH, M ; KNOCKE, M. (Hrsg.): *1st SUMO User Conference 2013, Berichte aus dem DLR-Institut für Verkehrssystemtechnik* ( Nr. 21). Berlin : Deutsches Zentrum für Luft- und raumfahrt, 2013
- [11] BEHRISCH, M. ; BIEKER, L. ; ERDMANN, J. ; KRAJZEWICZ, D.: SUMO-Simulation of Urban MObility-an Overview. In: *SIMUL 2011: The Third International Conference on Advances in System Simulation*, 2011, S. 55–60
- [12] BEHRISCH, M. ; ERDMANN, J. ; KRAJZEWICZ, D.: *Adding intermodality to the microscopic simulation package SUMO* : MESM, 2010
- [13] BEIN, W. ; KAMBUROWSKI, J. ; STALLMANN, M.: Optimal reduction of two-terminal directed acyclic graphs. In: *SIAM Journal on Computing* Bd. 21 (1992), Nr. 6, S. 1112–1129
- [14] BIEKER, L. ; KRAJZEWICZ, D. ; MORRA, A. ; MICHELACCI, C. ; CARTOLANO, F.: Traffic Simulation for All: A Real World Traffic Scenario from the City of Bologna. In: BEHRISCH, M. ; WEBER, M. (Hrsg.): *Modeling Mobility with Open Data*. Cham : Springer International Publishing, 2015 — ISBN 978-3-319-15023-9, S. 47–60
- [15] BORNE, P. ; FAYECH, B. ; HAMMADI, S. ; MAOUCHE, S.: Decision support system for urban transportation networks. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* Bd. 33 (2003), Nr. 1, S. 67–77
- [16] BURGHOUT, W.: *Hybrid microscopic-mesoscopic traffic simulation*. Stockholm, Dept. of Infrastructure, Royal Institute of Technology, University Stockholm, Dissertation, 2004
- [17] CACCHIANI, V. ; CAPRARA, A. ; FISCHETTI, M.: A Lagrangian Heuristic for Robustness, with an Application to Train Timetabling. In: *Transportation Science* Bd. 46 (2012), Nr. 1, S. 124–133
- [18] CACCHIANI, V. ; GALLI, L. ; TOTH, P.: A tutorial on non-periodic train timetabling and platforming problems. In: *EURO Journal on Transportation and Logistics* (2014), S. 1–36
- [19] CATS, O.: Multi-agent Transit Operations and Assignment Model. In: *Procedia Computer Science* Bd. 19 (2013), S. 809–814
- [20] CEDER, A. ; GOLANY, B. ; TAL, O.: Creating bus timetables with maximal synchronization. In: *Transportation Research Part A: Policy and Practice* Bd. 35 (2001), Nr. 10, S. 913–928
- [21] CHAMPION, A. ; ESPIÉ, S. ; MANDIAU, R. ; AUBERLET, J. M. ; KOLSKI, C.: A game-based, multi-agent coordination mechanism - application to road traffic and driving simulations. In: *Summer Computer Simulation Conference*. Canada : Society for Computer Simulation International, 2003, S. 644–649

- [22] CHAPMAN, R. A. ; MICHEL, J. F.: Modelling the Tendency of Buses to Form Pairs. In: *Transportation Science* Bd. 12 (1978), Nr. 2, S. 165–175
- [23] CORMAN, F. ; D'ARIANO, A. ; HANSEN, I. A.: Disruption handling in large railway networks. In: *WIT Transactions on The Built Environment*. Bd. 114, 2010, S. 629–640
- [24] CORMAN, F. ; D'ARIANO, A. ; PACCIARELLI, D. ; PRANZO, M.: A tabu search algorithm for re-routing trains during rail operations. In: *Transportation Research Part B: Methodological* Bd. 44 (2010), Nr. 1, S. 175–192
- [25] DAKIN, R. J.: A tree-search algorithm for mixed integer programming problems. In: *The Computer Journal* Bd. 8 (1965), S. 250–255
- [26] D'ARIANO, A. ; CORMAN, F. ; PACCIARELLI, D. ; PRANZO, M.: Reordering and local rerouting strategies to manage train traffic in real-time. In: *Transportation Science* Bd. 42 (2008), Nr. 4, S. 405–419
