



FRIEDRICH-ALEXANDER  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
TECHNISCHE FAKULTÄT

# Lehrstuhl für Informatik 7

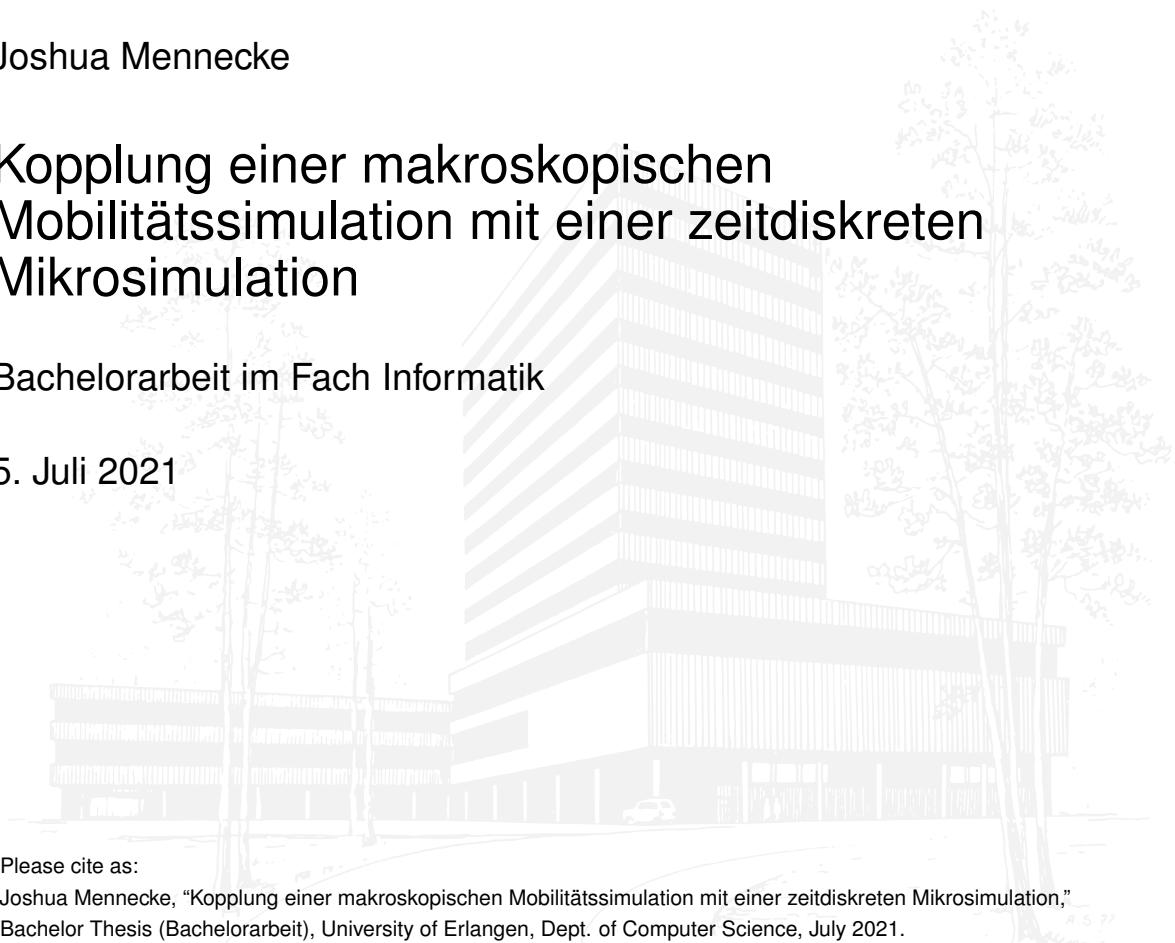
## Rechnernetze und Kommunikationssysteme

Joshua Mennecke

# Kopplung einer makroskopischen Mobilitätssimulation mit einer zeitdiskreten Mikrosimulation

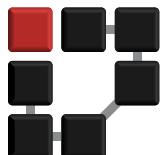
Bachelorarbeit im Fach Informatik

5. Juli 2021



Please cite as:

Joshua Mennecke, "Kopplung einer makroskopischen Mobilitätssimulation mit einer zeitdiskreten Mikrosimulation,"  
Bachelor Thesis (Bachelorarbeit), University of Erlangen, Dept. of Computer Science, July 2021.



Friedrich-Alexander-Universität Erlangen-Nürnberg  
Department Informatik  
Rechnernetze und Kommunikationssysteme  
Martensstr. 3 · 91058 Erlangen · Germany  
<http://www7.cs.fau.de/>

# **Kopplung einer makroskopischen Mobilitätssimulation mit einer zeitdiskreten Mikrosimulation**

Bachelorarbeit im Fach Informatik

vorgelegt von

**Joshua Mennecke**

geb. am 12. Juli 1999  
in Erfurt

angefertigt am

**Lehrstuhl für Informatik 7  
Rechnernetze und Kommunikationssysteme**

**Department Informatik  
Friedrich-Alexander-Universität Erlangen-Nürnberg**

Betreuer:                   **Prof. Dr.-Ing. Reinhard German**  
                                 **Moritz Gütlein, M. Sc.**

Abgabe der Arbeit:   **5. Juli 2021**

## Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.

Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

## Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.

I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Joshua Mennecke)

Erlangen, 5. Juli 2021

---

## Kurzfassung

---

Die Verwendung von Simulatoren geht in den meisten Fällen mit einem Kompromiss zwischen Rechenaufwand und Genauigkeit einher, abhängig von der verwendeten Abstraktionsebene. Um bei Verkehrssimulationen möglichst die Vorteile zu kombinieren und Nachteile minimieren zu können soll in dieser Bachelorarbeit ein entwickelter hybrider Ansatz zweier Simulationsprogramme beschrieben und evaluiert werden. Das Ziel ist es dabei, große Verkehrsgebiete mit begrenztem Berechnungsaufwand, bestimmte Teilnetze oder Kreuzungen jedoch trotzdem mit hoher Genauigkeit und konkreten Fahrzeugobjekten darstellen und untersuchen zu können. Dafür wird auf makroskopischer Ebene beispielhaft der Simulator "Verkehr in Städten Umgebung (VISUM)" und auf der mikroskopischen "simulation of urban mobility (SUMO)" verwendet.

Die größte Herausforderung liegt an den Übergangsstellen der Programme, an denen die unterschiedlichen Simulationsdaten umgewandelt werden müssen. Für die Kopplung wurden deshalb Python-Skripte entwickelt, die diese konsistente Umwandlung in beide Richtung sicherstellen und dabei zum Einspeisen beziehungsweise Auslesen von Daten Zugriff auf die Simulatoren besitzen. An der Verbindungsstelle vom Makroskopischen zum Mikroskopischen werden aus aggregierten Werten des Verkehrsflusses einzelne Fahrzeuge mit individuellen Routen und Abfahrtszeiten erzeugt. Diese müssen an den Strecken von SUMO nach VISUM wieder gezählt und aggregiert in das makroskopische Programm eingespeist werden.

Die Evaluation anhand verschiedener Beispielszenarien zeigt besonders bei hohem Verkehrsaufkommen eine starke Annäherung des Verhaltens der Kopplung an das einer ausschließlich makroskopischen Simulation. Auch die Änderung der Umlegungen in VISUM anhand von genaueren Verkehrsdaten aus SUMO hat sich als zuverlässig erwiesen. Lediglich bei geringem Verkehrsaufkommen sind starke Schwankungen der Werte der Kopplung festzustellen, wodurch möglicherweise eine rein makroskopische Simulation vorgezogen werden sollte.

Die Qualität und Realitätsnähe der Kopplung könnte durch Optimierungen weiter verbessert werden. Mögliche Ansätze dazu werden ebenfalls ausführlich beschrieben.

---

# Inhaltsverzeichnis

---

|   |            |
|---|------------|
| <b>Kurzfassung</b>  | <b>iii</b> |
| <b>1 Einführung</b>   | <b>1</b>   |
| <b>2 Grundlagen</b>   | <b>3</b>   |
| 2.1 Verwandte Arbeiten . . . . .                                | 3          |
| 2.2 Beispielhaft verwendete Simulatoren . . . . .               | 6          |
| 2.2.1 Makroskopisches Simulationsprogramm: VISUM . . . . .      | 6          |
| 2.2.1.1 Benutzeroberfläche . . . . .                            | 7          |
| 2.2.1.2 OD-Matrizen und Umlegungen . . . . .                    | 8          |
| 2.2.1.3 COM-Schnittstelle . . . . .                             | 10         |
| 2.2.2 Mikroskopisches Simulationsmodell: SUMO . . . . .         | 12         |
| 2.2.2.1 Benutzeroberfläche und Dateitypen . . . . .             | 12         |
| 2.2.2.2 Traci-Schnittstelle . . . . .                           | 13         |
| 2.3 Exponentialverteilung . . . . .                             | 14         |
| <b>3 Entwickelte Architektur</b>                                | <b>17</b>  |
| 3.1 Verwendete Soft- und Hardware . . . . .                     | 17         |
| 3.2 Theoretischer Ansatz . . . . .                              | 18         |
| 3.2.1 Aufbau der Simulatorenkopplung . . . . .                  | 18         |
| 3.2.2 Pfadplanung über Abbiegewahrscheinlichkeiten . . . . .    | 20         |
| 3.2.3 Kapazitätsbeschränkungen zur Umlegungsanpassung . . . . . | 22         |
| 3.2.4 Festgelegte Vereinfachungen . . . . .                     | 26         |
| 3.3 Übersicht über die entwickelte Kopplung . . . . .           | 28         |
| 3.4 Implementierung . . . . .                                   | 29         |
| 3.4.1 Controller . . . . .                                      | 29         |
| 3.4.2 VISUM nach SUMO . . . . .                                 | 30         |
| 3.4.3 SUMO nach VISUM . . . . .                                 | 32         |
| 3.4.4 SUMO . . . . .  | 33         |
| 3.4.5 Utility . . . . .   | 33         |

---

**Inhaltsverzeichnis** **v**

|   |           |
|---|-----------|
| <b>4 Evaluation</b>                                       | <b>34</b> |
| 4.1 Funktionsweisen der einzelnen Komponenten . . . . .   | 34        |
| 4.1.1 Szenario a): Bezirke und Kreuzungen . . . . .       | 35        |
| 4.1.1.1 Fahrzeugerzeugung . . . . .                       | 37        |
| 4.1.1.2 Routenplanung . . . . .                           | 40        |
| 4.1.2 Szenario b): Streckensperrung in SUMO . . . . .     | 44        |
| 4.2 Verwendung mehrerer mikroskopischer Gebiete . . . . . | 51        |
| <b>5 Fazit</b>  | <b>53</b> |
| <b>6 Anhang</b>   | <b>62</b> |

---

## Kapitel 1

---

### Einführung

---

Simulationen werden heutzutage in vielen Bereichen eingesetzt, da sie auch bei komplexen Problemstellungen vergleichsweise kostengünstig und schnell umsetzbar sind. Außerdem können sie leicht an sich verändernde Voraussetzungen angepasst werden und ermöglichen die Modellierung und Analyse vieler relevanter Grenzfälle. Die auf einer solchen Analyse beruhenden Entscheidungen wurden dann bereits in einem von der Wirklichkeit abstrahierten Modell getestet und sind daher in den meisten Fällen relativ zuverlässig.

Laut [Fah] ist allein in den Jahren 2005 bis 2017 die Fahrleistung aller PKW privater Haushalte in Deutschland von 482 um fast 8% auf ungefähr 520 Milliarden Kilometer gestiegen. Das zunehmende Verkehrsaufkommen auf den Straßen macht den Ausbau des Streckennetzes erforderlich. Jedoch ist die Analyse des Verkehrs bereits heute aufgrund der großen Anzahl von Straßen und der Individualität einzelner Fahrzeuge sehr komplex.

In den Bereichen Verkehr und Verkehrsplanung werden deshalb bereits seit einigen Jahren Simulationen eingesetzt. Durch Modelle können noch in der Planungsphase die voraussichtlichen Auswirkungen geplanter Bauvorhaben simuliert und nach anschließender Analyse Optimierungen daran vorgenommen werden. Mobilitätsimulationen lassen sich unterschiedlichen Abstraktionsebenen zuordnen. Da sich die Ebenen vor allem in der Genauigkeit und dem Rechenaufwand unterscheiden, muss bei der Wahl häufig ein Kompromiss getroffen werden. Es ist jedoch auch möglich zur Vermeidung von diesem Simulatoren mehrerer Ebenen miteinander zu verknüpfen. In dieser Arbeit soll ein solches System mit Simulationsprogrammen zweier Abstraktionsebenen aufgebaut werden. Es wurden zwei konkrete Simulatoren ausgewählt, um diesen hybriden Ansatz anschließend testen zu können.

Der Fokus der Arbeit liegt dabei auf der Umwandlung von Simulationsattributen und -werten an den Übergangsstellen zwischen den Programmen. Dabei soll das konsistente Verhalten der beiden Simulatoren miteinander sowie im Vergleich zu einer

ausschließlich makroskopischen Variante sichergestellt werden. Die Anwendbarkeit sowie die Erhaltung der Konsistenz sollen im Anschluss noch an Beispielszenarien überprüft werden.

Im Folgenden wird der grobe Aufbau dieser Arbeit kurz erläutert. Zunächst werden in Kapitel 2 verschiedene bereits veröffentlichte Ansätze beschrieben und eine Abgrenzung zum Inhalt dieser Arbeit gezogen. Außerdem sollen Hintergrundinformationen zu mathematischen Grundlagen einer Funktion gegeben werden. Darauf folgt eine Beschreibung des entwickelten Systems, wobei der Aufbau und die Funktionsweise zunächst in einem kurzen Überblick und schließlich komponentenweise genauer erläutert werden soll. In Kapitel 4 wird das Modell anhand von verschiedenen künstlich erstellten Beispielszenarien analysiert, wobei mehrere unterschiedliche Fälle betrachtet werden sollen. Zum Schluss folgt ein kurzes Fazit der Arbeit sowie ein Ausblick auf mögliche Erweiterungen beziehungsweise Optimierungen der entwickelten Kopplung.

---

## Kapitel 2

---

# Grundlagen

---

Dieses Kapitel soll ein grundlegendes Verständnis von Verkehrssimulationen vermitteln und darüber hinaus verwendete Methoden erläutern. Dafür werden zunächst verwandte wissenschaftliche Arbeiten beschrieben und zusammengefasst. Im Anschluss sollen die beiden verwendeten Simulationsprogramme "Verkehr in Städten Umlegung (VISUM)" und "simulation of urban mobility (SUMO)" kurz vorgestellt und ihre Auswahl begründet werden. Im letzten Abschnitt folgen schließlich mathematische Grundlagen zur Exponentialverteilung und Erläuterungen ihrer Rolle an den Übergangsstellen zwischen beiden Simulatoren.

### 2.1 Verwandte Arbeiten

Verkehrssimulationen lassen sich nach [AKM14] im Wesentlichen einer der drei folgenden Abstraktionsebenen zuordnen:

- **makroskopisch:** Betrachtung von Verkehrsflüssen als große Fahrzeuggruppen, die aggregierte Simulationsattribute wie die Auslastungen von Netzobjekten pro Zeiteinheit besitzen.
- **mesoskopisch:** Sich ähnlich verhaltende Fahrzeuge werden zu Gruppen zusammengefasst. Arbeitet ebenfalls mit aggregierten Werten, allerdings auf einer geringeren Objektanzahl.
- **mikroskopisch:** Verwendet konkrete Fahrzeugobjekte, die sich, zum Beispiel Geschwindigkeit oder Routenwahl betreffend, individuell unterschiedlich verhalten können.

Teilweise ist noch von einer als "submicroscopic" bezeichneten vierten Ebene die Rede, die einzelne Elemente eines Fahrzeugs wie beispielsweise die Schaltung individuell simuliert [Lop+18]. Diese Abstraktionsebene wird in dieser Arbeit jedoch

außer Acht gelassen. Durch die zu hohe Genauigkeit und den damit verbundenen wesentlich höheren Rechenaufwand ist sie für eine Kopplung mit einem Simulator auf makroskopischen Ebene nicht sehr gut geeignet.

Die Simulatoren der verschiedenen Ebenen unterscheiden sich also vor allem in den betrachteten Objekten und den Simulationsattributen. Außerdem beschränkt deren Genauigkeit aufgrund der begrenzt zur Verfügung stehenden Rechenkapazität auch die maximal simulierbare Fläche. Die wichtigsten Unterschiede der verschiedenen Ebenen sind im Anhang in Tabelle 6.1 noch einmal tabellarisch gegenüber gestellt. Darüber hinaus werden in den einzelnen Abstraktionsebenen unterschiedliche Verfahren zur Modellierung verwendet. So ist im mikroskopischen Bereich der Einsatz von unter anderem car-following- oder Cellular Automata (CA) Modellen verbreitet. Eine Auflistung und Differenzierung der unterschiedlichen Verkehrsflussmodelle ist nicht Ziel der Arbeit und wird daher an dieser Stelle nicht näher ausgeführt. Eine ausführliche Übersicht ist in [WK+15] zu finden.

Im folgenden sollen nun die Ansätze von bereits entwickelten Simulatorenkopplungen beschrieben werden. Vor allem in den letzten 20 Jahren hat sich das Thema "Kopplungen von Verkehrssimulationen" durch verschiedene Veröffentlichungen stark weiterentwickelt. Bereits 2004 entwickelte Burghout im Zuge seiner Doktorarbeit den hybriden Ansatz einer Verkehrssimulation auf mikro- und meso-Ebene [Bur04]. Er verknüpfte dabei das mikroskopische Simulationsmodell "MITSIMLab" mit dem neu entwickelten mesoskopischen Modell "Mezzo" und legte den Fokus vor allem auf die Konsistenz der Daten zwischen beiden Simulatoren. Die Evaluation anhand verschiedener Teilnetze zeigte die Anwendbarkeit dieser Kopplung und die Annäherung an das reale Verkehrsgeschehen.

Eine weitere hybride Variante veröffentlichten 2004 El Hmam et al. und beschrieben dabei die Kopplung von makro- und mikro-Simulatoren [EH+04]. Dabei gingen sie allerdings nicht auf konkrete Software ein, sondern legten den Fokus auf die Umsetzung der virtuellen Übergangsstellen. Für die Umwandlung der Daten an den Schnittstellen der beiden Simulatoren beschrieben sie die Verwendung von sogenannten Up- beziehungsweise Downstream Agents. Diese sollten Konsistenz gewährleisten und für Synchronisation der beiden Modelle sorgen.

Helbing et al. verwendeten in ihrer Kopplung [Hel+02] ebenfalls die Abstraktionsebenen Makro und Mikro. Es wurde versucht, relevante Simulationsattribute wie Fahrzeugdichte, -geschwindigkeit und -beschleunigung der unterschiedlichen Ebenen mithilfe von komplexen Differentialgleichungen umrechnen zu können. Für die Mikro-Ebene wählten sie dabei einen car-following Ansatz zur Simulation der einzelnen Fahrzeuge. Dieser basiert auf der Annahme, dass das Verhalten eines Fahrzeugs (abgesehen von einigen individuellen Parametern) nur von dem des vorausfahrenden abhängt. Im Makro-Bereich nutzten sie ein non-local gas-kinetic-based traffic (GKT)

Modell, welches normalerweise auf Meso-Ebene angewandt wird. Dabei werden Attribute wie die Verkehrsdichte von einer Wahrscheinlichkeitsdichtefunktion bestimmt, die Fahrzeuge in unterschiedlichen Status darstellt [TAH03]. Jedoch wurde auch bei dieser Arbeit die Kopplung nicht anhand konkreter Simulationssoftware getestet.

2014 entwickelten Abouaïssa et al. in [AKM14] einen grundsätzlich anderen Ansatz. Bei diesem war es das Ziel eine Möglichkeit zu entwickeln, um die Sicht auf eine Verkehrssimulation dynamisch von Makro zu Mikro und umgekehrt ändern zu können. Der "multi-level agent"-basierte Simulator "JAM-FREE" orientiert dabei auf der Vorgehensweise des Modells "SIMILAR", welches ebenfalls mit einer dynamisch variablen Abstraktionsklasse arbeitet. Anschließend wurde diese Simulationssoftware mit großen Datenmengen und sehr vielen Testfällen umfangreich erprobt und die Anwendbarkeit bestätigt.

Nach [Ma+11] et al. liegt die wichtigste Herausforderung von Simulatorenkopplungen in der Konsistenz des hybriden Modells. An den Übergangsstellen einer mikro-Simulation zu makro- oder mesoskopischen Bereichen gehen durch die Aggregation Informationen zu einzelnen Fahrzeugobjekten verloren. An Schnittstellen in entgegengesetzter Richtung müssen hingegen einzelne Objekte erzeugt werden, ohne jedoch die Konsistenz der Daten zu verlieren. Dabei werden oft Wahrscheinlichkeitsverteilungen eingesetzt, um unter anderem individuelle Abfahrtszeiten, Routen oder Geschwindigkeiten der neu erzeugten Fahrzeuge einzeln festzulegen. Diese anhand von Zufallsvariablen erstellten Attributwerte können jedoch nach [Tay+18] die Realitätsnähe des Systems beeinträchtigen und die Reproduzierbarkeit von Ergebnissen erschweren. [PJL02] et al. kamen bei einer Auswertung von mehr als 2200 Veröffentlichungen zu dem Ergebnis, dass die Zuverlässigkeit der Auswertungen von Simulationen vor allem durch den Einsatz von Wahrscheinlichkeiten und Zufall stark verringert wird. Daran ist zu erkennen, dass das Thema "Konsistenz" in diesem Forschungsgebiet von essentieller Bedeutung ist.