- [27] D'ARIANO, A. ; PACCIARELLI, D. ; PRANZO, M.: A branch and bound algorithm for scheduling trains in a railway network. In: *European Journal of Operational Research* Bd. 183 (2007), Nr. 2, S. 643–657
- [28] *Das Unternehmen KVB - Bus*. URL <http://www.kvb-koeln.de/german/unternehmen/kvb/bus.html>. - abgerufen am 2015-12-09
- [29] DEMEULEMEESTER, E. ; HERROELEN, W. ; ELMAGHRABY, S.: Optimal procedures for the discrete time/cost trade-off problem in project networks. In: *European Journal of Operational Research* Bd. 88 (1996), Nr. 1, S. 50–68
- [30] DESAULNIERS, G. ; HICKMAN, M.: Public Transit. In: *Handbooks in Operations Research and Management Science*. Bd. 14 : Elsevier, 2007 — ISBN 978-0-444-51346-5, S. 69–127
- [31] DONIEC, A. ; MANDIAU, R. ; PIECHOWIAK, S. ; ESPIÉ, S.: A behavioral multi-agent model for road traffic simulation. In: *Engineering Applications of Artificial Intelligence* Bd. 21 (2008), Nr. 8, S. 1443–1454
- [32] DREO, J. ; PETROWSKI, A. ; SIARRY, A. ; TAILLARD, E.: *Metaheuristics for Hard Optimization* : Springer, 2006
- [33] EISSFELDT, N.: *Vehicle-based modelling of traffic. Theory and application to environmental impact modelling*, Universität zu Köln, Dissertation, 2004
- [34] ERANKI, A.: *A model to create bus timetables to attain maximum synchronization considering waiting times at transfer stops*. Tampa, University of South Florida, Masterarbeit, 2004
- [35] FAY, A.: A fuzzy knowledge-based system for railway traffic control. In: *Engineering Applications of Artificial Intelligence* Bd. 13 (2000), Nr. 6, S. 719–729
- [36] FRANZ, S.: *GenetiBounds - Entwurf und Entwicklung eines mehrstufigen Optimierungsverfahrens für Stadtbahnfahrpläne unter Berücksichtigung verkehrsplanerischen Vorgaben*. Köln, Universität zu Köln, Diplomarbeit, 2011
- [37] FU, L. ; LIU, Q. ; CALAMAI, P.: Real-time optimization model for dynamic scheduling of transit operations. In: *Transportation Research Record: Journal of the Transportation Research Board* Bd. 1857 (2003), Nr. 1, S. 48–55
- [38] FUJIMOTO, R.: *Parallel and Distributed Simulation Systems*. 1. Aufl. New York, NY : Wiley, 2000 — ISBN 0-471-18383-0
- [39] GENÇ, Z.: *Ein neuer Ansatz zur Fahrplanoptimierung im ÖPNV: Maximierung von zeitlichen Sicherheitsabständen*. Köln, Universität zu Köln, Dissertation, 2003
- [40] GINKEL, A. ; SCHÖBEL, A.: To wait or not to wait? The bicriteria delay management problem in public transportation. In: *Transportation Science* Bd. 41 (2007), Nr. 4, S. 527–538
- [41] GÓMEZ, P.: *A robust schedule for Montpellier's extended Tramway network of 2017*. Köln, Universität zu Köln, Bachelorarbeit, 2014
- [42] HADAS, Y. ; CEDER, A.: Optimal coordination of public-transit vehicles using operational tactics examined by simulation. In: *Transportation Research Part C: Emerging Technologies* Bd. 18 (2010), Nr. 6, S. 879–895
- [43] HEALTH MARKET SCIENCE: *Jackcess*. URL <http://jackcess.sourceforge.net/>. - abgerufen am 2015-11-28
- [44] *HSQldb*. URL <http://hsqldb.org/>. - abgerufen am 2015-11-28
- [45] HUANG, Y. ; SECK, M. D. ; VERBRAECK, A.: Component-based light-rail modeling in discrete event systems specification (DEVS). In: *SIMULATION* Bd. 91 (2015), Nr. 12, S. 1027–1051