Es lässt sich zusammenfassen, dass dieser Themenbereich in den letzten Jahren sehr stark an Bedeutung gewonnen hat. Es wurden bereits verschiedene Kopplungen zwischen makro-, meso- und mikroskopischen Simulatoren entwickelt und getestet, wobei sie den Schwerpunkt vor allem auf die dabei verwendeten Paradigmen legten. Es wurden jedoch ebenfalls Kopplungen spezieller Simulatoren sowie Simulationsmodelle mit dynamisch möglicher Variation der Abstraktionsebene veröffentlicht. In den beschriebenen hybriden Ansätzen wurden die Simulatoren nebeneinander laufen gelassen und an Verbindungsstellen wurde die Umwandlung der Daten durchgeführt. In dieser Arbeit soll nun eine Kopplung mit konkreten Simulatoren beschrieben werden, bei der das komplette Straßennetz makroskopisch und bestimmte Teilnetze zusätzlich mikroskopisch betrachtet werden können (siehe auch Abschnitt 3.2.1).

Veränderungen des Verkehrsflusses auf Mikro-Ebene nehmen dennoch wie in den anderen Varianten Einfluss auf die Makro-Ebene.

## 2.2 Beispielhaft verwendete Simulatoren

Im Folgenden sollen die beiden verwendeten Simulationsprogramme VISUM und SUMO kurz beschrieben werden. Außerdem wird begründet, wieso sie für makroskopische beziehungsweise mikroskopische Simulatoren stellvertretend verwendet werden können. Ein tabellarischer Vergleich besonders nennenswerter Unterschiede ist außerdem im Anhang in Tabelle 6.2 zu finden.

### 2.2.1 Makroskopisches Simulationsprogramm: VISUM

Die PTV Group ist seit ihrer Gründung 1979 in den Bereichen Verkehr, Mobilität und Logistik aktiv [Vis]. Sie beschäftigt sich schon seit Jahrzehnten in verschiedenen Projekten und unter Verwendung eigens entwickelter Computerprogramme mit Touren- und Liniennetzplanung von Bus- und Straßenverkehr in Städten. 2009 etablierte sich die PTV Group schließlich durch die Veröffentlichung des Officepakets "PTV Vision" als Marktführer im Bereich der Verkehrsmodellierung und Berechnung von Verkehrsströmen. Laut eigenen Angaben wird ihre Software mittlerweile in mehr als 120 Ländern und über 2500 Städten genutzt. Das Officepaket setzt sich dabei aus mehreren Teilprogrammen zusammen, zu denen unter anderem auch die in dieser Arbeit verwendete Simulationssoftware VISUM gehört. Der Simulator arbeitet mit dem weltweiten Standard für makroskopische Verkehrssimulationen und ist daher als Stellvertreter der entsprechenden Abstraktionsebene gut geeignet. Zudem beinhaltet die Software bereits mit der Studenten-Lizenz einen großen Umfang an Funktionalitäten, die für diese Arbeit größtenteils ausreichend sind. Die maximale Nutzungsdauer ist bei dieser Lizenz allerdings auf 45 Minuten begrenzt. Diese Einschränkung wird in Kapitel 4 bei der Auswahl des Analyseintervalls noch eine Rolle spielen.

In den folgenden Absätzen soll nun ein kurzer Überblick über den Aufbau und die Funktionsweise der Software gegeben werden. Dafür wird zunächst allgemein auf wichtige Funktionen und die Anwendung der Software eingegangen, um darauf folgend Besonderheiten bezüglich der Verwendung von sogenannten origin-destination-Matrizen (OD-Matrizen) und Umlegungen herauszustellen. Anschließend soll die Möglichkeit gezeigt werden, mithilfe der von der Software bereitgestellten Component Object Model (COM)-Schnittstelle Funktionen von einem Python-Skript aus aufzurufen. Aufgrund der Komplexität der Software begrenzt sich die Beschreibung dabei jedoch im Wesentlichen auf die in Kapitel 3 (Architektur) und Kapitel 4 (Evaluation) verwendeten Aspekte und erhebt daher keinen Anspruch

auf Vollständigkeit. Ausführlichere Erläuterungen zur Verwendung von VISUM sind in [Sch] zu finden.

### 2.2.1.1 Benutzeroberfläche

Szenarios werden in der Simulationssoftware standardmäßig als .ver-Dateien (Versions-Dateien) abgespeichert und enthalten neben den Objekten unter anderem alle vorher getroffenen Einstellungen, geöffnete Fenster und eventuelle Änderungen an den Grafikparametern. Eigene Verkehrsnetze können im Bearbeitungsmodus des Programms durch Hinzufügen von Objekten unterschiedlicher Typen selbst erstellt werden. Anschließend ist es möglich verschiedene Attribute entweder einzeln oder gruppenweise anzupassen. Neben der für die meisten Objekttypen vorhandenen eindeutigen Identifikationsnummer (ID) haben diese zusätzlich typspezifische Attribute. So besitzt ein Knoten beispielsweise Koordinaten in x- und y-Richtung, während für eine Strecke die IDs der angrenzenden Knoten, die Länge und Kapazität gespeichert wird. Darüber hinaus ist das Sperren von Strecken für bestimmte Verkehrsmittel wie Personenkraftwagen (PKW) oder Lastkraftwagen (LKW) möglich. Die vom Nutzer nicht bearbeitbaren Streckenauslastungen werden durch eine Umlegung für alle Zeitintervalle berechnet und einzeln für jedes Intervall gespeichert (siehe auch Abschnitt 2.2.1.2). Eine bidirektionale Strecke zwischen zwei Knoten erhält in VISUM zwar nur eine ID, die unidirektionalen Verbindungen sind jedoch durch die Knotennummern der Start- bzw. Zielknoten oder alternativ über das abgeleitete Attribut "IsForward" unterscheidbar. Ein weiterer wichtiger Objekttyp sind Abbieger, die nicht über eigene IDs, sondern über die drei beteiligten Knoten (Start-, Über- und Zielknoten) identifiziert werden können. Die Auslastung von Abbiegern wird äquivalent wie bei Streckenobjekten berechnet und auch eine Sperrung ist möglich (siehe Abbildung 2.1).

The figure consists of three side-by-side windows from the VISUM software:

- Schnellansicht (Knoten):** Shows a table with one row for a node. The columns are: Anzahl: 1, Nr (Value: 13), Name, KapIV (Value: 1000), BellIV (Value: 382), XCoord (Value: 2000.0000), YCoord (Value: -2000.0000).
- Schnellansicht (Strecken):** Shows a table with one row for a street segment. The columns are: Anzahl: 1, Nr (Value: 66), VonKnotNr (Value: 13), NachKnotNr (Value: 8), VSysSet (Value: B,L,P), B,L,P, Länge (Value: 1.000km), KapIV (Value: 100), v0IV (Value: 50km/h), BelFzglV(AP) (Value: 191), IstHin (checkbox checked).
- Schnellansicht (Abbieger):** Shows a table with one row for a roundabout. The columns are: Anzahl: 1, Nr (Value: 8), VonKnotNr (Value: 13), ÜberKnotNr (Value: 14), NachKnotNr (Value: 14), VSysSet (Value: B,L,P), B,L,P, KapIV (Value: 100), BelFzglV(AP) (Value: 95), BelFzglV(AktInt) (Value: 4).

Abbildung 2.1 – Beispiele für Schnellansichten zu den Objekttypen "Knoten", "Strecke" und "Abbieger"

Schließlich existieren noch die über Anbindungen mit dem restlichen Straßennetz verbundenen Bezirke. Sie werden als Start- oder Zielpunkte von Verkehrsflüssen definiert und ermöglichen die Verwendung von Umlegungen. Der Objekttyp "Anbindung" symbolisiert Einspeisungsstrecken zwischen Bezirken und dem Verkehrsnetz. Sie lassen sich als zusätzliche Strecken verstehen, die jedoch nur von eingehenden beziehungsweise ausgehenden Fahrzeugen und nicht von Transitverkehr genutzt werden können. Es ist also im Beispielnetz in Abbildung 2.2 nicht möglich die beiden Anbindungen von Bezirk B zu nutzen, um Verkehr von Bezirk A nach C zu leiten.



Abbildung 2.2 – Verkehrsflüsse über andere Bezirke nicht möglich

### 2.2.1.2 OD-Matrizen und Umlegungen

Mithilfe von sogenannten OD-Matrizen können in VISUM Verkehrsflüsse zwischen je zwei Bezirken geplant werden, wobei die Zeilen dem Startbezirk und die Spalten dem jeweiligen Zielbezirk entsprechen. Die quadratische Matrix gibt also die Anzahl der auf einer Verbindung zweier Bezirke im gesamten Analyseintervall geplanten Fahrzeugen an. Dabei symbolisiert die Summe einer Zeile die Anzahl ausgehender und die einer Spalte die der eingehenden Fahrzeuge. Die Diagonale einer Matrix enthält normalerweise nur Nullen, da in der Regel keine Fahrzeuge denselben Start- und Zielbezirk haben.

| Matrixeditor (Matrix '1 Matrix 1') |          |        |          |      |          |  |
|------------------------------------|----------|--------|----------|------|----------|--|
| 3 x 3                              |          |        | 1        | 2    | 3        |  |
|                                    | Name     |        | Bezirk 1 | SUMO | Bezirk 2 |  |
|                                    |          | Summe  | 100.00   | 0.00 | 100.00   |  |
| 1                                  | Bezirk 1 | 100.00 | 0.00     | 0.00 | 100.00   |  |
| 2                                  | SUMO     | 0.00   | 0.00     | 0.00 | 0.00     |  |
| 3                                  | Bezirk 2 | 100.00 | 100.00   | 0.00 | 0.00     |  |

Abbildung 2.3 – Beispiel für eine OD-Matrix

Das in Abbildung 2.3 gezeigte Beispiel bezieht sich dabei auf ein Szenario, welches ähnlich wie das im vorherigen Abschnitt aufgebaut ist:



Abbildung 2.4 – Beispielnetz zur Erklärung von OD-Matrizen

Durch die Verwendung von Nachfrageganglinien ist es außerdem möglich, Zeitschritte im Analyseintervall unterschiedlich zu gewichten, um beispielsweise Stoßzeiten zu simulieren. In Abbildung 2.5 ist eine solche dargestellt, die ein Analyseintervall von 8 Stunden mit stündlich definierter Gewichtung zeigt. Die Daten orientieren sich dabei an [Ros] und soll durch die Gewichte eine Stoßzeit zwischen 6 und 8 Uhr modellieren.

| Number:   | 4         |            |         |          |         |         |                   |
|-----------|-----------|------------|---------|----------|---------|---------|-------------------|
| Name:     | Ganglinie |            |         |          |         |         |                   |
| <hr/>     |           |            |         |          |         |         |                   |
| Number: 8 | Start day | Start time | End day | End time | Weight  | Share   | Accumulated share |
| 1         | 1         | 04:00:00   | 1       | 05:00:00 | 50.000  | 3.333%  | 3.333%            |
| 2         | 1         | 05:00:00   | 1       | 06:00:00 | 120.000 | 8.000%  | 11.333%           |
| 3         | 1         | 06:00:00   | 1       | 07:00:00 | 280.000 | 18.667% | 30.000%           |
| 4         | 1         | 07:00:00   | 1       | 08:00:00 | 375.000 | 25.000% | 55.000%           |
| 5         | 1         | 08:00:00   | 1       | 09:00:00 | 250.000 | 16.667% | 71.667%           |
| 6         | 1         | 09:00:00   | 1       | 10:00:00 | 140.000 | 9.333%  | 81.000%           |
| 7         | 1         | 10:00:00   | 1       | 11:00:00 | 145.000 | 9.667%  | 90.667%           |
| 8         | 1         | 11:00:00   | 1       | 12:00:00 | 140.000 | 9.333%  | 100.000%          |

Abbildung 2.5 – Ganglinie orientiert an [Ros], zwischen 4 und 12 Uhr

Unter Verwendung der OD-Matrix aus Abbildung 2.3, dem Beispielnetz aus Abbildung 2.4 und der Nachfrageganglinie aus Abbildung 2.5 wird bei Berechnung der Umlegung der folgende stündliche Verlauf der Auslastungen einer Strecke erreicht:

| Number: 2 | No | FromNodeNo | ToNodeNo | VolVehPrT(0400) | VolVehPrT(0500) | VolVehPrT(0600) | VolVehPrT(0700) | VolVehPrT(0800) | VolVehPrT(0900) | VolVehPrT(1000) | VolVehPrT(1100) |
|-----------|----|------------|----------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1         | 1  | 1          | 2        | 50              | 120             | 280             | 375             | 250             | 140             | 145             | 140             |
| 2         | 1  | 2          | 1        | 100             | 240             | 560             | 750             | 500             | 280             | 290             | 280             |

Abbildung 2.6 – Stündliche Auslastungen auf einer Strecke zwischen beiden Bezirken

Die Möglichkeiten der Beeinflussung des Verkehrsflusses mit Nachfrageganglinien sind allerdings auf die Anpassung der Fahrzeugsummen begrenzt. So ist es zum Beispiel nicht möglich damit die Richtung der Verkehrsströme zu ändern, da die festgelegten Gewichtungen für die gesamte simulierte Zeit gelten. Zur Modellierung von Verlagerungen des Verkehrsaufkommens in Hin- beziehungsweise Rückrichtungen muss das Analyseintervall in mehrere kleinere Zeitintervalle unterteilt werden, für die jeweils eine eigene OD-Matrix definiert wird. Die Möglichkeit der Richtungsänderung kann beispielsweise verwendet werden, um morgens Berufsverkehr in

Richtung der Arbeitsstätten und abends Feierabendverkehr in Richtung der Wohnbezirke darstellen zu können.

Bei der Berechnung der Umlegung werden für das ausgewählte Verkehrsmittel die OD-Matrizen und die Nachfrageganglinie kombiniert. Auf dieser Basis werden dann die Strecken- sowie Abbiegerauslastungen in allen Zeitintervallen von VISUM mit einem ausgewählten Algorithmus berechnet. Anschließend können die einzelnen diskreten Zeitschritte variabel in der Benutzeroberfläche ausgewählt und die Auslastungen betrachtet werden. Durch Anpassung der Grafikparameter ist es außerdem möglich, die aktuellen Auslastungen in Form von Balken oder Beschriftung der Strecken zu visualisieren.

### 2.2.1.3 COM-Schnittstelle

Von VISUM wird eine Schnittstelle bereitgestellt, mit der Attributwerte über ein Python-Skript aus der Simulation ausgelesen sowie geändert werden können. Außerdem ist es damit möglich auf Funktionen wie beispielsweise die Umlegung zugreifen. Eine übersichtliche Auflistung aller enthaltenen Objekte, Methoden und Attribute findet sich in der Software unter "Help/COM Help". In diesem Abschnitt soll ein grundlegendes Verständnis der wichtigsten Funktionen dieser sogenannten COM-Schnittstelle vermittelt werden. Die Ausführungen begrenzen sich auch hier wieder auf die verwendeten Funktionalitäten. Eine tabellarische Auflistung dieser inklusive konkreter Beispiele zur Verwendung ist zusätzlich im Anhang in Tabelle 6.3 zu finden.

#### Visum starten, Versions-Datei öffnen und speichern

Um die Funktionen von VISUM im Python-Skript verwenden zu können muss zunächst eine Instanz des Programms gestartet werden (Abbildung 2.7, Zeile 1). Diese wird anschließend verwendet, um die Schnittstelle zu nutzen. Als erstes sollte eine Version-Datei geöffnet werden, in der anschließend gearbeitet werden kann (Zeile 2). Dabei sollte der visumPath dem absoluten Pfad der Datei entsprechen, um Problemen bei der Adressierung vorzubeugen. Anschließend kann das Szenario in VISUM verwendet und bearbeitet werden. Nach der Verwendung kann die Datei dann wieder an einem (gegebenenfalls neuen) Pfad abgespeichert werden (Zeile 4).

---

```
1     visum = win32com.client.Dispatch("Visum.Visum")
2     visum.LoadVersion(visumPath)
3     // Verwendung von Visum...
4     visum.SaveVersion(visumPath)
```

---

Abbildung 2.7 – Starten, Laden und Speichern von Versions-Dateien

### Auswahl von Objekten

Um über das Skript mit einzeln oder gruppenweise mit VISUM-Objekten zu interagieren, müssen diese zunächst abgefragt werden. Die einfachste Variante besteht darin, mit

```
visum.Net.[objectType].ItemByKey([identificationAttributes])
```

ein Objekt über seine eindeutigen Attribute abzufragen. Die verschiedenen Typen unterscheiden sich dabei lediglich in ihren Identifikationsattributen. Während Knoten über die eigene ID adressiert werden, verwenden Strecken und Abbieger zur Identifikation die Nummern der zwei beziehungsweise drei angrenzenden Knotenobjekte. OD-Matrizen können wiederum über die eindeutige Matrixnummer adressiert werden.

Es ist ebenfalls möglich, Objekte gruppenweise als Collection abzufragen. Dafür gibt es die Funktion `FilteredBy()`, die mit dem Befehl

```
visum.Net.[objectType].FilteredBy([filterCondition])
```

aufgerufen wird. Die `filterCondition` ist dabei wie Bedingungen in Java oder C++ aufgebaut, das bedeutet einzelne Bedingungen können durch Verkettung mit "`| |`" (in Python: or) beziehungsweise "`&&`" (and) aneinander gereiht werden. Die erhaltenen Elemente können anschließend über einen Iterator abgelaufen werden. Eine Adressierung der in der Collection gespeicherten Objekte über einen Index ist mit der Schnittstelle nicht möglich.

### Auslesen und Änderung von Attributwerten

Zum Auslesen von Attributwerten existiert für fast jeden Objekttyp die Funktion

```
[object].AttValue([attribut]).
```

Dabei bezeichnet `object` ein Objekt, welches entweder über `ItemByKey()` oder eine Collection kombiniert mit einem Iterator ausgelesen wurde. Neu setzen lassen sich Attributwerte über Collections mit der Funktion

```
[collection].SetAllAttValues("[attribut]", "[newValue]"),
```

wobei die Anzahl der gespeicherten Elemente irrelevant ist. Es kann also auch über `FilteredBy()` eine Collection mit nur einem Element gebildet werden, um anschließend Änderungen an den Attributen dieses einzelnen Objekts durchzuführen.

Die Verwendung von Matrizen unterscheidet sich beim Auslesen und Ändern

etwas von den anderen Objekttypen. So können die Werte einzelner Zellen wie folgt über `row` und `col` abgefragt oder geändert werden:

```
[matrix].GetValue([row], [col])  
[matrix].SetValue([row], [col], [newValue])
```

### 2.2.2 Mikroskopisches Simulationsmodell: SUMO

Die mikroskopische Verkehrssimulation SUMO wird hauptsächlich vom Deutschen Zentrum für Luft- und Raumfahrt (DLR) entwickelt und ist seit 2001 kostenlos und Open-Source verfügbar [Sum]. Sie wurde für die Modellierung verschiedenartiger Verkehrssysteme entwickelt und bereits in mehreren nationalen und internationalen Projekten erfolgreich eingesetzt. Unter anderem im Bereich Car2Car-Kommunikation ist das Softwarepaket durch den frei zugänglichen und damit individuell anpassbaren Quellcode sehr verbreitet. Neben dem graphical user interface (GUI) sind in dem Paket ebenfalls Programme zur Bearbeitung und Konvertierung von Netzen enthalten. So ist es beispielsweise möglich, Verkehrsnetze von VISUM- oder OSM-Dateien in das von SUMO verwendete Format umzuwandeln. Außerdem existieren ausführlich dokumentierte Schnittstellen zu den Programmiersprachen C++ und Python. Aufgrund der mittlerweile weiten Verbreitung der Software und guten Anwendbarkeit auch in größeren Projekten bietet es sich deshalb an, diesen Simulator als beispielhaften Vertreter mikroskopischer Mobilitätssimulationen zu wählen.

Im Folgenden sollen die verwendeten Funktionen des Softwarepaketes kurz erläutert werden. Zunächst werden dafür die Verwendung der einzelnen Komponenten angesprochen und weitere Informationen zur Benutzeroberfläche und GUI bereitgestellt. Anschließend wird auf die wichtigsten Funktionen der Traci-Schnittstelle eingegangen, die für die Verwendung der SUMO-Funktionen über ein Python-Skript nötig sind. Auch diese Erklärungen erheben keinen Anspruch auf Vollständigkeit und sollen lediglich ein grundlegendes Verständnis vermitteln.

#### 2.2.2.1 Benutzeroberfläche und Dateitypen

Im Gegensatz zu VISUM sind bei SUMO nicht alle Informationen in einer Datei gespeichert, sondern auf mehrere aufgeteilt. Bei diesem Simulator wird zunächst das Straßennetz als .net.xml-File erstellt. Dafür gibt es im Wesentlichen die beiden folgenden Möglichkeiten:

- **netedit:** Mit dem Netzeditor von SUMO können eigene Netze durch Hinzufügen und Bearbeiten von Knoten, Strecken und anderen Elementen erzeugt werden. Außerdem besteht die Möglichkeit, bereits bestehende Netze zu öffnen und zu verändern.

- **netconvert:** Durch `netconvert` können Straßennetze aus anderen Formaten in das von SUMO verwendete konvertiert werden. Für diese Arbeit relevant ist vor allem die Umwandlung von VISUM-Netzen. Dafür müssen diese zunächst im makroskopischen Simulationsprogramm als Netz exportiert werden, bevor anschließend die Konvertierung mit `netconvert` zum Format von SUMO möglich ist. In Kapitel 6 wird die Umwandlung noch einmal schrittweise erklärt. Neben VISUM-Netzen können noch weitere Formate konvertiert werden. Dazu gehören unter anderem .osm-Dateien, die aus Open-Street-Map (OSM) exportiert werden können und neben dem Straßennetz auch eine große Menge an zusätzlichen Informationen enthalten.