- [46] IBARRA-ROJAS, O. ; RIOS-SOLIS, Y.: Synchronization of bus timetabling. In: *Transportation Research Part B: Methodological* Bd. 46 (2012), Nr. 5, S. 599–614
- [47] INGALLS, R.: Introduction to simulation. In: *Proceedings of the 40th Conference on Winter Simulation* : Winter Simulation Conference, 2008, S. 17–26
- [48] JAVA PLATFORM, STANDARD EDITION 8, API SPECIFICATION: *PriorityQueue*. URL <https://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html>. - abgerufen am 2015-12-01
- [49] JOISTEN, M.: *Simulation von Fahrplänen für den ÖPNV mittels Zellularautomaten*. Köln, Universität zu Köln, Diplomarbeit, 2002
- [50] KASTNER, K.-H. ; KEBER, R. ; PAU, P. ; SAMAL, M.: Real-Time Traffic Conditions with SUMO for ITS Austria West. In: *1st SUMO User Conference 2013, Berichte aus dem DLR-Institut für Verkehrstechnik*. Bd. 21. Berlin, 2013, S. 43–53
- [51] KASTNER, K.-H. ; PAU, P.: Experiences with SUMO in a Real-Life Traffic Monitoring System. In: *SUMO 2015 - Intermodal Simulation for Intermodal Transport, Berichte aus dem DLR-Institut für Verkehrstechnik*. Bd. 28. Berlin, 2015, S. 1–10
- [52] KENDZIORRA, A. ; WEBER, M.: Extensions for logistic and public transport in SUMO. In: *SUMO 2015 - Intermodal Simulation for Intermodal Transport, Berichte aus dem DLR-Institut für Verkehrstechnik*. Bd. 28. Berlin, 2015, S. 83–90
- [53] KLIEWER, N. ; SUHL, L.: A note on the online nature of the railway delay management problem. In: *Networks* Bd. 57 (2011), Nr. 1, S. 28–37
- [54] KRAJZEWICZ, D. ; ERDMANN, J. ; BEHRISCH, M. ; BIEKER, L.: Recent Development and Applications of SUMO—Simulation of Urban MObility. In: *International Journal On Advances in Systems and Measurements* Bd. 5 (2012), Nr. 3 and 4, S. 128–138
- [55] KRAJZEWICZ, D. ; HERTKORN, G. ; RÖSSEL, C. ; WAGNER, P.: Sumo (simulation of urban mobility). In: *Proc. of the 4th Middle East Symposium on Simulation and Modelling*, 2002, S. 183–187
- [56] LIEBCHEN, C. ; SCHACHTEBECK, M. ; SCHÖBEL, A. ; STILLER, S. ; PRIGGE, A.: Computing delay resistant railway timetables. In: *Computers & Operations Research* Bd. 37 (2010), Nr. 5, S. 857–868
- [57] LIEBL, F.: *Simulation*. 2. Aufl. München : Oldenbourg Verlag, 1995 — ISBN 3-486-23373-4
- [58] LÜCKEMEYER, G.: *A Traffic Simulation System Increasing the Efficiency of Schedule Design for Public Transport Systems Based on Scarce Data*. Aachen, Shaker Verlag, Dissertation, 2007
- [59] LÜCKEMEYER, G. ; SPECKENMEYER, E.: Comparing Applicability of Two Simulation Models in Public Transport Simulation. In: BECKER, M. ; SZCZERBICKA, H. (Hrsg.): *ASIM 2006 - 19. Symposium Simulationstechnik* : ASIM/Universität Hannover, 2006
- [60] LÜCKERATH, D.: *ParSiVaL - Entwurf und Entwicklung einer Anwendung zur parallelen Simulation von schienengebundenem Öffentlichen Personennahverkehr*. Köln, Universität zu Köln, Diplomarbeit, 2011
- [61] LÜCKERATH, D. ; ULLRICH, O. ; KUPICHA, A. ; SPECKENMEYER, E.: Multi-depot multi-vehicle-type vehicle scheduling for Cologne’s tram network. In: SCHEIBLE, J. ; BAUSCH-GALL, I. ; DEATCU, C. (Hrsg.) *Proceedings of ASIM-Workshop STS/GMMS 2014, ARGESIM Report 42, ASIM-Mitteilung AM 149, ARGESIM/ASIM Pub., TU Vienna/Austria* (2014), S. 191–197