Nach der Erstellung oder Umwandlung einer Netzdatei aus VISUM befinden sich in dieser nur die Positionen, Relationen und Attribute von Kreuzungen und Strecken. Zusätzliche Objekte wie Detektoren (InductionLoops) oder Bushaltestellen können in einer zusätzlichen Datei mit der Endung `.add.xml` definiert werden. Für die Speicherung ihrer Positionen wird die Strecken- und Spurnummer sowie eine relative Position des Objekts auf der Strecke (zwischen -1 und 1) gespeichert.

In einer `.sumocfg`-Datei (SUMO-Konfigurationsdatei) werden das Netz und die Zusatzobjekte schließlich referenziert und so miteinander verknüpft. Konfigurationsdateien können anschließend in der GUI von SUMO geladen und die Simulation daraufhin ausgeführt werden.

### 2.2.2.2 Traci-Schnittstelle

Die Traci-Schnittstelle von SUMO stellt Funktionen zur Steuerung der mikroskopischen Simulation bereit. Dafür wird zunächst eine Konfigurationsdatei geöffnet und anschließend eine Verbindung zur Schnittstelle hergestellt. Über diese können dann Befehle direkt an die mikroskopische Simulation weitergegeben werden. Eine ausführliche Dokumentation ist in [Tra] zu finden. Die wichtigsten verwendeten Funktionen sind allerdings die Folgenden:

- **step(time=0.0)**  
Weitersimulation bis zur übergebenen Simulationszeit (optional) oder alternativ um einen Schritt ( $\triangleq 1s$ )
- **getTime()**  
Abfrage der aktuellen Simulationszeit
- **addRoute(routeID, route)**  
Erstellung einer Route mit eindeutiger ID aus einer Liste von abzufahrenden Strecken
- **getRoute(vehicleID)**  
Ausgabe der aktuellen Route des Fahrzeugs mit der übergebenen ID

- **addVehicle(vehicleID, routeID)**  
Erstellung eines Fahrzeugs mit eindeutiger ID, welches nach der Erzeugung die übergebene Route abfährt
- **changeTarget(vehicleID, edgeID)**  
Änderung des aktuellen Fahrtziels eines Fahrzeugs und interne Generation einer neuen Route
- **getLastStepVehicleIDs(detectorID)**  
Abfrage der durch einen bestimmten Detektor im letzten Zeitschritt registrierten Fahrzeuge

## 2.3 Exponentialverteilung

In diesem Abschnitt sollen grundlegende Informationen zur Exponentialverteilung gegeben werden. Die Verteilung wird in der entwickelten Kopplung an Übergangsstellen von VISUM nach SUMO verwendet, um Abfahrtszeiten von Fahrzeugen zu erstellen.

Eine Zufallsvariable  $x$  heißt exponentialverteilt mit dem Parameter  $\lambda$ , wenn sie die Dichtefunktion

$$f(x) = \begin{cases} 0 & \text{falls } x < 0 \\ \lambda e^{-\lambda x} & \text{falls } x \geq 0 \end{cases} \quad (2.1)$$

besitzt [Fah+07]. Eine Wahrscheinlichkeitsdichtefunktion bildet dabei die Ereignisse auf ihre jeweiligen Wahrscheinlichkeiten ab. Die Fläche unter dem Graphen einer solchen Funktion auf einem Intervall  $I = [a, b]$  entspricht der Wahrscheinlichkeit, einen Wert in diesem Intervall, also zwischen  $a$  und  $b$ , zu erhalten. Das Integral einer Wahrscheinlichkeitsdichtefunktion wird als (Wahrscheinlichkeits-) Verteilung bezeichnet und ist im Fall der Exponentialfunktion einer Zufallsvariablen wie folgt definiert:

$$F(x) = \begin{cases} 0 & \text{falls } x < 0 \\ 1 - e^{-\lambda x} & \text{falls } x \geq 0 \end{cases} \quad (2.2)$$

$F$  wird daher auch Exponentialverteilung genannt. Der Parameter  $\lambda$  gibt dabei an, wie schnell die Exponentialfunktion für  $x \rightarrow \infty$  gegen 0 geht und bestimmt somit auch den Erwartungswert der exponentialverteilten Zufallsvariable:

$$E(x) = \lambda \int_0^{\infty} f(x) dx = \frac{1}{\lambda} \quad (2.3)$$

Für die Erzeugung exponentialverteilter unabhängiger Zufallsvariablen ist die Transformation von Zufallszahlen aus einer  $[0, 1]$ -Uniformverteilung nötig. Dafür wird die Inversionsmethode verwendet, die nun kurz erklärt werden soll.

Bei dieser Methode handelt es sich um ein Simulationsverfahren, welches basierend auf dem sogenannten Simulationslemma durch das Bilden und Anwenden einer inversen Funktion aus uniform verteilten Zufallszahlen andere Wahrscheinlichkeitsverteilungen erzeugt:

Sei  $X$  eine [...] reellwertige Zufallsvariable mit Verteilungsfunktion  $F$ .

Wir definieren die Pseudoinverse von  $F$  für  $x \in [0, 1]$  durch

$$F^{-1}(x) := \inf \{ q : F(q) \geq x \}. \quad (2.4)$$

[Sei außerdem  $U$  eine uniforme verteilte Zufallsvariable.]

Dann ist  $F^{-1}(U)$  genauso verteilt wie  $X$ .

[Pfa13]

Die Pseudoinverse der Exponentialverteilung für das Intervall  $I = [0, \infty)$  ist

$$F^{-1}(x) = -\frac{1}{\lambda} \cdot \ln(1-x). \quad (2.5)$$

Da  $x$  laut Annahme eine gleich verteilte Zufallszahl ist, gilt selbiges für  $1-x$ . Daher kann  $1-x$  durch  $x$  ersetzt werden. Außerdem kann die Flussrate  $\lambda$ , die die Anzahl Fahrzeuge pro Zeiteinheit darstellt, durch die Abfahrtsrate  $\mu$ , also die durchschnittliche Zeit bis das jeweils nächste Fahrzeug abfährt, beschrieben werden:

$$\lambda = \frac{1}{\mu} \text{ bzw. } \mu = \frac{1}{\lambda} \quad (2.6)$$

Es ergibt sich also schließlich für die Erzeugung von exponentialverteilten Abfahrtsintervallen  $t$  abhängig von der Abfahrtsrate  $\mu$  und der Zufallsvariablen  $x$  die folgende Formel:

$$t(x) = -\mu \cdot \ln(x). \quad (2.7)$$

Ye und Zhang beschreiben in [YZ09], dass nach aktuellen Forschungen die Verteilungen von Fahrzeugankünften beziehungsweise -abfahrten zwischen einer exponentiellen und einer uniformen Verteilung liegt. Außerdem untersuchten sie die Anwendbarkeit der Exponentialverteilung an einem Beispiel bei unterschiedlicher Wahl der Parameter und bestätigten dabei die Annahme, dass diese Verteilung zur Annäherung verwendet werden kann.

Daher soll Gleichung (2.7) bei dieser Kopplung verwendet werden, um an der Über-

gangsstelle von VISUM nach SUMO anhand der makroskopischen Auslastungen der Übergangsstrecken Fahrzeugerzeugungszeiten für die mikroskopische Simulation zu erstellen. In Kapitel 4 wird untersucht, ob diese Wahl bei der entwickelten Simulation sinnvoll war und die Erzeugung mit dieser Methode annähernd den Daten einer verglichenen ausschließlich makroskopischen Simulation entspricht.

---

## Kapitel 3

---

# Entwickelte Architektur

---

Dieses Kapitel befasst sich mit der entwickelten Simulatorenkopplung. Zunächst werden die Spezifikationen der verwendeten Soft- und Hardware kurz zusammengefasst. Darauf folgt eine Erläuterung des theoretischen Ansatzes und der Herangehensweise, wobei auch auf gewählte Vereinfachungen eingegangen werden soll. Anschließend wird zuerst ein kurzer Überblick über die Kopplung gegeben, bevor danach detaillierter auf die einzelnen Teilbereiche der Implementierung eingegangen wird.

### 3.1 Verwendete Soft- und Hardware

Für das System wurden die in Abschnitt 2.2 bereits beschriebenen Simulatoren VISUM und SUMO verwendet. Als Programmiersprache wurde Python in der Version 3.9 gewählt, da es von beiden Simulationsprogrammen unterstützt wird und die entsprechenden Schnittstellen gut dokumentiert sind. Für die Entwicklung der Skripte wird die IDE "Pycharm" der Firma Jetbrains mit einer kostenfreien Studentenlizenz genutzt. Als Betriebssystem wurde Windows gewählt, da VISUM nur für diese Plattform verfügbar ist. Tabelle 3.1 fasst die genauen Spezifikationen der Soft- und Hardware inklusive Versionsnummern und Lizenzen noch einmal zusammen.

| Hardwarekomponente/Software | Spezifikation   |
|-----------------------------|---|
| Prozessor                   | Intel Core i5-3550 (3,30 GHz)                           |
| Festplatten                 | Intenso SSD 128 GB, WDC HDD 2 TB                        |
| RAM                         | 16 GB DDR3  |
| Betriebssystem              | Windows 10 Pro Edition                                  |
| Python                      | Version 3.9   |
| IDE                         | Pycharm von Jetbrains, Version 2020.3.5, Lizenz Student |
| Makroskopischer Simulator   | VISUM der PTV Group, Version 2021, Lizenz Student       |
| Mikroskopischer Simulator   | SUMO, Version 1.8.0, open-source                        |

Tabelle 3.1 – Spezifikationen des verwendeten Systems

## 3.2 Theoretischer Ansatz

Im Folgenden soll auf den Aufbau der Kopplung eingegangen werden. Außerdem wird die Frage diskutiert, welche Teilnetze sinnvollerweise von SUMO und welche von VISUM simuliert werden sollten. Darauf folgt eine Beschreibung der Algorithmen zur Pfadplanung mikroskopischer Fahrzeugobjekte und der Anpassung der makroskopischen Umlegung. Abschließend werden die für das entwickelte System getroffenen Vereinfachungen kurz erläutert.

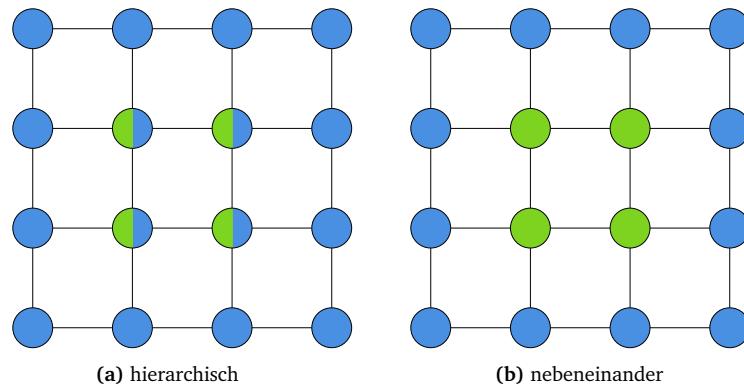
### 3.2.1 Aufbau der Simulatorenkopplung

Bei der Verwendung mehrerer Simulationsprogramme unterschiedlicher Abstraktionsebenen für die Simulation eines einzelnen Netzes müssen Überlegungen zur Anordnung und der Verwendung jener durchgeführt werden. In verwandten wissenschaftlichen Veröffentlichungen wurden bereits verschiedene Kombinationen und Varianten des Aufbaus verwendet und untersucht (siehe auch Abschnitt 2.1). In dieser Arbeit soll eine räumliche Aufteilung der Simulatoren genutzt werden, jeder Simulator modelliert also ein oder mehrere (Teil-) Gebiete vollständig. Es sollen dafür zunächst zwei mögliche Anordnungsvarianten betrachtet und diskutiert werden, bevor anschließend eine von ihnen für die Architektur ausgewählt wird. Die erste Möglichkeit besteht in einem hierarchischen Aufbau. Bei diesem wird das gesamte Streckennetz makroskopisch und bestimmte Teilnetze zusätzlich mikroskopisch modelliert. Das hat den Vorteil, dass VISUM die Auslastungen von Strecken und Abbiegern im gesamten Simulationsgebiet mithilfe der OD-Matrizen berechnen kann und diese anschließend für die Routenplanung über Abbiegewahrscheinlichkeiten genutzt werden können (siehe auch Abschnitt 3.2.2). Es werden zwar nun Teilnetze von beiden Simulatoren modelliert, jedoch ist davon auszugehen, dass die erzielten Vorteile bezüglich der Vereinfachung des gesamten Modells dem zusätzlichen Re-

rechenaufwand im makroskopischen Modell gegenüber überwiegen.

Ein weiterer Ansatz wäre, die Simulatoren nebeneinander ablaufen zu lassen. Das würde bedeuten, dass Teilbereiche mikroskopisch und das restliche Netz makroskopisch simuliert werden würde. Jeder Knoten und jede Strecke lässt sich dann genau einem Simulator zuordnen, da keine Überschneidungen auftreten. Bei dieser Variante könnte die Pfadplanung beziehungsweise Zielfindung von Fahrzeugobjekten jedoch nicht mit dem in Abschnitt 3.2.2 beschriebenen Algorithmus mithilfe von Abbiegewahrscheinlichkeiten durchgeführt werden. Stattdessen müssten komplexe Rechnungen aufgestellt werden, um in SUMO die eingehenden Auslastungen den ausgehenden zuzuordnen. Außerdem würde dieser Aufbau die Umlegungen und Verkehrsplanungen in VISUM erschweren. Es müsste eine Möglichkeit gefunden werden, Verkehr durch mikroskopische Gebiete zu planen, ohne dass dieses Gebiet jedoch makroskopisch simuliert wird. Eine Lösung dafür wären zusätzliche künstliche Bezirke an den Übergangsstellen, zu denen Fahrzeuge hin beziehungsweise von ihnen weg geplant werden können. Allerdings wäre die Aufstellung von passenden OD-Matrizen sehr komplex, da die vorherigen Flüsse analysiert und auf die neuen Bezirksbeziehungen aufgeteilt werden müssen. Dabei ist es notwendig bereits vor der Umlegung Vorüberlegungen zu den Routen anzustellen, um die künstlichen Bezirke realistisch einzubeziehen. Und auch die Umlegung selbst würde durch die wesentlich größere Anzahl an Bezirken rechenaufwändiger werden.

In Abbildung 3.1 werden die beiden Anordnungen noch einmal grafisch veranschaulicht. Grün gekennzeichnete Knoten stehen dabei für die mikroskopische Simulation SUMO, blaue für VISUM. Die Zweifarbigkeit der mittleren Knoten bei (a) soll ausdrücken, dass diese in beiden Simulatoren enthalten sind.



**Abbildung 3.1 – Verschiedene Anordnungen der beiden Simulatoren**

Aufgrund der beschriebenen schwerwiegenden Nachteile der zweiten Variante wird in dieser Arbeit ausschließlich der hierarchische Ansatz verwendet.

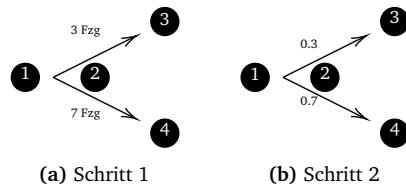
### 3.2.2 Pfadplanung über Abbiegewahrscheinlichkeiten

Bei der Erzeugung von Fahrzeugobjekten in SUMO muss jedem von ihnen initial eine Route zugewiesen werden. Da VISUM als makroskopische Simulation keine einzelnen Objekte modelliert, enthält es keine Informationen zu individuellen Routen, sondern nur die Auslastungen von Strecken und Abbiegern. Für die Erzeugung eines Fahrzeugs muss zunächst also eine gültige Streckenabfolge durch das mikroskopisch simulierte Teilnetz generiert werden. An dieser Stelle soll die in Abschnitt 3.4 dafür verwendete Methode kurz erläutert werden. Aufgrund der besseren Verständlichkeit behandelt der Abschnitt jedoch "Abbiegewahrscheinlichkeiten" statt der tatsächlich genutzten "Gewichten von Abbiegern". Beide Varianten beruhen auf demselben mathematischen Prinzip, weswegen diese Vereinfachung hier ausreichend sein soll. In Algorithmus 3.1 am Ende dieses Abschnitts ist die leicht unterschiedliche Verwendung mittels Gewichtungen dargestellt.

Für beide Umsetzungen sind die Auslastungen der Abbieger in VISUM essentiell. Diese können in Abbiegewahrscheinlichkeiten umgerechnet werden, indem die Auslastung jedes Abbiegers durch die Summe der Gesamtauslastung in dieser Richtung geteilt wird. Zur Bildung der Summe werden dabei alle  $n$  Abbieger einbezogen, bei denen Start- und Über-Knoten übereinstimmen. In Gleichung (3.1) ist die Berechnung der Wahrscheinlichkeiten noch einmal formal dargestellt. `Abbieger_abc` referenziert dabei denjenigen Abbieger, der den Startknoten mit ID `a`, den Überknoten `b` und den Zielknoten `c` gespeichert hat.

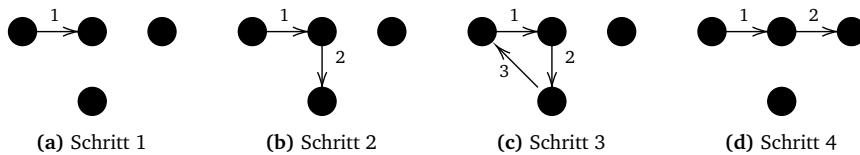
$$P_{\text{Abbieger\_xyz}} = \frac{V_{\text{Abbieger\_xyz}}}{\sum_{i=0}^n V_{\text{Abbieger\_xyi}}} \quad (3.1)$$

In Abbildung 3.2 ist außerdem ein einfaches Beispiel dargestellt, in dem es die beiden Abbieger 1-2-3 und 1-2-4 gibt. Die Summe beider Auslastungen lässt sich mit  $3 + 7 = 10$  Fahrzeugen bestimmen. Damit ergibt sich für die Auswahl des Abbiegers nach Knoten 3 eine Wahrscheinlichkeit von 30%, nach 4 hingegen 70%. Bei der Verwendung von Gewichtungen würden 3 und 7, beziehungsweise die Bereiche  $[0, 3)$  und  $(3, 10]$ , verwendet werden. Anschließend wird eine zufällige Zahl zwischen 0 und der Summe der Auslastungen bestimmt und so ein Abbieger ausgewählt.



**Abbildung 3.2 – Berechnung von Abbiegewahrscheinlichkeiten anhand der Auslastungen**

Anschließend wird an jeder Kreuzung unter Einbezug der Abbiegewahrscheinlichkeiten beziehungsweise -gewichte ein Abbieger zufällig ausgewählt und einer Liste von Knoten auf der Route hinzugefügt. Um Zyklen zu vermeiden müssen bei einem wiederholten Einfügeversuch alle seit dem letzten Auftreten gewählten Knoten-IDs wieder entfernt werden (siehe Beispiel in Abbildung 3.3). Damit wird garantiert, dass die Fahrzeuge keine unnötig langen Routen wählen. Vermutlich wird außerdem eine bessere Annäherung an die Realität erreicht, da reale Fahrzeuge im Allgemeinen ebenfalls Zyklen vermeiden.



**Abbildung 3.3 – Entfernung von Zyklen bei der Pfadplanung**

Dieses Auswahlverfahren wird so lange durchgeführt, bis der Algorithmus einen nicht mehr von SUMO modellierten Knoten hinzufügt. An dieser Stelle wird das Fahrzeug nach dem Abfahren der Route wieder aus der mikroskopischen Simulation entfernt und später gezählt (siehe Abschnitt 3.2.3). Daher ist die Erstellung der Route abgeschlossen und das neue Fahrzeug kann anschließend erzeugt werden. Es gibt in SUMO zwei Wege ein Fahrzeug mit einer Route zu initialisieren. Da sie in Kapitel 4 verglichen werden sollen, folgt nun eine kurze Erklärung der beiden Varianten:

1. Dem erzeugten Fahrzeug wird die Route bei Erzeugung in Form einer Liste von Strecken übergeben, die es anschließend in SUMO abfährt. Die Fahrzeugobjekte der mikroskopischen Ebene verhalten sich also genau entsprechend der Auslastungen aus der makroskopischen, sofern die oben beschriebene Routenplanung anhand von Abbiegewahrscheinlichkeiten konsistente Routen erstellt. Nur im Fall einer zeitweise gesperrten Strecke wird die Route von SUMO umgeplant.

2. Dem erzeugten Fahrzeug wird anfangs nur eine "Dummy-Route" zugewiesen. Diese kann zum Beispiel aus der ersten Strecke des vorher erstellten Streckenverlauf bestehen. Anschließend wird das Ziel des Fahrzeugs mithilfe von `changeTarget()` auf die Zielstrecke der Route, also das letzte Element der Liste, gesetzt. Bei diesem Vorgehen wird nicht der gesamte Streckenverlauf des Objekts von den Abbiegewahrscheinlichkeiten bestimmt, sondern lediglich das Ziel (Tabelle 3.2). Die Routenplanung wird von SUMO anhand der Start- und Zielstrecke selbst berechnet. In der mikroskopischen Simulation können durch die höhere Genauigkeit wesentlich mehr Informationen gespeichert und bei der Routenplanung in Betracht gezogen werden. Dazu gehören beispielsweise geografische Umstände, wie der Grad des Anstiegs/Gefälles der Straße oder enge Kurven, durch die Geschwindigkeit der Fahrzeuge geringer ist. Durch derartige Zusatzinformationen kann dann von SUMO möglicherweise eine bessere Wahl der Route getroffen werden, als es durch die makroskopische Umlegung von VISUM möglich ist.

|            | Erzeugt       | Verwendet            |
|------------|---------------|----------------------|
| Variante 1 | 1, 2, 3, 4, 5 | Route: 1, 2, 3, 4, 5 |
| Variante 2 | 1, 2, 3, 4, 5 | Start: 1, Ziel: 5    |

**Tabelle 3.2 – Unterschiede beider Varianten der Routenplanung**

---

**Require:** Übergangsstrecke  $a$ , Auslastungen aller Abbieger in VISUM

- 1: Erstelle Liste mit Knoten-IDs der beiden Knoten von  $a$ :  $r = [a_1, a_2]$
  - 2: **repeat**
  - 3: Wähle die individuellen Auslastungen als Gewichte der jeweiligen Abbieger
  - 4: Erzeuge eine Zufallszahl  $x$  im Bereich  $[0, \sum \text{Gewichte}]$
  - 5: Wähle anhand von  $x$  und den Gewichten einen der Abbieger aus
  - 6: Sei  $z$  der Zielknoten des gewählten Abbiegers
  - 7: **if**  $z \in r$  **then**
  - 8: Entferne alle Elemente seit dem ersten Auftreten von  $z$  aus  $r$
  - 9: **else**
  - 10: Füge  $z$  zur Liste  $r$  hinzu
  - 11: **end if**
  - 12: **until** Letztes Element von  $r$  ist kein SUMO-Knoten:  $r[-1] \notin \text{sumo\_nodes}$
- 

**Algorithmus 3.1 – Festlegung einer Route anhand von Gewichtungen der Abbieger**

### 3.2.3 Kapazitätsbeschränkungen zur Umlegungsanpassung

Eine weitere Schwierigkeit liegt bei der Anpassung der makroskopischen Simulation basierend auf dem Verhalten der Fahrzeugeobjekte im mikroskopischen Bereich.

Durch die höhere Genauigkeit von SUMO kann es wie bereits beschrieben vorkommen, dass dieser Simulator über mehr Daten zum Verkehrsverhalten verfügt als VISUM. Anhand dieser zusätzlichen Informationen kann der Verkehr im Mikroskopischen eventuell realistischer dargestellt werden und es wäre sinnvoll, die makroskopische Simulation entsprechend daran anzupassen. Zwei Szenarien sind dabei besonders interessant:

- Es entsteht auf einer mikroskopischen Straße ein Stau. Das kann dazu führen, dass weitere Fahrzeuge, die über diese Strecke geplant wurden, möglicherweise durch die geringere Geschwindigkeit nicht so schnell an ihrem Ziel ankommen wie von VISUM berechnet. Bei der Variante 2 der Routenplanung können sie gegebenenfalls um den Stau herum fahren, falls eine Alternativroute existiert. In Variante 1 ist dies nicht möglich, solange die Strecke nicht komplett gesperrt ist.
- Eine Strecke wird, beispielsweise wegen eines Verkehrsunfalls im Mikroskopischen, vorübergehend gesperrt. Unabhängig von der gewählten Variante versucht SUMO eine neue Route für die betroffenen Fahrzeuge zu finden. In dem Fall, dass eine solche nicht existiert, warten betroffene Fahrzeugobjekte am Anfang der Strecke auf die Aufhebung der Sperrung. Selbst wenn eine Alternativroute existiert kann es vorkommen, dass durch den gegebenenfalls weiten Umweg Fahrzeuge nicht rechtzeitig, das heißt in der von VISUM erwarteten Zeit, an ihrem Ziel ankommen.

In beiden Fällen könnte also die Anzahl der Objekte, die von den Detektoren an den Übergangsstellen registriert werden, geringer sein als nach Berechnungen der makroskopischen Umlegung erwartet. Umgekehrt kann es ebenfalls vorkommen, dass mehr Fahrzeuge als erwartet an einer Ausgangsstrecke ankommen, zum Beispiel weil sich ein Stau aufgelöst oder eine Sperrung beendet wurde. Um das Verkehrsverhalten in der gesamten Kopplung möglichst realistisch darstellen zu können sollten die Auslastungen der Strecken in VISUM daher angepasst werden. Eine entscheidende Rolle spielen dabei die Kapazitätsbeschränkungen von Straßen, da es nicht möglich ist die berechneten Auslastungen direkt zu bearbeiten.

An dieser Stelle sollen Kapazitätsbeschränkungen eingesetzt werden. Diese geben an, wie viele Fahrzeuge eine Strecke in einem Analysezeitintervall befahren können. Daher eignen sie sich gut, um eine maximale Auslastung einer Straße in VISUM zu erzwingen und die Umlegung entsprechend anzupassen. Es soll dafür das folgende Beispiel betrachtet werden, bei dem grüne Knoten wieder für mikroskopisch und blaue für makroskopische simuliert stehen:

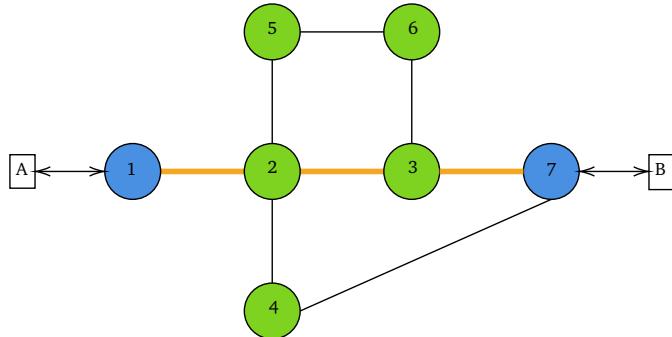


Abbildung 3.4 – Beispielnetz ohne gesperrte Strecke

Das Netz besteht aus einem einfachen Streckenverlauf mit drei unterschiedlich langen Routen zwischen den zwei Bezirken A und B. Bei normalem Verkehr wird von Fahrzeugen auf beiden Abstraktionsebenen die kürzeste, also über die Knoten 1, 2, 3 und 7, gewählt (orange markiert). Im Folgenden soll der Fall betrachtet werden, dass die Strecke von 2 nach 3 vollständig gesperrt wird (nachfolgend mit einem Kreuz markiert). Ein Stau ist im Prinzip nur ein abgeschwächter Sonderfall dieses Szenarios und kann davon abgeleitet werden. Daher soll an dieser Stelle die Betrachtung einer Sperre ausreichen.

Bei beiden in Abschnitt 3.2.2 beschriebenen Varianten wird die Route von SUMO neu geplant und die Fahrzeuge werden anschließend über die obere Alternative, also über die Knoten 2, 5, 6 und 3, fahren (siehe Abbildung 3.5). Die Route ist wesentlich länger als die Alternativroute über Knoten 4, jedoch ist für die erneute Routenplanung in SUMO die Übergangsstrecke 3-7 als Ziel vorgegeben. Daher können die Fahrzeuge mikroskopisch nicht über 4-7 geleitet werden. Da die obere Route länger ist kommen möglicherweise nur noch weniger Fahrzeuge im aktuellen Zeitschritt an ihrem Ziel an und werden bei Strecke 3-7 registriert.

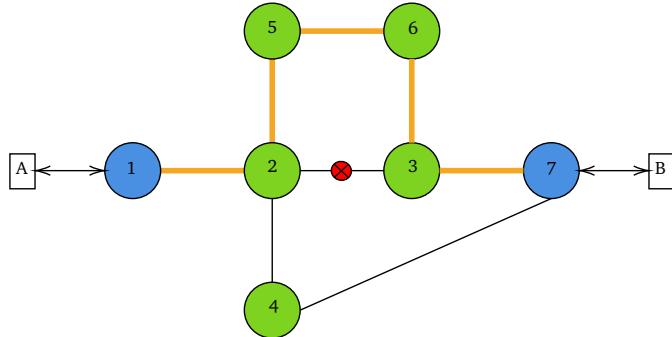


Abbildung 3.5 – Beispielnetz mit gesperrter Strecke, nach neuer Routenplanung von SUMO

An dieser Stelle werden die Kapazitätsbeschränkungen eingesetzt. Um die Umlegung in VISUM an die Detektorwerte anzupassen, wird die Kapazität der entsprechenden Übergangsstrecke von Knoten 2 nach 6 auf die Anzahl gezählter Fahrzeuge begrenzt. Anschließend wird die makroskopische Umlegung neu gestartet. Der Verkehrsfluss wird nun aufgrund der gesperrten Straße vermutlich teilweise über 2, 4 und 7 verlaufen und damit die Auslastung der mittleren Übergangsstrecke teilweise auf die untere verschieben (Abbildung 3.6). Durch die Beschränkung werden also mindestens die Fahrzeuge, die zu wenig in VISUM angekommen sind, über eine andere Route geplant. Da sich bei dieser dann auch die Abbiegewahrscheinlichkeiten und damit die erzeugten Routen ändern, können jetzt in SUMO wahrscheinlich Fahrzeuge über den Knoten 4 geleitet werden. Die Verkehrsflüsse nach der neuen Umlegung sind noch einmal in Abbildung 3.6 dargestellt. Dabei ist zu beachten, dass durch die Existenz der Zielroute vermutlich nicht alle Fahrzeuge über die untere Route geleitet werden. Die Breite einer Linie in der Abbildung soll daher die Größe des Verkehrsflusses symbolisieren.

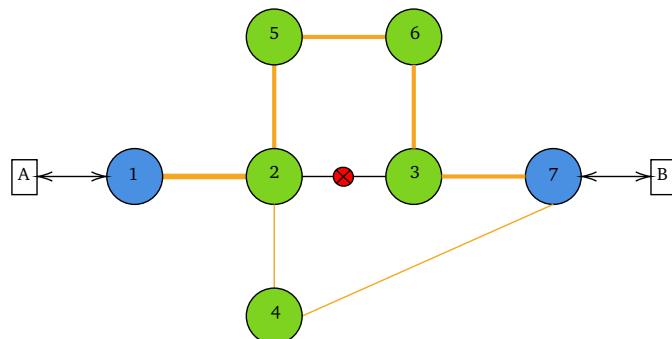


Abbildung 3.6 – Beispielnetz mit gesperrter Strecke, nach erneuter Umlegung

Nun soll noch kurz auf das Verhalten eingegangen werden, falls keine Alternativroute existiert. Dafür wurden aus dem oben gezeigten Beispiel die Knoten 5 und 6 entfernt:

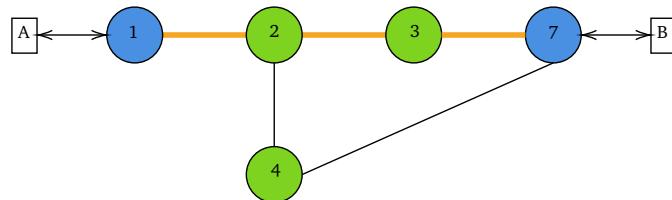


Abbildung 3.7 – Beispielnetz ohne Alternativroute und ohne gesperrter Strecke

Anschließend soll wieder die Strecke 2-3 gesperrt werden (Abbildung 3.8). In diesem Fall kann von SUMO keine Alternativroute zur Zielstrecke 3-7 gefunden werden, weswegen die Fahrzeuge bei Knoten 2 warten, bis die Sperrung aufgehoben wird.

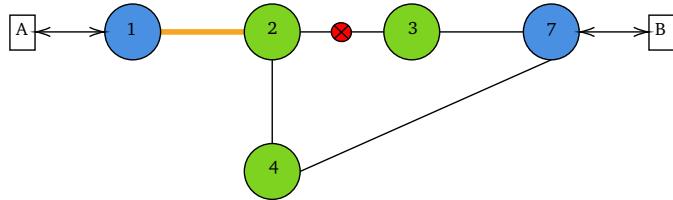


Abbildung 3.8 – Beispielnetz ohne Alternativroute und mit gesperrter Strecke

Es kommen also keine Fahrzeuge bei Bezirk B beziehungsweise an der Übergangsstrecke 3-7 an. Daher wird die Kapazität dieser Strecke nach dem Algorithmus auf 0 gesetzt und die Umlegung neu gestartet. Nach dieser werden nun alle Fahrzeuge über Knoten 4 geleitet, woraufhin sich die Auslastungen der Abbieger und damit die entsprechenden Wahrscheinlichkeiten beziehungsweise Gewichtungen ändern (siehe Abbildung 3.9). Bei einer erneuten Durchführung würden die Fahrzeuge also über die untere Route fahren.

In Kapitel 4 wird dieser Algorithmus noch anhand dieses beispielhaften Szenarios analysiert und bewertet.

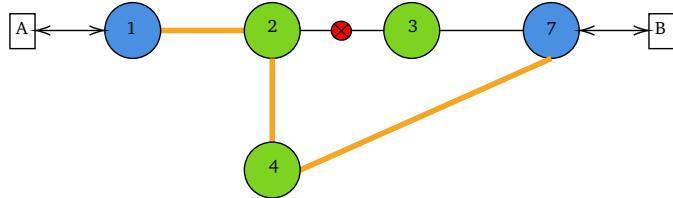


Abbildung 3.9 – Beispielnetz ohne Alternativroute mit gesperrter Strecke, nach erneuter Umlegung

### 3.2.4 Festgelegte Vereinfachungen

Um die Aufgabe der Simulatorenkopplung im Rahmen dieser Bachelorarbeit lösen zu können, wurden mehrere Vereinfachungen und Abstraktionen festgelegt, die in diesem Abschnitt kurz erläutert und begründet werden sollen.

Zunächst einmal wurden zwei spezielle Simulatoren für die Kopplung ausgewählt. Dadurch ist es möglich das entwickelte System an konkreten Beispielszenarien auszutesten und zu optimieren. Wie in Abschnitt 2.2 bereits beschrieben eignen sich VISUM und SUMO durch die große Verbreitung und Verwendung von weltweiten Standards zwar sehr gut als beispielhafte Simulatoren der makroskopischen beziehungsweise mikroskopischen Abstraktionsebene. Dennoch wird durch diese Auswahl die Allgemeingültigkeit des Systems eingeschränkt, sodass die Auswertungen und gewonnenen Erkenntnisse zunächst nur bei diesen beiden Programmen tatsächlich

anwendbar sind. Durch Unterschiede bei den Schnittstellen, Algorithmen oder internen Strukturen kann die fehlerfreie Funktion der entwickelten Implementierung bei Verwendung anderer Simulatoren nicht garantiert werden. Jedoch sollten einige Schlüsse durch den verbreiteten Einsatz von Standards durchaus übertragbar sein.

Außerdem wurde festgelegt, dass Bezirke nur im makroskopischen Gebiet existieren und somit alle Start- und Endpunkte von Verkehrsflüssen ausschließlich von VISUM modelliert werden. Das hat den Vorteil, dass die Fahrzeugsummen der in ein mikroskopisches Gebiet einfahrenden Fahrzeuge in jedem Zeitintervall gleich der der ausfahrenden ist. Dadurch kann die Planung der Verkehrsflüsse vollständig von VISUM übernommen werden und es ist keine simulatorübergreifende Umlegung nötig. Das beschränkt zwar die Realitätsnähe der Simulation, stellt dafür aber eine große Vereinfachung dar. Wenn Fahrzeugrouten in SUMO enden könnten, müsste die bei Erstellung von Fahrzeugobjekten durchgeführte Routenplanung stark abgeändert werden und auch Konsistenzprobleme der beiden Simulationen könnten auftreten. Außerdem wurde die mögliche Anzahl an Start-/Endpunkten begrenzt, da jeder von ihnen einen extra Bezirk und damit eigene Zeilen und Spalten der OD-Matrix benötigt. Aufgrund der Umsetzbarkeit der Auswertung werden daher in den Beispielen sowie in Kapitel 4 der Einfachheit wegen nur wenige Bezirke erstellt. Diese beiden Vereinfachungen erleichtern zwar die Umsetzung stark, bringen jedoch auch Nachteile mit sich. Der wichtigste besteht darin, dass das Simulationsmodell durch diese starken Abstraktionen an Realitätsnähe verliert. In der Realität können Fahrzeuge an nahezu jedem Punkt starten oder enden und schon bei einem kurzen Betrachtungsintervall kann eine große Menge an unterschiedlichen Routen beobachtet werden. Im entwickelten System ist es jedoch möglich Teilnetze wie beispielsweise Wohn- oder Industriegebiete als einzelne Start-/Zielbezirke zusammenzufassen und zu modellieren, von denen dann Verkehrsflüsse ausgehen können. Die Auswirkungen der Abstraktion können also durch das Zusammenfassen von Startpunkten eines Gebiets etwas ausgeglichen werden. Nichtsdestotrotz vergrößern diese Vereinfachungen den Unterschied zwischen Simulation und Realität und wurden deshalb an dieser Stelle genannt.

Bei der Entwicklung der Kopplung wurde nur der Fahrzeugtyp "PKW" betrachtet. Andere möglichen Typen wie beispielsweise "LKW" oder "Motorradfahrer" wurden außer Acht gelassen, um die Simulation auf das Wesentliche zu beschränken. Da der Schwerpunkt des Systems auf der Konsistenz und Anwendbarkeit mehrerer kombinierter Simulatoren besteht reicht die Verwendung eines einzigen Fahrzeugtyps zunächst aus. Die Differenzierung unterschiedlicher Verkehrsteilnehmer kann zwar Einfluss auf den simulierten Verkehr nehmen, ist jedoch nachrangig und kann dem System gegebenenfalls später noch hinzugefügt werden. In diesem Fall sollten außerdem, sofern vorhanden, Verkehrsdaten der einzelnen Fahrzeugtypen genutzt werden, um die Erweiterung anschließend damit zu evaluieren.

Bei der Umlegung des Verkehrs in VISUM können zur Berechnung der Verkehrsflüsse auf Basis von OD-Matrizen und Nachfrageganglinien verschiedene Verfahren angewandt werden. Für diese Implementierung wurde das dynamic user equilibrium (DUE) ausgewählt. Bei diesem Algorithmus wird eine iterative Methode solange ausgeführt, bis entweder Konvergenz oder ein Zeitlimit erreicht ist [Gaw98]. Dabei soll für jedes Fahrzeug des Verkehrsflusses die optimale Route, also die mit den geringsten Reisekosten, bestimmt werden. Der Zustand, in dem alle individuellen Routen diese Bedingung erfüllen, wird "user equilibrium" genannt. An dieser Methode wird bereits seit einigen Jahrzehnten geforscht und es wurden immer wieder neue Veröffentlichungen bezüglich verbesserter Kostenfunktionen und weiterer Optimierungen publiziert. Aufgrund der Aktualität dieses Verfahrens ist es eine gute Wahl zur Berechnung der Umlegung und wird daher im Folgenden ausschließlich verwendet. Da auf den unterschiedlichen Berechnungsmethoden der makroskopischen Umlegung in dieser Arbeit kein Fokus liegt, soll die Anwendung dieses Algorithmus an dieser Stelle genügen. Da nur die Berechnung der Auslastungen der Strecken von dieser Wahl beeinflusst wird, sollte die Kopplung für jedes andere Verfahren ähnlich funktionieren. Es könnte lediglich nötig sein, die Eingabedaten OD-Matrix und Nachfrageganglinie dafür um weitere Daten zu erweitern.

Außerdem wurden die Strecken der Beispielnetze zur Vereinfachung auf eine Spur in jede Richtung begrenzt. Dadurch werden Überholmanöver und deren Auswirkungen ausgeschlossen und es kann schon bei einer geringen Anzahl an Fahrzeugen auf einer Strecke zu Staus kommen. Für die Bewertung der Simulatorenkopplung sollte eine Spur bereits ausreichen. Als Erweiterung des Systems könnten allerdings weitere hinzugefügt und das veränderte Verkehrsverhalten analysiert werden.

Trotz der hier beschriebenen Abstraktionen kann die Funktionsweise und Anwendbarkeit der entwickelten Kopplung bereits mit diesem System getestet und bewertet werden. Ausführliche Erläuterungen und Analysen dazu sind in Kapitel 4 zu finden.

### 3.3 Übersicht über die entwickelte Kopplung

Das entwickelte System besteht aus mehreren Komponenten, die zur Übersichtlichkeit in einzelne Dateien gegliedert wurden:

- **controller.py:**  
Parametrisierung und Starten der Kopplung. Enthält außerdem die Initialisierung der beiden Simulatoren VISUM und SUMO.
- **visum\_sumo.py:**  
Zuständig für die Simulation der Zeit im Skript sowie alle Funktionalitäten bezüglich der Umwandlung an den Schnittstellen Makro nach Mikro.

- **sumo\_visum.py:**

Funktionen für die Übergangsstellen Mikro nach Makro.

- **sumo.py:**

Notwendig für die Anbindung der Traci-Schnittstelle, leitet alle Parameter über die Schnittstelle an die SUMO-Simulation weiter.

- **utility.py:**

Enthält Hilfsmethoden, die in den anderen Komponenten verwendet werden.

Für den Anwender der Kopplung sind alle wichtigen Funktionalitäten in der `controller`-Datei zu finden. Da alle relevanten Parameter darin gesetzt werden können ist es nicht nötig Änderungen an anderen Komponenten vorzunehmen.

## 3.4 Implementierung

In diesem Abschnitt soll nach der kurzen Übersicht nun näher auf die bereits genannten Komponenten eingegangen werden. Die Erklärungen sollen dabei lediglich die Funktionsweise und Intention hinter den Algorithmen beschreiben und gehen daher nicht näher auf Aufrufe von Funktionen aus Standardbibliotheken oder programiersprachenspezifische Besonderheiten ein. Der vollständige Code ist ebenfalls auf GitHub zu finden.<sup>1</sup> Die wichtigsten Stellen werden dabei in der Implementierung mittels Kommentaren erklärt und es wurde darauf geachtet sprechende Namen für Variablen und Funktionen zu wählen.