- [62] LÜCKERATH, D. ; ULLRICH, O. ; SPECKENMEYER, E.: Modeling time table based tram traffic. In: *Simulation Notes Europe* Bd. 22 (2012), Nr. 2, S. 61–68
- [63] LÜCKERATH, D. ; ULLRICH, O. ; SPECKENMEYER, E.: Applicability of rescheduling strategies in tram networks. In: REICHARDT, R. (Hrsg.) *Proceedings of ASIM-Workshop STS/GMMS 2013 - ARGESIM Report 41, ASIM-Mitteilung 145, ARGESIM/ASIM Pub. TU Vienna* (2013), S. 57–64
- [64] MACAL, C. M. ; NORTH, M. J.: Introductory Tutorial: Agent-based modeling and simulation. In: *Winter Simulation Conference (WSC) 2013* : IEEE, 2013, S. 362–376
- [65] MARCO, A.: *UCanAccess*. URL <http://ucanaccess.sourceforge.net/site.html>. - abgerufen am 2015-11-28
- [66] MARIA, A.: Introduction to modeling and simulation. In: *Proceedings of the 1997 Winter Simulation Conference* : IEEE Computer Society, 1997, S. 7–13
- [67] MEIGNAN, D. ; SIMONIN, O. ; KOUKAM, A.: Simulation and evaluation of urban bus-networks using a multiagent approach. In: *Simulation Modelling Practice and Theory* Bd. 15 (2007), Nr. 6, S. 659–671

- [68] NAGEL, K.: Multi-modal traffic in TRANSIMS. In: *Pedestrian and evacuation dynamics* (2001), S. 161–172
- [69] NAGEL, K. ; SCHRECKENBERG, M.: A cellular automaton model for freeway traffic. In: *Journal de Physique I* Bd. 2 (1992), Nr. 12, S. 2221–2229
- [70] NESHELI, M. ; CEDER, A.: A robust, tactic-based, real-time framework for public-transport transfer synchronization. In: *Transportation Research Part C: Emerging Technologies* Bd. 60 (2015), S. 105–123
- [71] VAN ORT, N. ; VAN NES, R.: Regularity analysis for optimizing urban transit network design. In: *Public Transport* Bd. 1 (2009), Nr. 2, S. 155–168
- [72] OSOGAMI, T. ; IMAMICHI, T. ; MIZUTA, H. ; MORIMURA, T. ; RAYMOND, R. ; SUZUMURA, T. ; TAKAHASHI, R. ; IDÉ, T.: *Research Report IBM Mega Traffic Simulator* : Technical report, 2012
- [73] ROSSI, F. ; VAN BEEK, P. ; WALSH, T. (Hrsg.): *Handbook of constraint programming* : Elsevier, 2006
- [74] RUSSEL ; NORVIG: *Artificial Intelligence : A modern approach* : Prentice Hall, 2002
- [75] SAHARIDIS, G. ; DIMITROPOULOS, C. ; SKORDILIS, E.: Minimizing waiting times at transitional nodes for public bus transportation in Greece. In: *Operational Research* Bd. 14 (2014), Nr. 3, S. 341–359
- [76] SALIDO, M. ; GIRET, A.: Feasible distributed CSP models for scheduling problems. In: *Engineering Applications of Artificial Intelligence* Bd. 21 (2008), Nr. 5, S. 723–732
- [77] SCHMIDT, M. ; GOERIGK, M. ; MÜLLER-HANNEMANN, M. ; SCHÖBEL, A.: Robuste Fahrplanauskunft. In: *Proceedings of Heureka '14 - Optimierung in Verkehr und Transport*. Stuttgart : FGSV Verlag, 2014, S. 466–483
- [78] SCHNABEL, W. ; LOHSE, D.: *Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung*. Bd. 2 v. 2. 3. Aufl. Berlin/Bonn : Beuth Verlag/Kirschbaum Verlag — ISBN 978-3-7812-1816-1
- [79] SCHNABEL, W. ; LOHSE, D.: *Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung*. Bd. 1 v. 2. 3. Aufl. Berlin/Bonn : Beuth Verlag/Kirschbaum Verlag — ISBN 978-3-7812-1815-4