### 3.4.1 Controller

Wie bereits beschrieben ist dieses Modul für das Starten der beiden Simulatoren zuständig. Anschließend werden dann die jeweiligen Dateien geladen, die das ausgewählte Szenario enthalten. Als Parameter werden der Methode `simulateTime()` das VISUM-Objekt, der Pfad der benötigten Konfigurationsdatei für die Umlegungen und eine Liste der mikroskopisch modellierten Knoten übergeben. Außerdem können optional noch Simulationsparameter wie Start-/Endzeit oder die Länge der makroskopisch umgelegten Zeitintervalle mitgegeben werden, um die Simulation anzupassen. Außerdem kann für das Erstellen von Statistiken noch eine Liste aller in SUMO eingehenden Strecken übergeben werden. Eine Auflistung aller möglichen Parameter und ihrer gegebenenfalls definierten Standardbelegung ist in Tabelle 3.3 zu sehen.

---

<sup>1</sup><https://github.com/la08rehy/BA>

| Parameter     | Standardwert | Beschreibung  |
|---------------|--------------|---|
| visum         | -            | Instanz der Visum-Simulation                              |
| procedurePath | -            | Pfad zur Umlegungs-config-Datei                           |
| sumo_nodes    | -            | Liste der von SUMO simulierten Knoten                     |
| startTime     | 0 [s]        | Startzeit der Simulation                                  |
| endTime       | 230000 [s]   | Endzeit der Simulation                                    |
| visumStep     | 10000 [s]    | Zeitintervalle der Umlegung von VISUM                     |
| scriptStep    | 1 [s]        | Schrittweite im Skript                                    |
| threshold     | 0            | Parameter zur Anpassung der Kapazitätsbeschränkungen      |
| option        | 2            | Variante, wie die Routenplanung in SUMO durchzuführen ist |
| inLinksSumo   | None         | Nach SUMO eingehende Strecken für Statistik               |

**Tabelle 3.3** – Mögliche Parameter zur Anpassung des Ablaufs vom Skript

### 3.4.2 VISUM nach SUMO

Die Simulation der Zeit läuft im Skript in der Methode `simulateTime()`. Dabei wird im Wesentlichen eine Schleife durchlaufen, die mithilfe einer Variablen die aktuelle Zeit mitzählt. Nach jeder Iteration kann der Wert dieser Variablen abhängig von der Wahl des Parameters `scriptStep` angepasst werden. Sie ist außerdem für die Synchronisation mit der Zeit in den beiden Simulationsprogrammen essentiell.

Stimmt der aktuelle Zeitpunkt mit einem VISUM-Schritt überein, müssen die Daten des nächsten Zeitschritt ausgelesen werden. Anschließend ist es möglich die Fahrzeugerzeugungszeiten des nächsten Zeitintervalls zu berechnen (siehe Algorithmus 3.2). Es werden jedoch zunächst die Zähldaten des letzten Intervalls in den Funktionen von `sumo_visum.py` ausgewertet und daraufhin gegebenenfalls Kapazitätsbeschränkungen von Strecken angepasst (siehe Abschnitt 3.2.3). Die Beschreibung der entsprechenden Methoden folgt in Abschnitt 3.4.3. Wenn eine Einschränkung eingerichtet, geändert oder entfernt wurde muss die Umlegung in VISUM erneut berechnet werden, um die Änderungen wirksam zu machen. Erst danach können die Auslastungen an den Übergangsstrecken im nächsten Zeitintervall aus der makroskopischen Simulation ausgelesen werden. Auf Basis dieser Daten folgt in der Methode `sampleAbsoluteArrivalTimes()` die Erstellung von Abfahrtszeiten. Diese Zeiten entsprechen den Erzeugungszeiten der Fahrzeuge und werden zunächst in chronologischer Reihenfolge zwischengespeichert.

---

**Require:** Aktuelle Zeit des Skripts entspricht Zeitschritt in VISUM

- 1: Auswertung der Zähldaten des letzten Zeitintervalls
  - 2: Gegebenenfalls Anpassung der Kapazitätsbeschränkungen der Strecken
  - 3: **if** Änderung von Beschränkungen **then**
  - 4:    Neue Umlegung berechnen
  - 5: **end if**
  - 6: **for all** Nach SUMO eingehende Strecken **do**
  - 7:    Lies Auslastung aus VISUM aus
  - 8:    Berechne absolute Abfahrtszeiten mittels Exponentialverteilung  
      `sampleAbsoluteArrivalTimes()`
  - 9: **end for**
  - 10: Füge anstehenden Eventzeiten in chronologische Liste ein
- 

**Algorithmus 3.2** – Bestimmung der Erzeugungszeiten von Fahrzeugen im nächsten makroskopischen Zeitintervall

Die Funktion `sampleAbsoluteArrivalTimes()` erstellt aus der von VISUM berechneten Auslastung einer Strecke mithilfe einer Exponentialverteilung eine Liste an absoluten Erzeugungszeiten. Dabei wird wie in Abschnitt 2.3 beschrieben vorgegangen, um aus der durchschnittlichen Abfahrtsrate zufällige Zeitintervalle zwischen zwei Abfahrten festzulegen. In Algorithmus 3.3 ist das Vorgehen noch einmal formal beschrieben.

---

**Require:** Aktuelle Auslastung der betrachteten Strecke  $V_{Strecke}$

- 1: Berechne durchschnittliche Abfahrtsrate:  $\mu = \frac{visumInterval}{V_{Strecke}}$
  - 2: Berechne Ende des Zeitintervalls:  $endTime = curTime + visumInterval$
  - 3: Definiere eine Variable für das Hochzählen der Eventzeiten ( $tempTime$ )
  - 4: Erstelle leere Liste für die Abfahrtszeiten
  - 5: **while**  $tempTime < endTime$  **do**
  - 6:    Bestimme zufälliges Zeitintervall  $\Delta t_{next}$  bis zur Abfahrt des nächsten Fahrzeugs mittels Exponentialverteilung
  - 7:    Berechne die absolute Abfahrtszeit  $t_{next}$  durch Addition von  $\Delta t_{next}$  und der letzten Eventzeit
  - 8:    Füge  $t_{next}$  zur Liste hinzu
  - 9: **end while**
  - 10: **return** Liste der Eventzeiten
- 

**Algorithmus 3.3** – Bestimmung zufallsverteilter Erzeugungszeiten auf einem Link mithilfe einer Exponentialverteilung

Die Erstellung der Eventliste wird nur am Anfang jedes Zeitintervalls in VISUM angewendet. Es gibt jedoch auch Funktionen, die in jedem simulierten Schritt des Skriptes ausgeführt werden müssen. So wird die Eventliste in jedem Zeitschritt geprüft und gegebenenfalls ein oder mehrere Fahrzeuge in der mikroskopischen Simulation auf dem dazugehörigen Link erzeugt. Dabei muss für jedes Fahrzeug

zunächst eine Route geplant werden. Der dafür verwendete Algorithmus wurde in Abschnitt 3.2.2 bereits erläutert. Die beiden Möglichkeiten, Fahrzeuge in SUMO zu erzeugen wurden in demselben Abschnitt behandelt und sollen daher an dieser Stelle nicht erneut beschrieben werden. Die zu verwendende Variante der Routenplanung wird als Parameter *option* bereits beim Start der Kopplung übergeben (siehe auch Tabelle 3.3).

In jedem Zeitschritt müssen außerdem an ihrem jeweiligen Ziel angekommene Fahrzeuge registriert werden. Über Detektoren wird in SUMO eine Liste dieser Fahrzeugobjekte erstellt. Diese muss dann jedoch über die Traci-Schnittstelle direkt ausgelesen und zwischengespeichert werden, da sie andernfalls im nächsten Schritt überschrieben wird. Die IDs und Routen aller angekommenen Fahrzeuge werden daher lokal in eine Liste eingefügt, die in jeder Iteration der Schleife erweitert wird. Diese Liste ist wiederum bei der Zählung von Fahrzeugen und der Kapazitätsbeschränkung relevant (siehe Abschnitt 3.2.3).

### 3.4.3 SUMO nach VISUM

In dieser Komponente sind nur die Umsetzung der Kapazitätsbeschränkungen und der dazugehörige Algorithmus enthalten. Für die Methode `capacityModification()` werden die berechneten Auslastungen aus VISUM und die tatsächlich gezählten aus SUMO benötigt. Erstere können für jede Übergangsstrecke direkt über die Schnittstelle zur makroskopischen Simulation abgefragt werden. Die Zählwerte des mikroskopischen Bereichs hingegen müssen mit der Methode `getVol_Sumo()` über die Liste der angekommenen Fahrzeuge berechnet werden (siehe Algorithmus 3.4).

- 
- Require:** Liste der Routen aller im letzten makroskopischen Zeitintervall angekommenen Fahrzeuge
- 1: Erstelle eine neue Liste `dest_list` mit dem jeweils letzten Element jeder Route
  - 2: Häufigkeit eines Wertes  $s$  in `dest_list` entspricht Anzahl Fahrzeuge auf Übergangsstrecke  $s$
- 

**Algorithmus 3.4** – Bestimmen der Zählwerte der Übergangsstrecken aus SUMO

Es wird dabei also das Vorkommen jeder Zielübergangsstrecke in den zwischengespeicherten Streckenverläufen im letzten Zeitintervall angekommener Fahrzeuge gezählt. Das darauf folgende Verfahren zur Kapazitätsbeschränkung wurde in Abschnitt 3.2.3 bereits ausführlich an Beispielen beschrieben.

Außerdem ist in dieser Komponente noch eine Methode für das Speichern von Daten für spätere Statistiken enthalten. Diese soll an dieser Stelle jedoch nicht weiter beschrieben werden, da sie für die Funktion der Kopplung keine Rolle spielt.

### 3.4.4 SUMO

Dieses Modul enthält vor allem die in Abschnitt 2.2.2.2 beschriebenen Funktionsaufrufe, die über die Traci-Schnittstelle an die mikroskopische Simulation weitergegeben wird. Außerdem wurden noch Methoden zum Starten und Laden einer mikroskopischen Konfigurationsdatei sowie zum Beenden von SUMO definiert.

### 3.4.5 Utility

Hilfsfunktionen, die aus mehreren Teilen des Skripts aufgerufen werden müssen, sind in dieser Datei zusammengefasst. Die wichtigsten sind dabei das Auslesen von Auslastungen aus VISUM (siehe auch Abschnitt 2.2.1.3), der Algorithmus zur Auswahl eines Abbiegers mittels von der Auslastung abhängigen Gewichte (siehe Abschnitt 3.2.2) und die Anwendung der Exponentialverteilung (siehe Abschnitt 2.3). Die Funktionsweisen dieser Methoden wurden bereits an anderer Stelle erläutert und werden deswegen hier nicht noch einmal näher erklärt. Weitere Hilfsmethoden sind für das Starten beziehungsweise Speichern einer Versions-Datei in VISUM, zur Umrechnung der unterschiedlichen Zeitformate oder für weitere einfache Umwandlungen von Datentypen in Python zuständig.

---

## Kapitel 4

---

# Evaluation

---

Dieses Kapitel beschreibt Auswertungen der Verwendung des entwickelten Systems und soll so die Kopplung evaluieren. Dafür wurden mehrere künstliche Beispieldaten entwickelt, anhand derer unterschiedliche Tests bezüglich der Funktionsweise der Komponenten durchgeführt werden sollen. Die Verwendung der Kopplung wird dabei mit einer Variante verglichen, die ausschließlich makroskopisch simuliert, um mögliche Fehler und Probleme feststellen zu können. Der Fokus liegt vor allem auf den Unterschieden dieser einzelnen Simulation und der Kopplung und nicht auf Parametern wie Ausführungszeit oder Speicherbedarf.

Die Spezifikationen des Systems, auf dem die Evaluation ausgeführt wird, entsprechen denen aus Kapitel 3 (siehe auch Tabelle 3.1).

### 4.1 Funktionsweisen der einzelnen Komponenten

Die einzelnen Komponenten der Kopplung werden einzeln nacheinander analysiert und bewertet. Die Netze der verwendeten Beispieldaten wurden dafür künstlich erzeugt und beruhen nicht auf realen Gebieten oder Daten. Dennoch sollte es mit ihnen möglich sein die allgemeine Funktion der Kopplung auf Schwachstellen und Fehlerquellen zu untersuchen. Zunächst wird in Abschnitt 4.1.1 die Erstellung von Abfahrtszeiten und anschließende Erzeugung von Fahrzeugen betrachtet. Das zweite Beispieldaten in Abschnitt 4.1.2 soll hingegen zur Analyse der Routenplanung und der in Abschnitt 3.2.3 beschriebenen Kapazitätsbeschränkungen eingesetzt werden.

Zwar wurden die Netze so konzipiert, dass sie bei der Analyse der entsprechenden Komponente möglichst viele verschiedene Fälle abdecken. Dennoch wird das System nur in einem gewissen Rahmen getestet. Es können daher an dieser Stelle möglicherweise nicht alle Sonderfälle näher untersucht werden. Um die Qualität der Auswertungen zu verbessern wird versucht, die statistischen Schwankungen aufgrund der Verwendung von Wahrscheinlichkeiten möglichst gut herauszurechnen.

Dafür wurden bei jeder Untersuchung mehrere Durchgänge durchgeführt und die Daten anschließend in sogenannten "BoxPlots" dargestellt, die neben dem Median zusätzlich Quantile und gegebenenfalls auftretende Ausreißer visualisieren.

#### 4.1.1 Szenario a): Bezirke und Kreuzungen

Das in Abbildung 4.1 gezeigte Beispieldatenzenario enthält einige Knoten und Strecken, die in einem Rastermuster angeordnet sind. Es existieren vier Bezirke, die an jeweils einer der vier Ecken des Netzes positioniert sind und Anbindungen an die jeweils drei nächsten Knoten besitzen. Außerdem befindet sich in der Mitte ein Teilnetz, welches mikroskopisch von SUMO simuliert werden soll. Es wird in VISUM ebenfalls als Bezirk visualisiert, ist jedoch weder an Knoten angebunden noch besitzt es Einträge in der verwendeten OD-Matrix.

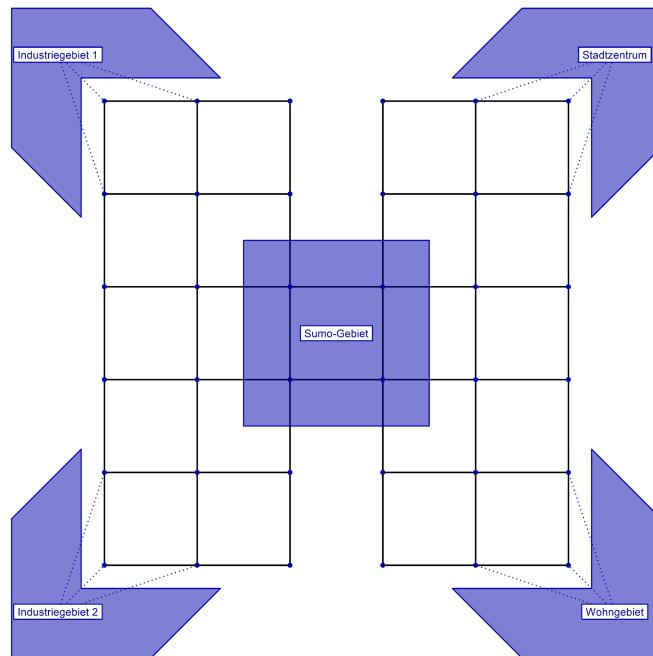


Abbildung 4.1 – Netz des Beispieldatenzenarios "BezirkeUndKreuzungen"

Das Netz wurde so konzipiert, dass umgelegte Fahrzeugflüsse zwischen der linken und rechten Seite (horizontal) immer durch das mikroskopische Gebiet verlaufen. Im Gegensatz dazu werden vertikale Verkehrsflüsse, wie beispielsweise vom Bezirk "Wohngebiet" zum "Stadtzentrum", vermutlich nur makroskopisch simuliert. Für das implementierte System sind vor allem diejenigen Verläufe relevant, die das mittlere Gebiet beinhalten. Ausschließlich makroskopische Verkehrsflüsse werden nur von VISUM modelliert und haben daher lediglich einen geringen Einfluss auf die

Simulatorenkopplung. Sie werden deshalb im Folgenden nicht extra beschrieben.

Die erlaubte Maximalgeschwindigkeit wurde im gesamten Netz auf 50 km/h und die Länge jeder Strecke auf einen Kilometer festgelegt.

Für die Ausführung der Kopplung wurden zunächst alle Strecken mit einer Kapazitätsbeschränkung von 100 Fahrzeugen pro Zeitintervall versehen. Bei den in SUMO liegenden wurde die Kapazität anschließend auf 500 erweitert, da durch die künstlich erzeugte Engstelle an dieser Stelle ein erhöhter Verkehrsfluss erwartet werden kann.

Das Analyseintervall wurde auf 15 Stunden beschränkt, wobei die Umlegung in VISUM stündliche Zeitintervalle berechnet. Bei einer Verlängerung der Simulation könnte die Berechnungszeit die maximale Nutzungsdauer der Studentenversion von 45 Minuten übersteigen. Jedoch sollte auch diese Zeitspanne bereits ausreichen, um Aussagen über die Funktionalitäten und Algorithmen treffen zu können. Zur Verbesserung der Qualität der Daten wurden mehrere Durchläufe gestartet und ausgewertet. Damit kann der Einfluss von vereinzelt auftretenden starken Schwankungen verringert werden.

Die verwendete Nachfrageganglinie ist in Abbildung 4.2 zu sehen. Sie wurde so gewählt, dass die Gewichtungen der Zeitschritte kontinuierlich ansteigt. Dadurch wird die Kopplung in diesem Szenario bei unterschiedlichen Auslastungen getestet und es kann untersucht werden, ob die Anzahl der Fahrzeuge Einfluss auf die Qualität der Kopplung nimmt.

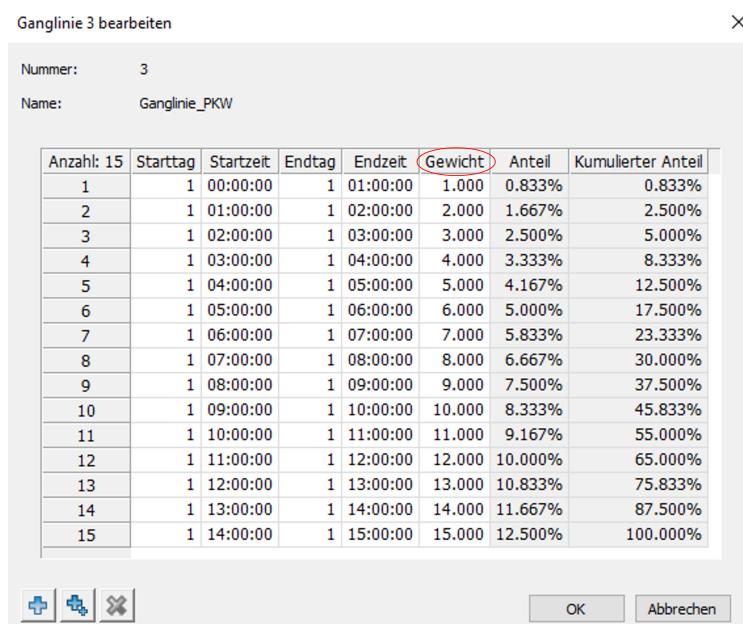


Abbildung 4.2 – Nachfrageganglinie mit steigender Auslastung

Für dieses Szenario wurde die Größe der Verkehrsflüsse zwischen jeweils zwei Bezirken einheitlich gewählt. Unterschiedlich stark ausgelastete Bezirksrouten sind für die Bewertung der folgenden Funktionen nicht relevant. Um das Beispiel möglichst einfach zu halten und auf die wesentlichen Elemente zu beschränken, wurde daher eine einheitliche Parametrisierung gewählt. Es wurden dabei nacheinander die beiden nachfolgend gezeigten OD-Matrizen mit den Zellwerten 100 beziehungsweise 300 getestet. Auch dieser Vergleich soll mögliche Unterschiede zwischen geringem und hohem Verkehrsaufkommen aufzeigen.

| Matrixeditor (Matrix '2 M_PKW') |                   |                        |                 |                   |                  |                        |
|---------------------------------|-------------------|------------------------|-----------------|-------------------|------------------|------------------------|
| 5 x 5                           | Name              | 1<br>Industriegebiet 1 | 2<br>Wohngebiet | 3<br>Stadtzentrum | 4<br>Sumo-Gebiet | 5<br>Industriegebiet 2 |
|                                 | Summe             | 300.00                 | 300.00          | 300.00            | 0.00             | 300.00                 |
| 1                               | Industriegebiet 1 | 300.00                 | 0.00            | 100.00            | 100.00           | 0.00                   |
| 2                               | Wohngebiet        | 300.00                 | 100.00          | 0.00              | 100.00           | 0.00                   |
| 3                               | Stadtzentrum      | 300.00                 | 100.00          | 100.00            | 0.00             | 100.00                 |
| 4                               | Sumo-Gebiet       | 0.00                   | 0.00            | 0.00              | 0.00             | 0.00                   |
| 5                               | Industriegebiet 2 | 300.00                 | 100.00          | 100.00            | 0.00             | 0.00                   |

(a) Jeweils 100 Fahrzeuge

| Matrixeditor (Matrix '2 M_PKW') |                   |                        |                 |                   |                  |                        |
|---------------------------------|-------------------|------------------------|-----------------|-------------------|------------------|------------------------|
| 5 x 5                           | Name              | 1<br>Industriegebiet 1 | 2<br>Wohngebiet | 3<br>Stadtzentrum | 4<br>Sumo-Gebiet | 5<br>Industriegebiet 2 |
|                                 | Summe             | 900.00                 | 900.00          | 900.00            | 0.00             | 900.00                 |
| 1                               | Industriegebiet 1 | 900.00                 | 0.00            | 300.00            | 300.00           | 0.00                   |
| 2                               | Wohngebiet        | 900.00                 | 300.00          | 0.00              | 300.00           | 0.00                   |
| 3                               | Stadtzentrum      | 900.00                 | 300.00          | 300.00            | 0.00             | 300.00                 |
| 4                               | Sumo-Gebiet       | 0.00                   | 0.00            | 0.00              | 0.00             | 0.00                   |
| 5                               | Industriegebiet 2 | 900.00                 | 300.00          | 300.00            | 0.00             | 0.00                   |

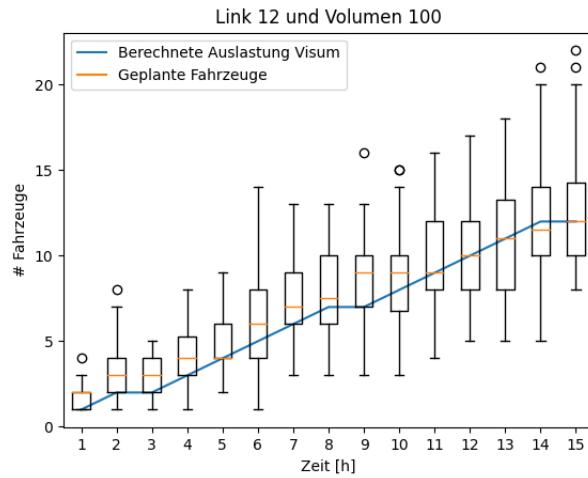
(b) Jeweils 300 Fahrzeuge

**Abbildung 4.3 – OD-Matrizen mit unterschiedlichen Auslastungen der Bezirksbeziehungen im Analyseintervall**

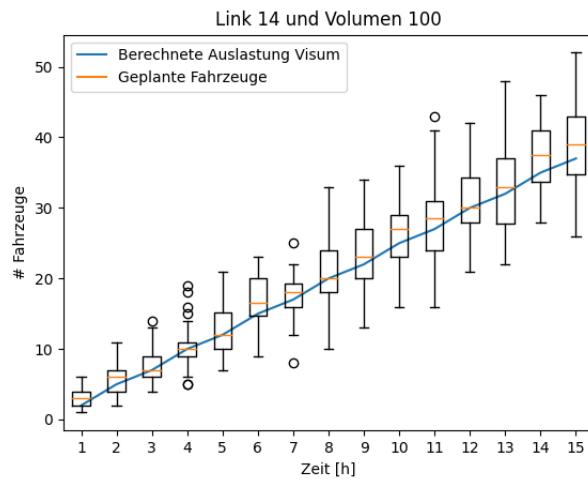
#### 4.1.1.1 Fahrzeugerzeugung

Zunächst wurde die Planung von Fahrzeugerzeugungen betrachtet. Dafür sollte die Anzahl der mit der Exponentialverteilung geplanten Fahrzeuge mit den berechneten Auslastungen in VISUM verglichen werden. Es wird dabei die Anwendbarkeit des in Abschnitt 2.3 beschriebenen Vorgehens bewertet. Für die Auswertung wurden zunächst das jeweilige makroskopischen Volumen sowie die Anzahl der geplanten Events nach jedem Durchgang für jede der Strecken gespeichert.