- [80] SCHÜLE, I. ; SCHRÖDER, M. ; HANSEN, N.: Fahrplansynchronisierung im öffentlichen Nahverkehr - Mathematische Optimierung zur Verbesserung komplexer Abstimmungsprozesse. In: *Heureka '11 Optimierung in Verkehr und Transport* : FGSV-Verlag, 2011 — ISBN 978-3-941790-72-8, S. 472–485
- [81] SPECKENMEYER, E. ; LI, N. ; LÜCKERATH, D. ; ULLRICH, O.: *Socio-Economic Objectives in Tram Scheduling* (Technical Report). Köln : Universität zu Köln, 2012
- [82] STAHLBERG, M.: *Ein Verfahren zum Lösen des Fahrplan-Problems mit verkehrsplanerischen Vorgaben bei nicht-konvexem Lösungsraum*. Köln, Universität zu Köln, Masterarbeit, 2015
- [83] BEHRISCH, M. ; KNOCKE, M. (Hrsg.): *SUMO2014 - Modeling Mobility with Open Data, Berichte aus dem DLR-Institut für Verkehrssystemtechnik* ( Nr. 24). Berlin : Deutsches Zentrum für Luft- und raumfahrt, 2014
- [84] BEHRISCH, M. ; WEBER, M. (Hrsg.): *SUMO 2015 - Intermodal Simulation for Intermodal Transport, Berichte aus dem DLR-Institut für Verkehrssystemtechnik* ( Nr. 28). Berlin : Deutsches Zentrum für Luft- und raumfahrt, 2015
- [85] SUZUMURA, T. ; KANEZASHI, H.: Multi-modal traffic simulation platform on parallel and distributed systems. In: TOLK, A. ; DIALLO, S. Y. ; RYZHOV, I. O. ; YILMAZ, L. ; BUCKLEY, S. ; MILLER, J. A. (Hrsg.): *Proceedings of the 2014 Winter Simulation Conference* : IEEE Press, 2014, S. 769–780
- [86] SUZUMURA, T. ; MCARDLE, G. ; KANEZASHI, H.: A high performance multi-modal traffic simulation platform and its case study with the Dublin city. In: YILMAZ, L. ; CHAN, W. K. V. ; MOON, I. ; ROEDER, T. M. K. ; MACAL, C. ; ROSSETTI, M. D. (Hrsg.): *Proceedings of the 2015 Winter Simulation Conference* : IEEE Press, 2015, S. 767–778
- [87] TIAN, L.: Traffic Flow Simulation in a Scenario with Signalized Intersection and Bus Stop. In: *Journal of Transportation Systems Engineering and Information Technology* Bd. 12 (2012), Nr. 5, S. 90–96
- [88] TOLEDO, T. ; CATS, O. ; BURGHOUT, W. ; KOUTSOPOULOS, H. N.: Mesoscopic simulation for transit operations. In: *Transportation Research Part C: Emerging Technologies* Bd. 18 (2010), Nr. 6, S. 896–908

- [89] ULLRICH, O.: *Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz*. Köln, Universität zu Köln, Dissertation, 2014
- [90] ULLRICH, O. ; LÜCKERATH, D. ; FRANZ, S. ; SPECKENMEYER, E.: Simulation and optimization of Cologne's tram schedule. In: *Simulation Notes Europe* Bd. 22 (2012), Nr. 2, S. 69–76
- [91] ULLRICH, O. ; LÜCKERATH, D. ; SPECKENMEYER, E.: Reduzieren robuste Fahrpläne Verspätungen in Stadtbahnnetzen? - Es kommt drauf an. In: *Proceedings of Heureka' 14 - Optimierung in Verkehr und Transport*. Stuttgart : FGSV Verlag, 2014, S. 448–465
- [92] ULLRICH, O. ; LÜCKERATH, D. ; SPECKENMEYER, E.: Model-based parallelization of discrete traffic simulation models. In: *Simulation Notes Europe* Bd. 24 (2014), Nr. 3–4, S. 115–122
- [93] ULLRICH, O. ; LÜCKERATH, D. ; SPECKENMEYER, E.: A robust schedule for Montpellier's Tramway network. In: *Simulation Notes Europe* Bd. 25 (2015), Nr. 1, S. 1–8
- [94] ULLRICH, O. ; LÜCKERATH, D. ; SPECKENMEYER, E.: Do regular time tables help to reduce delays in tram networks? - It depends! In: *Public Transport* Bd. 8 (2016), Nr. 1, S. 39–56
- [95] ULLRICH, O. ; PROFF, I. ; LÜCKERATH, D. ; KUCKERTZ, P. ; SPECKENMEYER, E.: Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation. In: BUSCH, F. ; SPANGLER, M. (Hrsg.) *ITS for Connected Mobility - Selected Papers from the mobil.TUM 2013 International Conference on Mobility and Transport* (2014), S. 89–98
- [96] VERBAND DEUTSCHER VERKEHRSUNTERNEHMEN EV: *VDV-Standardschnittstelle Liniennetz/Fahrplan, VDV-Schriften* ( Nr. 452), 2008
- [97] VINCE, A.: Scheduling periodic events. In: *Discrete Applied Mathematics* Bd. 25 (1989), Nr. 3, S. 299–310
- [98] VOSSLOH KIEPE GMBH: *Elektrische Ausrüstung der Niederflur-Stadtbahnwagen K4000 der Kölner Verkehrs-Betriebe AG*. (Druckschrift Nr. 00KV7DE), 2003
- [99] VOSSLOH KIEPE GMBH: *Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG* (Druckschrift Nr. 00KB5DE), 2003
- [100] VOSSLOH KIEPE GMBH: *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4500 für die Kölner Verkehrs-Betriebe AG* (Druckschrift Nr. 00KN2DE), 2003
- [101] WALTER, D.: *Optimierung von ÖPNV-Fahrplänen unter Berücksichtigung verkehrsplanerischer Vorgaben*. Köln, Universität zu Köln, Diplomarbeit, 2010
- [102] WEIMAR, J.: *Simulation with Cellular Automata*. 2. Aufl. Berlin : Logos Verlag, 2003 — ISBN 3-89722-026-1
- [103] WU, Y. ; TANG, J. ; YU, Y. ; PAN, Z.: A stochastic optimization model for transit network timetable design to mitigate the randomness of traveling time by adding slack time. In: *Transportation Research Part C: Emerging Technologies* Bd. 52 (2015), S. 15–31
- [104] YAN, Y. ; MENG, Q. ; WANG, S. ; GUO, X.: Robust optimization model of schedule design for a fixed bus route. In: *Transportation Research Part C: Emerging Technologies* Bd. 25 (2012), S. 113–121
- [105] YANG, Q. ; KOUTSOPOULOS, H. N. ; BEN-AKIVA, M. E.: *A Simulation Laboratory for Evaluating Dynamic Traffic Management Systems, Transportation Research Record* ( Nr. 00–1688), 1999
- [106] YU, B. ; YANG, Z.: A dynamic holding strategy in public transit systems with real-time information. In: *Applied Intelligence* Bd. 31 (2009), Nr. 1, S. 69–80
- [107] YU, B. ; YANG, Z. ; LI, S.: Real-time partway deadheading strategy based on transit service reliability assessment. In: *Transportation Research Part A: Policy and Practice* Bd. 46 (2012), Nr. 8, S. 1265–1279
- [108] ZAMITH, M. ; LEAL-TOLEDO, R. ; CLUA, E. ; TOLEDO, E. ; MAGALHÃES, G.: A new stochastic cellular automata model for traffic flow simulation with drivers' behavior prediction. In: *Journal of Computational Science* Bd. 9 (2015), S. 51–56
- [109] ZOLFAGHARI, S. ; AZIZI, N. ; JABER, M. Y.: A model for holding strategy in public transit systems with real-time information. In: *International Journal of Transport Management* Bd. 2 (2004), Nr. 2, S. 99–110

## Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Ewald Speckenmeyer betreut worden.

(Daniel Lückcrath)

- Ullrich, O., Lückcrath, D., Speckenmeyer, E.: *Do regular time tables help to reduce delays in tram networks? - It depends!* In: Public Transport, Springer-Verlag Berlin Heidelberg, Volume 8, Number 1, März 2016, pp. 39-56, ISSN 1866-749X, DOI 10.1007/s12469-015-0115-6.
- Lückcrath, D., Ullrich, O., Rische, N.: *System and Method for Intelligent, Demand-Responsive Transit Recommendation*. U.S. Patent pending, appl. no. 14/991,341, filed January 8, 2016.