Zunächst soll die OD-Matrix aus Abbildung 4.3 (a) verwendet werden. Die Kopplung wurde dafür 40 Mal ausgeführt und die Ergebnisse protokolliert. Für die Strecken 12 und 14 sind sie in Abbildung 4.4 als BoxPlots dargestellt. Diese Straßen wurden beispielhaft ausgewählt, da der Verkehrsfluss über Strecke 14 in jedem Zeitschritt doppelt so groß ist wie über Strecke 12. Die Diagramme zu den restlichen relevanten Übergangsstrecken sind im Anhang in Abbildung 6.1 zu finden.



(a) Strecke 12

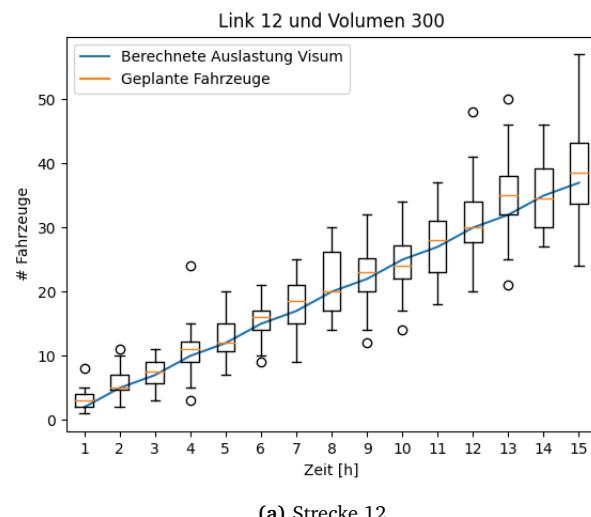


(b) Strecke 14

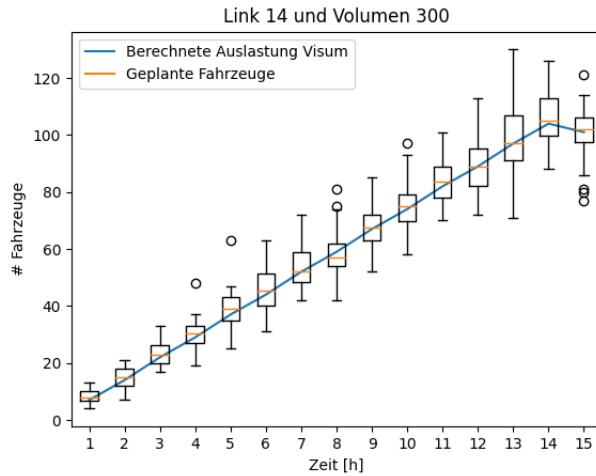
**Abbildung 4.4 – Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (100er OD-Matrix)**

Es ist deutlich zu erkennen, dass sich der Median der Anzahl mit dem Algorithmus geplanter Fahrzeuge ähnlich zur von VISUM berechneten Auslastung der Strecken verhält. Beim direkten Vergleich der beiden Diagramme fällt auf, dass die Boxen, in der die mittleren 50% der Daten liegen, beim Diagramm von Strecke 14 in der Regel wesentlich kleiner sind als bei dem von Strecke 12. Diese geringere Streuung der Daten lässt sich wahrscheinlich auf die höhere Anzahl erzeugter Fahrzeuge zurückführen. Die Längen der Antennen nehmen hingegen auf einer einzeln betrachteten Strecke bei steigender Auslastung zu. Diese sogenannten Whisker zeigen

das Minimum beziehungsweise Maximum einer Datenreihe, wobei Ausreißer außer Acht gelassen werden. Diese sind im Diagramm vereinzelt als Kreise zu erkennen. Um die Beobachtungen noch einmal zu überprüfen wird die gleiche Analyse nun für die OD-Matrix in Abbildung 4.3 (b) wiederholt. Es werden auch diesmal wieder 40 Durchgänge betrachtet. Die Ergebnisse für die Strecken 12 und 14 sind in Abbildung 4.5 zu sehen und die der anderen wieder im Anhang (Kapitel 6).



(a) Strecke 12



(b) Strecke 14

**Abbildung 4.5 – Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (300er OD-Matrix)**

Diese beiden Diagramme bestätigen die Beobachtung aus Abbildung 4.4. Der Median nähert sich bei steigendem Verkehrsaufkommen immer näher an die Vergleichskurve aus VISUM an und auch der Interquartilsabstand, also die Größe der Boxen, ist im Verhältnis zu Anzahl der erzeugten Fahrzeuge wesentlich kleiner. Der Abstand zwischen Minimum und Maximum ist relativ zur Anzahl ebenfalls gesunken. Jedoch kommt es wie auch bereits bei der OD-Matrix mit geringeren Werten auch hier zu vereinzelten Ausreißern.

Es lässt sich also zusammenfassend sagen, dass die mithilfe der Exponentialverteilung erstellten Abfahrtszeiten annähernd dieselbe Anzahl Fahrzeuge erzeugen, wie von VISUM berechnet wurden. Dabei ist besonders bei steigendem Verkehrsaufkommen eine geringere Abweichung zu erkennen und die Streuung sinkt bei höheren Auslastungen ebenfalls. Bei kleineren Fahrzeugzahlen sind die Schwankungen zwar größer, die Annäherung ist jedoch auch bei Auslastungen von 5 bis 10 Fahrzeugen bereits zu erkennen (siehe Abbildung 4.4 (a)). Daher kann die Exponentialverteilung bei der Erstellung individueller Abfahrtszeiten von Fahrzeugen sinnvoll eingesetzt werden, um ein realistisches individuelles Abfahrtsverhalten von Fahrzeugen zu erzeugen. Es könnte jedoch möglicherweise bei noch niedrigeren Auslastungen zu größeren Schwankungen kommen, mit denen der Einsatz dieser Kopplung nicht mehr sinnvoll wäre. Das Festlegen eines Grenzwertes für diese Abweichungen gegenüber einer nur makroskopischen Simulation liegt allein beim Nutzer und ist, wie soeben gezeigt wurde, stark von den Auslastungen des Anwendungsbereichs abhängig.

#### 4.1.1.2 Routenplanung

Als nächstes soll die Komponente betrachtet werden, die sich mit der Erzeugung von Routen und der anschließenden Übergabe an die erstellten Fahrzeuge beschäftigt. Es werden an dieser Stelle wieder die beiden in Abschnitt 3.2.2 vorgestellten Varianten unterschieden:

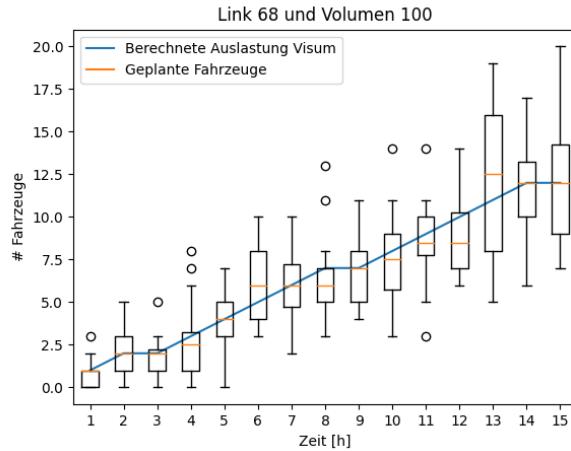
- **Variante 1:** Fahrzeuge fahren die Routen ab, die anhand von Abbiegewahrscheinlichkeiten aus VISUM erstellt werden.
- **Variante 2:** Die Routen werden von SUMO geplant, nur die Start- und Zielstrecken werden von der erzeugten Route übernommen.

Im Folgenden wird nur die Strecke 68 betrachtet und mit beiden Varianten sowie den in Abbildung 4.3 dargestellten OD-Matrizen getestet. Dennoch lassen sich alle Beobachtungen und Auswertungen auch an den restlichen Straßen feststellen. Die übrigen Diagramme sind dafür wieder im Anhang zu finden (Kapitel 6).

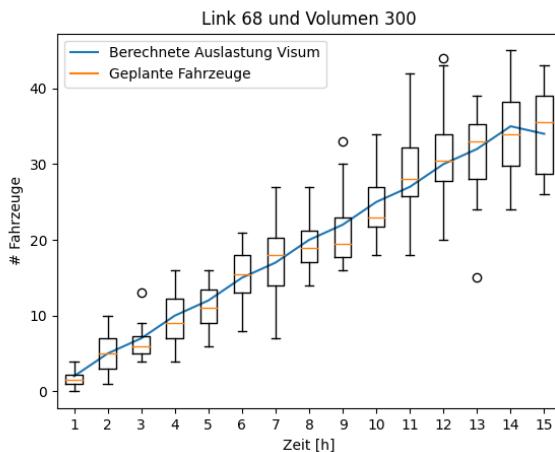
Für die Auswertung der ersten Möglichkeit werden alle erstellten Routen und die

jeweiligen Abfahrtszeiten der Fahrzeuge gespeichert. Für jeden Zeitschritt wird dann für jede Strecke gezählt, wie oft sie in den Streckenverläufen vorkommt. Diese Anzahl entspricht der Auslastung der entsprechenden Strecke im aktuellen Zeitintervall. Zur Vereinfachung werden dabei Fahrzeuge immer in dem Zeitschritt gezählt, in dem sie erzeugt werden. Eine Aufteilung der Strecke auf mehrere Intervalle wäre durch das dafür benötigte Verfolgen der einzelnen Fahrzeugobjekte sehr aufwendig und wird an dieser Stelle deshalb nicht umgesetzt. Es kann außerdem davon ausgegangen werden, dass die Zahl der betroffenen Fahrzeuge relativ gering ist und keinen großen Einfluss auf die Auswertung ausübt.

Bei dieser Analyse wurden 20 Durchgänge pro OD-Matrix und Variante der Routenplanung durchgeführt. In Abbildung 4.6 sind die BoxPlots der Auswertung von Variante 1 für Strecke 68 zu sehen.



(a) 100er OD-Matrix



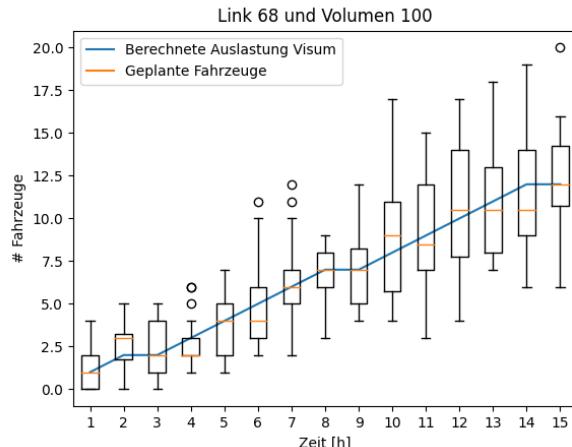
(b) 300er OD-Matrix

**Abbildung 4.6 – Auswertung der Routenplanung bei unterschiedlichem Verkehrsaufkommen, Variante 1**

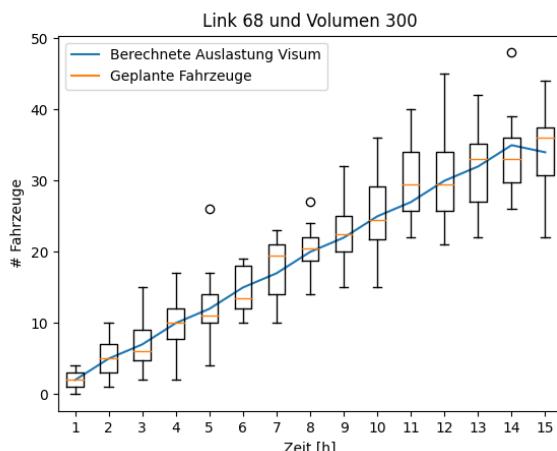
In diesen Diagrammen ist ähnlich wie bereits in Abschnitt 4.1.1.1 eine Annäherung der Mediane der gezählten Fahrzeuge an die berechneten Werte aus VISUM festzustellen. Auch bei dieser Auswertung ist der Interquartilsabstand beim höherem Verkehrsaufkommen in (b) geringer. Die Streuung der Werte nimmt also bei Verwendung der Variante 1 wieder bei steigender Auslastung ab.

Dieselbe Auswertung wurde ebenfalls für Möglichkeit 2 durchgeführt. Bei dieser Variante werden die Routen der Fahrzeuge bei Aufruf des Befehls `changeTarget()` von SUMO neu berechnet. Die neuen Streckenverläufe werden daraufhin mit `getRoute()`

über die jeweilige Fahrzeug-ID abgefragt. Anschließend können die Auslastung wie bereits beschrieben gezählt werden. Die Ergebnisse sind in Abbildung 4.6 zu sehen.



(a) 100er OD-Matrix



(b) 300er OD-Matrix

**Abbildung 4.7 – Auswertung der Routenplanung bei unterschiedlichem Verkehrsaufkommen, Variante 2**

Die bisher getroffenen Beobachtungen treffen wie erwartet ebenfalls bei dieser Auswertung zu. Es fällt jedoch außerdem auf, dass sich die Diagramme der beiden Varianten der Routenplanung nur sehr geringfügig voneinander unterscheiden. Bei Variante 2 sind zwar die Interquartilsabstände in den Zeitschritten zwischen 10 und 12 etwas größer. Das kann jedoch auch an den statistischen Schwankungen liegen und ist nicht aussagekräftig genug für eine entsprechende Schlussfolgerung. In diesem einfachen Beispielszenario ist also kein nennenswerter Unterschied zwischen

beiden Varianten zu erkennen. Das könnte möglicherweise an der geringen Größe des SUMO-Netzes und damit den wenigen alternativen Routen liegen. Die beiden Möglichkeiten sollten daher noch mit einem größeren und komplexeren Beispiel getestet werden, um Unterschiede besser feststellen zu können. Dabei wäre es sinnvoll, wenn die mikroskopische Simulation genauere Daten zu Umgebung, Fahrverhalten von Fahrzeugen oder ähnlichen Informationen enthalten würde. Dann kann erwartet werden, dass bei Variante 2 die von SUMO gewählten Routen stärker von denen in VISUM abweichen. Da die mikroskopische Simulation mehr Informationen als die makroskopische in Betracht ziehen kann ist das jedoch kein Fehler der Kopplung, sondern sogar wünschenswert. Die Auswirkungen der Zusatzinformationen können dann über die bereits beschriebenen Kapazitätsbeschränkungen zur Anpassung von VISUM verwendet werden. Die höhere Genauigkeit von SUMO wird ebenfalls in Abschnitt 4.1.2 noch einmal betrachtet und ausgenutzt.

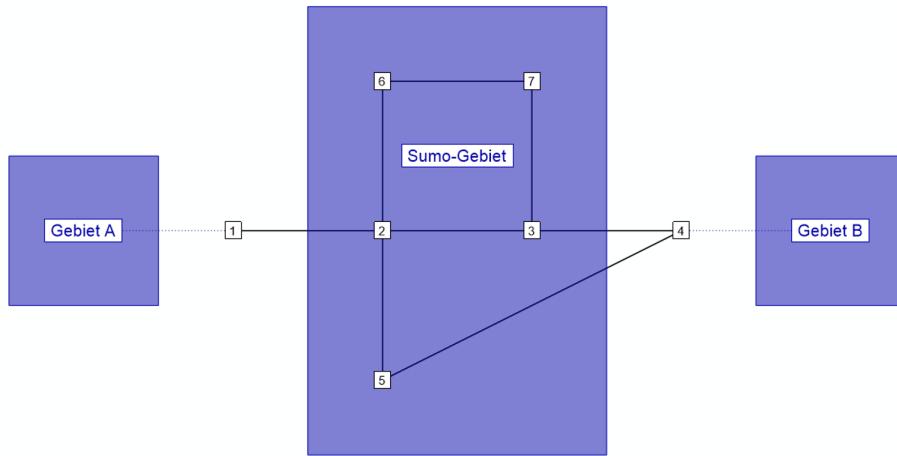
Es lässt sich also zusammenfassend feststellen, dass sich auch diese Komponente besonders bei hohem Verkehrsaufkommen sehr ähnlich zu der verglichenen makroskopischen Simulation verhält und die Schwankungen bei steigender Fahrzeugzahl sinken. Damit lässt sich vermuten, dass die Routenplanung anhand von Abbiegewahrscheinlichkeiten gut geeignet ist, um ein Verkehrsgeschehen mit individuellen Fahrzeugobjekten aus aggregierten Fahrzeugflüssen zu erzeugen. Die Unterschiede der beiden Möglichkeiten, einem Fahrzeug in SUMO eine Route zu übergeben konnten jedoch an diesem Beispiel nicht ausreichend analysiert werden. Sie müssen zur besseren Auswertung nochmals an anderen Beispielen untersucht werden.

Die beiden Varianten der Routenplanung sind vermutlich nur speziell bei diesem mikroskopischen Simulator möglich. Da der Fokus dieser Arbeit auf der Kopplung an sich liegt und die Beispielsimulatoren lediglich zur Bewertung der Funktionsweise dienen sollen, werden die beiden Möglichkeiten an dieser Stelle nicht weiter analysiert.

#### 4.1.2 Szenario b): Streckensperrung in SUMO

Dieses Beispielszenario wurde speziell zum Testen der Kapazitätsbeschränkungen und der Routenanpassung über SUMO konzipiert. Es wurde bereits in Abschnitt 3.2.3 kurz vorgestellt, weswegen die Beschreibung an dieser Stelle kurz gehalten werden soll. Die Auswertung orientiert sich stark an den in Kapitel 3 bereits beschriebenen Schritten und soll lediglich die dort genannten Fälle mit Daten testen und bewerten. Nach dieser Untersuchung wird noch kurz auf den Parameter "Threshold" und seinen Einfluss auf die erneuten Umlegungen eingegangen. Das verwendete Streckennetz ist in Abbildung 4.8 noch einmal zu sehen. Es besteht im Wesentlichen aus zwei Bezirken und drei unterschiedlich langen Verbindungs Routen. Der mikroskopische

Bereich ist in der Mitte zwischen den Gebieten A und B zu finden. Das Beispiel wurde sehr einfach gehalten, um die Auswirkungen der Beschränkungen erkennen und andere Einflüsse auf das Verkehrsverhalten ausschließen zu können.



**Abbildung 4.8 – Netz des Beispieldatenmodells "Streckensperrung"**

In Abschnitt 3.2.3 wurden die beiden folgenden interessanten Szenarien bereits beschrieben:

- Stau auf einer mikroskopischen Strecke
- (Vorübergehende) Sperrung einer mikroskopischen Strecke, beispielsweise durch einen Unfall

geht in beiden Fällen darum, dass SUMO durch die höhere Genauigkeit Kenntnis von Informationen hat, die VISUM bei der Umlegung nicht einbeziehen konnte. Die Berechnung der makroskopischen Auslastungen wird jedoch zunächst normal berechnet. An der Übergangsstelle von SUMO nach VISUM wird der Unterschied der gezählten Fahrzeuge zum berechneten Wert erkannt und daraufhin der Algorithmus zur Kapazitätsbeschränkung angewendet (siehe Abschnitt 3.2.3). Durch diese Beschränkungen und die darauffolgende erneute Umlegung soll die makroskopische Simulation an die genaueren Informationen angepasst werden.

Das Szenario "Stau" ist im Prinzip nur eine abgeschwächte Variante einer Straßensperrung und das zu erwartende Verhalten der Kopplung sehr ähnlich. Daher soll es an dieser Stelle genügen den Fall einer Sperrung genauer zu betrachten und anschließend die wenigen Unterschiede bei einem Stau zu beschreiben. Wie in Abschnitt 3.2.3 bereits differenziert werden auch an dieser Stelle zwei leicht unterschiedliche Netze untersucht. Zunächst wird das in Abbildung 4.8 gezeigte verwendet, bevor anschließend ein Teil entfernt beziehungsweise gesperrt wird.

Als Kapazitätsbeschränkungen der Straßen wurden wieder 100 Fahrzeuge pro Zeitschritt und eine maximal mögliche Geschwindigkeit von 50 km/h gewählt. Die Strecken sind, mit Ausnahme der zwischen Knoten 5 und 4, alle einen Kilometer lang. Das Analyseintervall ist wieder auf 0 bis 15 Uhr beschränkt und die Nachfrageganglinie entspricht ebenfalls der in Abbildung 4.2 dargestellten. Die verwendete OD-Matrix ist in Abbildung 4.9 zu sehen. Es wird außerdem nur ein Verkehrsfluss von Gebiet A nach B modelliert, da die Gegenrichtung für die folgenden Betrachtungen irrelevant ist. Die Untersuchungen, die bei diesem Netz in der Hinrichtung angestellt werden können, sind in der Gegenrichtung weder möglich noch sinnvoll, da das Szenario nur für eine Richtung konzipiert wurde.

| Matrixeditor (Matrix '1 Matrix1') |             |        |          |          |             |  |
|-----------------------------------|-------------|--------|----------|----------|-------------|--|
| 3 x 3                             |             |        | 1        | 2        | 3           |  |
|                                   | Name        |        | Gebiet A | Gebiet B | Sumo-Gebiet |  |
|                                   |             | Summe  | 0.00     | 100.00   | 0.00        |  |
| 1                                 | Gebiet A    | 100.00 | 0.00     | 100.00   | 0.00        |  |
| 2                                 | Gebiet B    | 0.00   | 0.00     | 0.00     | 0.00        |  |
| 3                                 | Sumo-Gebiet | 0.00   | 0.00     | 0.00     | 0.00        |  |

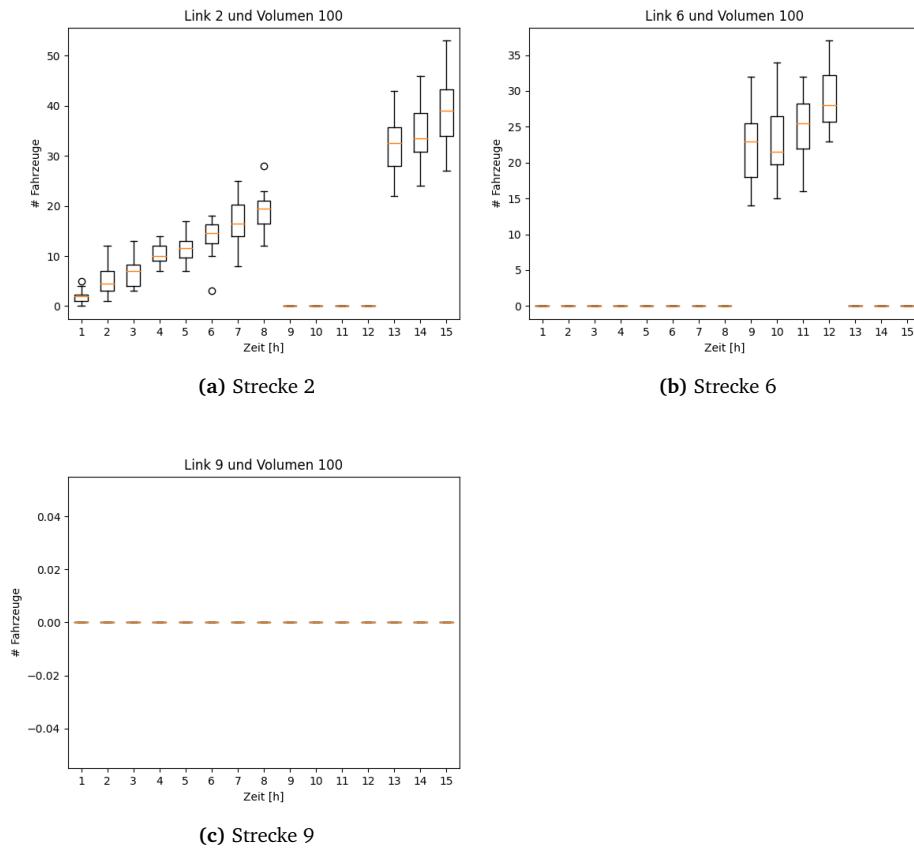
Abbildung 4.9 – Verwendete OD-Matrix für das Beispielszenario "Streckensperrung"

Da bei dieser Untersuchung die komplette Sperrung einer Straße betrachtet wird, ist die Wahl der Art der Routenplanung irrelevant. Beide Varianten würden denselben alternativen Streckenverlauf wählen.

Für die Ausführung wurde die Strecke zwischen Knoten 2 und 3 für den Zeitraum von 8 bis 12 Uhr gesperrt. Eine Sperrung kann über die Traci-Schnittstelle mit den Methoden `setDisallow()` erstellt und mit `setAllow()` wieder aufgehoben werden. Es wurden jeweils die Auslastungen der mikroskopischen Straßen ausgelesen und gespeichert. Dafür kam wieder der im vorherigen Abschnitt beschriebene Algorithmus zum Einsatz, der zum Bestimmen der Auslastung die Sammlung von gefahrenen Routen verwendet. Für die Auswertung werden die Strecken 2 (von Knoten 2 nach 3), 9 (Knoten 2 nach 5) und 6 (Knoten 2 nach 6) betrachtet. Da sie jeweils den Anfang einer der drei Streckenverläufe darstellen, müssen die übrigen nicht extra ausgewertet werden. Sie sind jedoch der Vollständigkeit halber im Anhang zu finden (Kapitel 6). Daneben ist noch die Strecke 3 (von Knoten 3 nach 4) interessant, die eine der beiden Übergangsstrecken zurück nach VISUM ist. Es wurden zur besseren Qualität der Daten wieder 20 Durchgänge gestartet und die Daten von diesen verwendet.

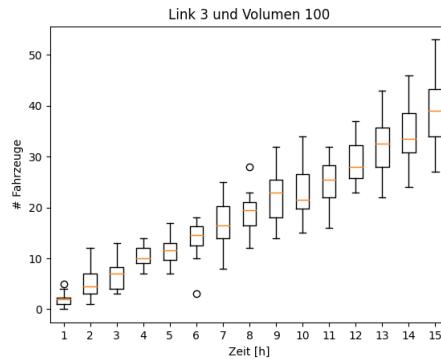
In Abbildung 4.10 sind die Ergebnisse der Strecken 2, 6 und 9 zu sehen. Letztere besitzt im gesamten Analyseintervall keine Auslastung, da es eine Alternativroute gibt

und damit die Kapazitätsbeschränkungen in VISUM bei diesem kleinen Beispiel noch nicht angewandt werden. Daher wird die Umplanung nur von der mikroskopischen Routenplanung durchgeführt und der Verkehrsfluss wird in der Zeit der Sperrung von 8 bis 12 Uhr über die Knoten 2, 6, 7 und 3 statt über den Knoten 5 geleitet. Diese Verschiebung der Auslastung ist in der Abbildung in **(a)** und **(b)** zu sehen.



**Abbildung 4.10** – Auswertung der relevanten Strecken bei vorhandener Alternativroute in SUMO

In der folgenden Abbildung ist nun die Auslastung an der Übergangsstrecke von Knoten 3 nach 4 noch einmal dargestellt:

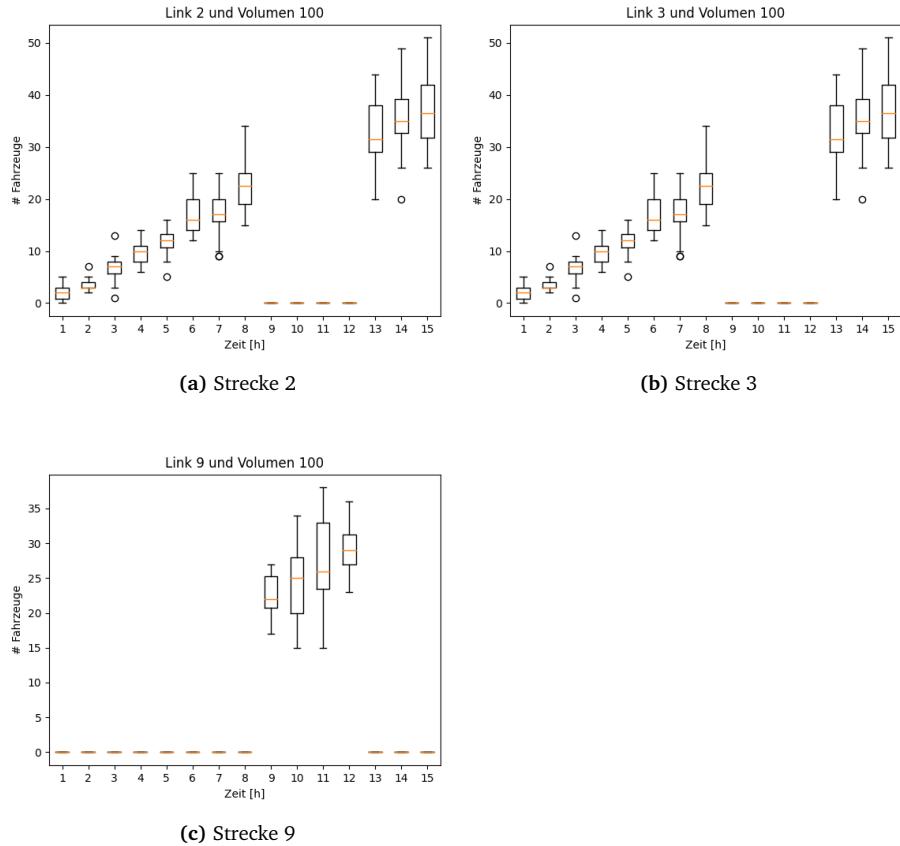


**Abbildung 4.11 – Auswertung der Übergangsstrecke zwischen 3 und 4 bei vorhandener Alternativroute in SUMO**

An dieser Stelle ist kein Einfluss durch die Sperrung erkennbar, da es eine passende Alternativroute zur Zielstrecke gibt und diese nicht wesentlich länger ist. Bei starker Verlängerung dieser Ausweichroute, wäre an der Strecke vermutlich eine etwas geringere Auslastung zu bemerken. Außerdem würde je nach Threshhold möglicherweise der Algorithmus der Kapazitätsbeschränkungen angewendet werden, wodurch einige Fahrzeuge nach der Umplanung auf die untere Alternative ausweichen würden.

Nun soll das Szenario so abgeändert werden, dass keine Umplanung durch SUMO mehr möglich ist. Diese Änderung wurde auch in Abschnitt 3.2.3 bereits beschrieben. Es werden dafür die drei oberen Strecken 2-6, 6-7 und 7-3 in VISUM für den Fahrzeugtyp "PKW" gesperrt und auch die mikroskopische Konfiguration entsprechend angepasst. Alle anderen Bedingungen, wie OD-Matrix, Nachfrageganglinien und Kapazitäten der Strecken, bleiben unverändert.

In Abbildung 4.12 sind die Auslastungen der nun interessanten Strecken 2 (von Knoten 2 nach 3), 3 (3 nach 4) und 9 (2 nach 5) im Zeitverlauf dargestellt. Es ist festzustellen, dass der Verkehrsfluss im Zeitraum der Sperrre vollständig über die untere Alternativroute geleitet wird. Daraus kann geschlossen werden, dass die Kapazitätsbeschränkung mit anschließender neuer Umlegung funktioniert hat. Es wäre sonst in SUMO nicht möglich, eine andere Zielstrecke bei der Routenplanung auszuwählen.



**Abbildung 4.12 – Auswertung der relevanten Strecken ohne Alternativroute in SUMO**

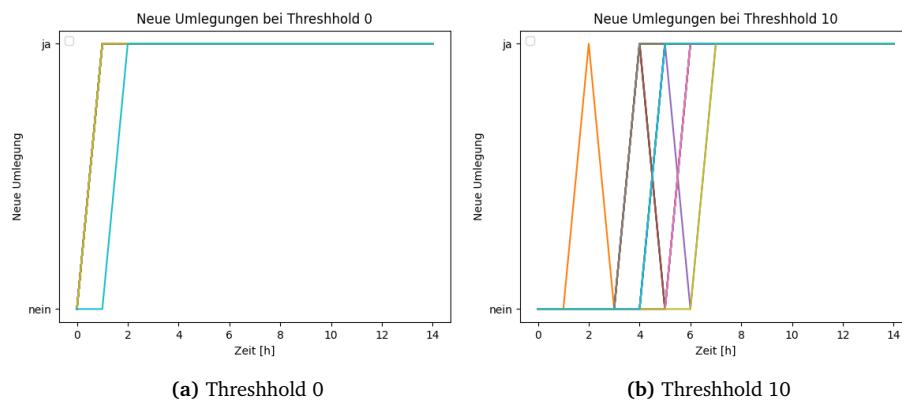
Bei einem Stau würden die Auslastungen der Strecken ähnlich aussehen. Im Szenario mit Alternativroute würde ein Teil der Fahrzeuge über diese umgeleitet werden, während einige vermutlich auf der mittleren Strecke bleiben. Falls die obere Umgehung entfernt oder gesperrt wird, müssen die Fahrzeuge zunächst die gestauta Strecke befahren. Dadurch entsteht eine Verzögerung und es werden wahrscheinlich nicht alle Fahrzeuge rechtzeitig an der Übergangsstrecke nach VISUM ankommen. Daher wird je nach Threshold und Größe der Differenz gegebenenfalls die Kapazitätsbeschränkung und anschließende erneute Umlegung durchgeführt. Dadurch würden anschließend einige Fahrzeuge über die untere Ausweichroute geplant werden.

Es lässt sich also feststellen, dass das Verhalten bei einem Stau und einer Sperrung ähnlich ist. Daher soll die bisherige Auswertung bezüglich einer gesperrten Strecke an dieser Stelle genügen.

Der Threshhold wurde bei allen bisher gezeigten Varianten auf 0 gesetzt. Das be-

deutet, dass Kapazitätsbeschränkungen und die darauffolgende neue Umlegung schon bei geringfügig unterschiedlichen Werten durchgeführt werden und es keinen Toleranzbereich von Abweichungen gibt. Mit Vergrößerung des Parameters kann der Rechenaufwand möglicherweise begrenzt werden, da kleine Schwankungen keine neue Umlegung auslösen.

In den Diagrammen in Abbildung 4.13 ist dargestellt, ob jeweils eine neue Umlegung gestartet wurde oder nicht. Es wurde dafür das Beispielszenario "Streckensperre" gewählt. Der Threshold in (a) auf 0 und in (b) auf 10 festgelegt worden. Die einzelnen Linien im Diagramm entsprechen den 20 durchgeföhrten Durchgängen. Es ist deutlich zu erkennen, dass beim höheren Threshold bei geringem Verkehrsaukommen wesentlich weniger neue Umlegungen durchgeführt werden. Hingegen wird bei Threshold 0 ab 2 Uhr in jedem Zeitschritt eine neue Umlegung durchgeführt. Eine neue Berechnung der makroskopischen Auslastungen ist dabei immer genau dann nötig, wenn sich die Kapazitätsbeschränkungen von mindestens einer Strecken geändert haben. Da die OD-Matrizen Nachfrageganglinien ebenfalls aus den vorherigen Beispielen gewählt werden lässt sich feststellen, dass bei steigendem Verkehrsvolumen neue Umlegungen selbst bei einem Parametrisierung mit 10 Fahrzeugen Abweichungen dennoch stattfindet. Bei der Wahl des Thresholds sollte also auch die Auslastung der Straßen und die möglicherweise höheren Schwankungen bei höherem Volumen einbezogen werden.



**Abbildung 4.13 – Einfluss des Thresholds beim Algorithmus der Kapazitätsbeschränkungen**

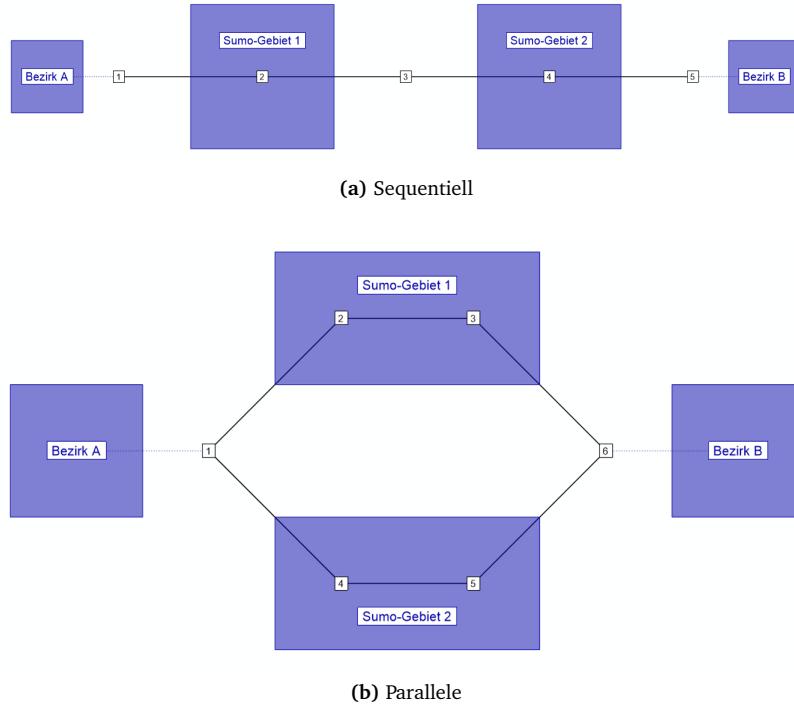
Die Auswertung in diesem Abschnitt hat also gezeigt, dass der entwickelte Algorithmus mit Kapazitätsbeschränkungen gut funktioniert, um die makroskopische Umlegung an die zusätzlichen mikroskopischen Verkehrsinformationen anzupassen. Es wurden nur die Fälle "Stau" und "Sperrung" näher betrachtet, da alle anderen Einflüsse aus SUMO vermutlich zu einem der beiden führen. Daher soll die Betrachtung

an dieser Stelle genügen. Weitere Fälle können zum Beispiel geringere Geschwindigkeiten, Unfälle, verändertes Fahrverhalten und Ähnliches sein. Der Threshold ist ein Parameter, über den der Toleranzbereich beim Algorithmus der Kapazitätsbeschränkungen angepasst werden kann. Darüber kann somit die Genauigkeit damit zusammenhängend der Rechenaufwand geringfügig angepasst werden.

## 4.2 Verwendung mehrerer mikroskopischer Gebiete

Bei Anwendung der Kopplung auf eine größere Stadt kann es sinnvoll sein statt einem großen zusammenhängenden Teilnetz besser mehrere kleine mit SUMO zu simulieren. An dem im Folgenden beschriebene Beispielszenario soll diskutiert werden, ob das entwickelte System auch mit mehreren Teilgebieten auf mikroskopischer Abstraktionsebene funktioniert. Dafür wurden zwei kleine Beispielszenarien erstellt:

1. Es wird eine Straße modelliert, die über mehrere gerade verlaufende Teilstrecken vom Bezirk A nach B führt (Abbildung 4.14 (a)). Auf dieser Route liegen nacheinander zwei mikroskopische Gebiete, die sich jedoch nicht berühren oder überschneiden. Sie sind also sequentiell hintereinander angeordnet.
2. Es wurde ein Netz erstellt, in dem sich die Route zwischen den Bezirken A und B zunächst in zwei parallel verlaufende, gleich lange Routen aufteilt und anschließend wieder zusammenläuft (Abbildung 4.14 (b)). Jede der beiden parallelen Strecken soll jeweils ein SUMO-Gebiet enthalten. Es soll dabei davon ausgegangen werden, dass sich der Verkehrsfluss auf beide Streckenverläufe aufteilt. Damit soll die Funktionsweise der Kopplung bei parallel verlaufenden mikroskopischen Gebieten betrachtet werden.



**Abbildung 4.14 – Unterschiedliche Möglichkeiten, mehrere SUMO-Gebiete anzugeordnen**

Der Kopplung wird auch bei mehreren SUMO-Gebieten eine einzelne Liste aller mikroskopischen Knoten übergeben. Dabei wird nicht zwischen der Zugehörigkeit zu den unterschiedlichen Bereichen differenziert. Es geht an dieser Stelle nicht um eine tiefgreifende Analyse der Wechselwirkungen mehrere SUMO-Bereiche und ihrer Einflüsse aufeinander. Es soll lediglich überprüft werden, dass die Komponenten der Kopplung trotz fehlender Zuordnungsmöglichkeit der einzelnen Knoten und Strecken fehlerfrei funktionieren.