- Ullrich, O., Lückcrath, D., Rische, N.: *Dynamic Routing of Transit Vehicles*. U.S. Patent pending, appl. no. 14/829,021, filed March 17, 2015.
- Ullrich, O., Lückcrath, D., Speckenmeyer, E.: *A robust schedule for Montpellier's Tramway network*. In: Simulation Notes Europe (SNE), ARGESIM/ Asim Pub., TU Vienna/Austria, Volume 25, Number 1, 2015, pp. 1-8, ISSN 2306-0271.
- Ullrich, O., Lückcrath, D., Speckenmeyer, E.: *Model-based parallelization of discrete simulation models*. In: Simulation Notes Europe (SNE), ARGESIM/ Asim Pub., TU Vienna/Austria, Volume 24, Number 3-4, 2014, pp. 115-122, ISSN 2306-0271.
- Ullrich, O., Lückcrath, D., Speckenmeyer, E.: *Reduzieren robuste Fahrpläne Verspätungen in Stadtbahnnetzen? - Es kommt drauf an!* In: Heureka' 14 Optimierung in Verkehr und Transport, FGSV Verlag, Stuttgart, April 2014, pp. 448-465.
- Lückcrath, D., Ullrich, O., Kupicha, A., Speckenmeyer, E.: *Multi-depot multi-vehicle-type vehicle scheduling for Cologne's tram network*. In: Proceedings of ASIM-Workshop STS/GMMS 2014, ARGESIM Report 42, ASIM-Mitteilung AM 149, ARGESIM/ASIM Pub., TU Vienna/Austria, Scheible, J.; Bausch-Gall, I., Deatcu, C. (Ed.), Februar 2014, pp. 191-197.
- Ullrich, O., Proff, I., Lückcrath, D., Kuckertz, P., Speckenmeyer, E.: *Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation*. In: ITS for Connected Mobility - Selected Papers from the mobil.TUM 2013 International Conference on Mobility and Transport, Busch, F., Spangler, M. (Ed.), 2014, pp. 89-98.
- Lückcrath, D., Ullrich, O., Speckenmeyer, E.: *Applicability of rescheduling strategies in tram networks*. In: Proceedings of ASIM-Treffen STS/GMMS 2013, ARGESIM Report 41, ASIM-Mitteilung AM 145, ARGESIM/ASIM Pub., TU Vienna/Austria, Reichardt, R. (Ed.), Februar 2013, 7 pg.
- Lückcrath, D.: *Thoughts on restoration of regular tram operation*. In: Proceedings of Sommer-treffen Verkehrssimulation 2012, ASIM-Mitteilung AM 143, ARGESIM/ASIM Pub., TU Vienna/Austria, Ullrich, O., Lückcrath, D. (Ed.), August 2012, pp. 5-7.
- Ullrich, O., Lückcrath, D., Franz, S., Speckenmeyer, E.: *Simulation and optimization of Cologne's tram schedule*. In: Simulation Notes Europe (SNE), ARGESIM/ Asim Pub., TU Vienna/Austria, Volume 22, Number 2, August 2012, pp. 69-76, ISSN 2305-9974.
- Lückcrath, D., Ullrich, O., Speckenmeyer, E.: *Modeling time table based tram traffic*. In: Simulation Notes Europe (SNE), ARGESIM/ Asim Pub., TU Vienna/Austria, Volume 22, Number 2, August 2012, pp. 61-68, ISSN 2305-9974.

## Lebenslauf

- 10.04.1986            Geboren in Köln
- 07/2005              Abitur an der Kaiserin-Theophanu-Schule in Köln
- 10/2005 – 04/2011    Studium der Wirtschaftsinformatik an der Universität zu Köln  
Abschluss mit Diplom
- 05/2011 – 08/2013    Wissenschaftlicher Mitarbeiter des Instituts für Nachrichtentechnik der Technischen Hochschule Köln
- 10/2014 – 04/2015    Gastwissenschaftler an der School of Computing and Information Sciences der Florida International University
- 05/2011 – 07/2016    Wissenschaftlicher Mitarbeiter des Instituts für Informatik der Universität zu Köln
- 12/2016              Abschluss der Dissertation