In beiden Beispielen sind keine Überschneidungen oder Kontaktpunkte zwischen den beiden mikroskopischen Gebieten vorhanden. Daher werden bei der Routenplanung, die in Abschnitt 3.2.2 beschrieben wurde, immer nur Streckenverläufe durch das entsprechende SUMO-Gebiet erstellt und es werden aufgrund der fehlenden Verbindung keine erstellten Fahrzeuge über beide geplant. Beim sequentiellen Aufbau werden stattdessen an jedem der Übergänge zum Mikroskopischen die Fahrzeuge erstellt und die Route geplant. Beim parallelen Aufbau gibt es noch weniger möglicherweise kritische Punkte, da der Verkehrsfluss bereits an Knoten 1 auf beide Routen aufgeteilt wird und so eine Trennung der beiden mikroskopischen Netze vorhanden ist.

Es wurde also in der Implementierung durch die ausgelesenen Abbiegewahrscheinlichkeiten und das verwendete mikroskopische Netz bereits ein konsistentes Verhalten bei mehreren SUMO-Gebieten sichergestellt.

---

## Kapitel 5

---

### Fazit

---

Das Ziel dieser Arbeit war es, eine Kopplung von Verkehrssimulationen zu entwickeln, die die makroskopische und die mikroskopische Abstraktionsebene verbindet. Dafür wurden konkrete Programme gewählt, die beispielhaft für die verschiedenen Ebenen stehen sollten. Der Schwerpunkt lag bei der Implementierung auf der Umwandlung der unterschiedlichen Simulationsattribute und -werte zwischen den Simulatoren. Dabei sollte die Konsistenz der Daten beider Simulationen gewahrt bleiben, um die sinnvolle und zuverlässige Nutzung der Kopplung zu ermöglichen.

In verschiedenen Beispielszenarien wurden mehrere unterschiedliche Fälle betrachtet, um die Anwendbarkeit und Erhaltung der Konsistenz zu überprüfen. Die Ergebnisse der Auswertungen wurden in Kapitel 4 analysiert. Dabei wurde gezeigt, dass die einzelnen Komponenten der Kopplung funktionieren und sich im Wesentlichen sehr ähnlich zu einer ausschließlich makroskopischen Simulation verhalten. Lediglich bei geringem Verkehrsaufkommen sollte aufgrund der höheren Streuung möglicherweise in Betracht gezogen werden, das Szenario nur mit VISUM zu modellieren.

Das entwickelte System bietet allerdings noch viel Verbesserungspotential. Die wichtigsten Aspekte sind dabei die Optimierung der Genauigkeit der Kopplung und damit die weitere Annäherung an die Realität.

In Abschnitt 3.2.4 wurden bereits einige Vereinfachungen besprochen, die Abstraktionen von der Wirklichkeit darstellen. In der folgenden Liste sind sie noch einmal kurz zusammengefasst:

- Beschränkung auf spezielle Simulatoren (VISUM und SUMO)
- Start- und Endpunkte von Verkehrsflüssen nur in VISUM
- Begrenzung der Anzahl von Start-/Endpunkten von Verkehrsflüssen
- Verwendung ausschließlich eines Fahrzeugtyps (PKW)

- Begrenzung auf einspurige Straßen
- Umlegung in VISUM nur mit dem Algorithmus DUE

Bei den meisten der aufgezählten Vereinfachungen ist es wahrscheinlich möglich, durch Erweiterung oder geringfügige Änderungen an der Kopplung die Realitätsnähe des hybriden Ansatzes zu verbessern. Die Begrenzung der Anzahl von Start-/Endpunkten hingegen lässt sich vermutlich nicht ohne enormen Rechenaufwand komplett entfernen. Natürlich kann dennoch eine höhere Genauigkeit durch Erhöhung der Bezirksanzahl erreicht werden. Es wird jedoch immer eine Abstraktion der Realität bleiben und nicht möglich sein, alle individuellen Start- und Endpunkte jedes einzelnen Fahrzeugs zu modellieren. Für eine Simulation mit festgelegtem Ziel und auf bestimmte Bereiche begrenzte Analyse ist das allerdings auch nicht nötig. Der große Rechenaufwand würde die Nutzung eher erschweren.

Als mögliche Verbesserung beziehungsweise Erweiterung der Evaluation wäre es sinnvoll, die entwickelte Simulatorenkopplung mit realen Daten einer Stadt oder eines Stadtteils auszutesten. Dafür müssen diese jedoch bestenfalls bereits in Form einer OD-Matrix oder ähnlichen Datenstruktur vorliegen. Es werden in einigen Städten wie beispielsweise Erlangen oder Nürnberg bereits regelmäßige Verkehrszählungen an wichtigen Kreuzungen und Straßen durchgeführt ([Erl21] und [Nü21]). Diese Daten müssten allerdings vor der Verwendung in VISUM zunächst aufwendig ausgewertet und umgewandelt werden. Aktuelle wissenschaftliche Veröffentlichungen wie [RNS20] und [Cao+21] diskutieren die verschiedenen Möglichkeiten, um aus Zählungen oder ähnlichen Daten passende OD-Matrizen zu erzeugen. Das Finden einer realistischen Lösung dieser Umwandlung ist dabei eine essentielle Voraussetzung für die effektive Anwendung von hybriden Ansätzen in realen Szenarien.

---

## Abkürzungsverzeichnis

---

|                  |   |
|------------------|---|
| <b>CA</b>        | Cellular Automata                         |
| <b>COM</b>       | Component Object Model                    |
| <b>DLR</b>       | Deutschen Zentrum für Luft- und Raumfahrt |
| <b>DUE</b>       | dynamic user equilibrium                  |
| <b>GKT</b>       | gas-kinetic-based traffic                 |
| <b>GUI</b>       | graphical user interface                  |
| <b>ID</b>        | Identifikationsnummer                     |
| <b>IDE</b>       | integrated development environment        |
| <b>LKW</b>       | Lastkraftwagen                            |
| <b>OD-Matrix</b> | origin-destination-Matrix                 |
| <b>OSM</b>       | Open-Street-Map                           |
| <b>PKW</b>       | Personenkraftwagen                        |
| <b>SUMO</b>      | simulation of urban mobility              |
| <b>VISUM</b>     | Verkehr in Städten Umlegung               |

---

# Abbildungsverzeichnis

---

|     |   |    |
|-----|---|----|
| 2.1 | Beispiele für Schnellansichten zu den Objekttypen "Knoten", "Strecke" und "Abbieger" . . . . .      | 7  |
| 2.2 | Verkehrsflüsse über andere Bezirke nicht möglich . . . . .  | 8  |
| 2.3 | Beispiel für eine OD-Matrix . . . . .   | 8  |
| 2.4 | Beispielnetz zur Erklärung von OD-Matrizen . . . . .  | 9  |
| 2.5 | Ganglinie orientiert an [Ros], zwischen 4 und 12 Uhr . . . . .                                      | 9  |
| 2.6 | Stündliche Auslastungen auf einer Strecke zwischen beiden Bezirken . . . . .                        | 9  |
| 2.7 | Starten, Laden und Speichern von Versions-Dateien . . . . .   | 10 |
| 3.1 | Verschiedene Anordnungen der beiden Simulatoren . . . . .   | 19 |
| 3.2 | Berechnung von Abbiegewahrscheinlichkeiten anhand der Auslastungen . . . . .                        | 21 |
| 3.3 | Entfernung von Zyklen bei der Pfadplanung . . . . .   | 21 |
| 3.4 | Beispielnetz ohne gesperrte Strecke . . . . .   | 24 |
| 3.5 | Beispielnetz mit gesperrter Strecke, nach neuer Routenplanung von SUMO . . . . .                    | 24 |
| 3.6 | Beispielnetz mit gesperrter Strecke, nach erneuter Umlegung . . . . .                               | 25 |
| 3.7 | Beispielnetz ohne Alternativroute und ohne gesperrter Strecke . . . . .                             | 25 |
| 3.8 | Beispielnetz ohne Alternativroute und mit gesperrter Strecke . . . . .                              | 26 |
| 3.9 | Beispielnetz ohne Alternativroute mit gesperrter Strecke, nach erneuter Umlegung . . . . .          | 26 |
| 4.1 | Netz des Beispielszenarios "BezirkeUndKreuzungen" . . . . .   | 35 |
| 4.2 | Nachfrageganglinie mit steigender Auslastung . . . . .  | 36 |
| 4.3 | OD-Matrizen mit unterschiedlichen Auslastungen der Bezirksbeziehungen im Analyseintervall . . . . . | 37 |
| 4.4 | Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (100er OD-Matrix) . . . . .          | 38 |
| 4.5 | Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (300er OD-Matrix) . . . . .          | 39 |

|   |    |
|---|----|
| 4.6 Auswertung der Routenplanung bei unterschiedlichem Verkehrsaufkommen, Variante 1 . . . . .          | 42 |
| 4.7 Auswertung der Routenplanung bei unterschiedlichem Verkehrsaufkommen, Variante 2 . . . . .          | 43 |
| 4.8 Netz des Beispielszenarios "Streckensperrung" . . . . .   | 45 |
| 4.9 Verwendete OD-Matrix für das Beispielszenario "Streckensperrung" . . . . .                          | 46 |
| 4.10 Auswertung der relevanten Strecken bei vorhandener Alternativroute in SUMO . . . . .               | 47 |
| 4.11 Auswertung der Übergangsstrecke zwischen 3 und 4 bei vorhandener Alternativroute in SUMO . . . . . | 48 |
| 4.12 Auswertung der relevanten Strecken ohne Alternativroute in SUMO . . . . .                          | 49 |
| 4.13 Einfluss des Threshholds beim Algorithmus der Kapazitätsbeschränkungen . . . . .                   | 50 |
| 4.14 Unterschiedliche Möglichkeiten, mehrere SUMO-Gebiete anzugeordnen . . . . .                        | 52 |
|   |    |
| 6.1 Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (100er OD-Matrix) . . . . .          | 65 |
| 6.2 Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (300er OD-Matrix) . . . . .          | 65 |
| 6.3 Auswertung der Routenplanung bei Variante 1 (100er OD-Matrix) . . . . .                             | 66 |
| 6.4 Auswertung der Routenplanung bei Variante 2 (100er OD-Matrix) . . . . .                             | 67 |
| 6.5 Auswertung der Routenplanung bei Variante 1 (300er OD-Matrix) . . . . .                             | 68 |
| 6.6 Auswertung der Routenplanung bei Variante 2 (300er OD-Matrix) . . . . .                             | 69 |
| 6.7 Auswertung der Strecken bei einer Sperrung, mit Alternativroute . . . . .                           | 70 |
| 6.8 Auswertung der Strecke -5 bei einer Sperrung, ohne Alternativroute . . . . .                        | 71 |

---

## **Tabellenverzeichnis**

---

|     |  |    |
|-----|--|----|
| 3.1 | Spezifikationen des verwendeten Systems . . . . .                  | 18 |
| 3.2 | Unterschiede beider Varianten der Routenplanung . . . . .          | 22 |
| 3.3 | Mögliche Parameter zur Anpassung des Ablaufs vom Skript . . . . .  | 30 |
| 6.1 | Unterschiede verschiedener Abstraktionsebenen von Simulatoren . .  | 62 |
| 6.2 | Unterschiede zwischen VISUM und SUMO . . . . .                     | 63 |
| 6.3 | Verschiedene Funktionen der COM-Schnittstelle mit Beispielen . . . | 63 |

---

## Literatur

---

- [AKM14] Hassane Abouaïssa, Yoann Kubera und Gildas Morvan. “Dynamic Hybrid Traffic Flow Modeling”. In: *arXiv:1401.6773 [cs]* (Jan. 2014). arXiv: 1401.6773. URL: <http://arxiv.org/abs/1401.6773> (besucht am 17.06.2021).
- [Bur04] Wilco Burghout. *Hybrid microscopic-mesoscopic traffic simulation*. en. OCLC: 186617287. Stockholm: Dept. of Infrastructure, Royal Institute of Technology, 2004. ISBN: 978-91-7323-099-5.
- [Cao+21] Yumin Cao u. a. “Day-to-day dynamic origin–destination flow estimation using connected vehicle trajectories and automatic vehicle identification data”. en. In: *Transportation Research Part C: Emerging Technologies* 129 (Aug. 2021), S. 103241. ISSN: 0968-090X. DOI: 10.1016/j.trc.2021.103241. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X21002540> (besucht am 03.07.2021).
- [EH+04] Mohamed Said El Hmam u. a. “Towards an Hybrid Simulation Approach of Transportation Systems”. en. In: *IFAC Proceedings Volumes*. 4th IFAC Workshop DECOM-TT 2004: Automatic Systems for Building the Infrastructure in Developing Countries, Bansko, Bulgaria, October 3-5, 2004 37.19 (Okt. 2004), S. 75–80. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)30662-6. URL: <https://www.sciencedirect.com/science/article/pii/S1474667017306626> (besucht am 17.06.2021).
- [Erl21] Stadt Erlangen. *Verkehrszählungen Erlangen*. de. Publisher: Stadt Erlangen. Juli 2021. URL: [https://www.erlangen.de/desktopdefault.aspx/tabid-1446/110\\_read-23676/](https://www.erlangen.de/desktopdefault.aspx/tabid-1446/110_read-23676/) (besucht am 03.07.2021).
- [Fah] *Fahrleistung der Pkw in privaten Haushalten in Deutschland*. de. 2021. URL: <https://de.statista.com/statistik/daten/studie/484040/umfrage/fahrleistung-pkw-in-privaten-haushalten-in-deutschland/> (besucht am 17.06.2021).

- [Fah+07] Ludwig Fahrmeir u. a. "Statistik: Der Weg zur Datenanalyse". de. In: 6. Aufl. Springer-Lehrbuch. Berlin Heidelberg: Springer-Verlag, 2007, S. 280, 322. ISBN: 978-3-540-69739-8. DOI: 10.1007/978-3-540-69739-8. URL: <https://www.springer.com/de/book/9783540697398> (besucht am 17.06.2021).
- [Gaw98] C. Gawron. "An Iterative Algorithm to Determine the Dynamic User Equilibrium in a Traffic Simulation Model". In: *International Journal of Modern Physics C* 09.03 (Mai 1998). Publisher: World Scientific Publishing Co., S. 393–407. ISSN: 0129-1831. DOI: 10.1142/S0129183198000303. URL: <https://www.worldscientific.com/doi/abs/10.1142/S0129183198000303> (besucht am 17.06.2021).
- [Hel+02] D. Helbing u. a. "Micro- and macro-simulation of freeway traffic". en. In: *Mathematical and Computer Modelling* 35.5 (März 2002), S. 517–547. ISSN: 0895-7177. DOI: 10.1016/S0895-7177(02)80019-X. URL: <https://www.sciencedirect.com/science/article/pii/S089571770280019X> (besucht am 17.06.2021).
- [Lop+18] Pablo Alvarez Lopez u. a. "Microscopic Traffic Simulation using SUMO". en. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI: IEEE, Nov. 2018, S. 2575–2582. ISBN: 978-1-72810-321-1 978-1-72810-323-5. DOI: 10.1109/ITSC.2018.8569938. URL: <https://ieeexplore.ieee.org/document/8569938/> (besucht am 17.06.2021).
- [Ma+11] Jian Ma u. a. "A study on multi-resolution scheme of macroscopic-microscopic traffic simulation model". In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. ISSN: 2153-0017. Okt. 2011, S. 1421–1426. DOI: 10.1109/ITSC.2011.6083058.
- [Nü21] Stadt Nürnberg. *Verkehrszählungen - Verkehrsplanung Nürnberg*. 2021. URL: <https://www.nuernberg.de/internet/verkehrsplanung/verkehrzaehlungen.html> (besucht am 03.07.2021).
- [Pfa13] Peter Pfaffelhuber. "Stochastik für Studierende der Informatik". de. In: Sep. 2013, S. 26. URL: <https://www.stochastik.uni-freiburg.de/professoren/pfaffelhuber/inhalte/StochInf2013> (besucht am 17.06.2021).
- [PJL02] K. Pawlikowski, H.-D.J. Jeong und J.-S.R. Lee. "On credibility of simulation studies of telecommunication networks". In: *IEEE Communications Magazine* 40.1 (Jan. 2002). Conference Name: IEEE Communications Magazine, S. 132–139. ISSN: 1558-1896. DOI: 10.1109/35.978060.

- [RNS20] Mojtaba Rostami Nasab und Yousef Shafahi. "Estimation of origin–destination matrices using link counts and partial path data". en. In: *Transportation* 47.6 (Dez. 2020), S. 2923–2950. ISSN: 1572-9435. DOI: 10.1007/s11116-019-09999-1. URL: <https://doi.org/10.1007/s11116-019-09999-1> (besucht am 03.07.2021).
- [Ros] Plaza de Rosalia. "Cappel+Kranzhoff Stadtentwicklung und Planung GmbH Poststraße 27 21709 Himmelpforten". de. In: (), S. 17.
- [Sch] Alexander Schraa. "Erstellen, Modifizieren und Auswerten von Szenarien mit dem Verkehrssimulator VISUM". de. In: (), S. 54.
- [Sum] *About Eclipse SUMO*. en. URL: <https://www.eclipse.org/sumo/about/> (besucht am 17.06.2021).
- [TAH03] Chris Tampère, B. Arem und Serge Hoogendoorn. "Gas-Kinetic Traffic Flow Modeling Including Continuous Driver Behavior Models". In: *Transportation Research Record: Journal of the Transportation Research Board* 1852 (Jan. 2003), S. 231–238. DOI: 10.3141/1852-28.
- [Tay+18] Simon J. E. Taylor u. a. "CRISIS, WHAT CRISIS – DOES REPRODUCIBILITY IN MODELING SIMULATION REALLY MATTER?" In: *2018 Winter Simulation Conference (WSC)*. ISSN: 1558-4305. Dez. 2018, S. 749–762. DOI: 10.1109/WSC.2018.8632232.
- [Tra] *Python: package traci*. URL: <https://sumo.dlr.de/pydoc/traci.html> (besucht am 03.07.2021).
- [Vis] *Unternehmensprofil | PTV Group*. URL: <https://company.ptvgroup.com/de/ueber-ptv-group/unternehmensprofil> (besucht am 17.06.2021).
- [WK+15] Femke van Wageningen-Kessels u. a. "Genealogy of traffic flow models". en. In: *EURO Journal on Transportation and Logistics* 4.4 (Dez. 2015), S. 445–473. ISSN: 2192-4376. DOI: 10.1007/s13676-014-0045-5. URL: <https://www.sciencedirect.com/science/article/pii/S2192437620301114> (besucht am 17.06.2021).
- [YZ09] Fan Ye und Yunlong Zhang. "Vehicle Type-Specific Headway Analysis Using Freeway Traffic Data". en. In: *Transportation Research Record: Journal of the Transportation Research Board* 2124.1 (Jan. 2009), S. 222–230. ISSN: 0361-1981, 2169-4052. DOI: 10.3141/2124-22. URL: <http://journals.sagepub.com/doi/10.3141/2124-22> (besucht am 01.07.2021).

---

## Kapitel 6

---

## Anhang

---

### Abstraktionsebenen von (Verkehrs-)Simulatoren

|                          | mikro                                   | meso  | makro                        |
|--------------------------|---|---|------------------------------|
| Objekte                  | Fahrzeuge                               | kleine Fahrzeuggruppen  | Verkehrsflüsse               |
| Umfang                   | Kreuzungen und kleine Teilnetze         | Stadtbezirke oder mittelgroße Teilnetze                                       | Ganze Städte oder Autobahnen |
| Attribute                | Geschwindigkeit, Beschleunigung, Routen | Aggregierte Werte, wie Ø Geschwindigkeit, Fahrzeuge pro Stunde oder Kilometer | wie bei meso                 |
| Beispiele<br>Simulatoren | SUMO, VISSIM,<br>MITSIMLab              | Mezzo   | VISUM                        |

Tabelle 6.1 – Unterschiede verschiedener Abstraktionsebenen von Simulatoren

## Unterschiede zwischen VISUM und SUMO

|                         | VISUM                                 | SUMO   |
|-------------------------|---------------------------------------|--|
| Firma                   | PTV Group                             | Open-Source  |
| Abstraktionsebene       | makro                                 | mikro  |
| Schnittstelle           | COM für Python                        | Traci für Python/C++   |
| Verwendete Dateiformate | .ver für Szenarien,<br>.net für Netze | .sumocfg für Szenarien,<br>.net.xml für Netze,<br>.add.xml für zusätzliche Objekte |

Tabelle 6.2 – Unterschiede zwischen VISUM und SUMO

## Funktionen der COM-Schnittstelle und ihre Nutzung in Python

| Funktion                     | Beispiel  |
|------------------------------|---|
| Starten des Simulators       | visum = win32com.client.Dispatch("Visum.Visum")   |
| Laden eines Szenarios        | visum.LoadVersion("C:/Visum/BezirkeUndKreuzungen.ver")  |
| Speichern eines Szenarios    | visum.SaveVersion("C:/Visum/BezirkeUndKreuzungen")  |
| Abfrage eines Objekts        | visum.Net.Nodes.ItemByKey(1)<br>visum.Net.Links.ItemByKey(1, 2)<br>visum.Net.Turns.ItemByKey(1, 2, 3) |
| Abfrage einer Collection     | visum.Net.Turns.FilteredBy("[FromNodeNo] != [ToNodeNo]")  |
| Auslesen eines Attributwerts | link1.AttValue("VOLVEHPRT(000000)")   |
| Setzen eines Attributwerts   | inLinks.SetAllAttValues("NAME", "inGoingLinks")   |
| Auslesen einer Matrixzelle   | matrix1.GetValue(1, 2)  |
| Setzen einer Matrixzelle     | matrix1.SetValue(1, 2, 100)   |

Tabelle 6.3 – Verschiedene Funktionen der COM-Schnittstelle mit Beispielen

## Umwandlung von VISUM- in SUMO-Netze

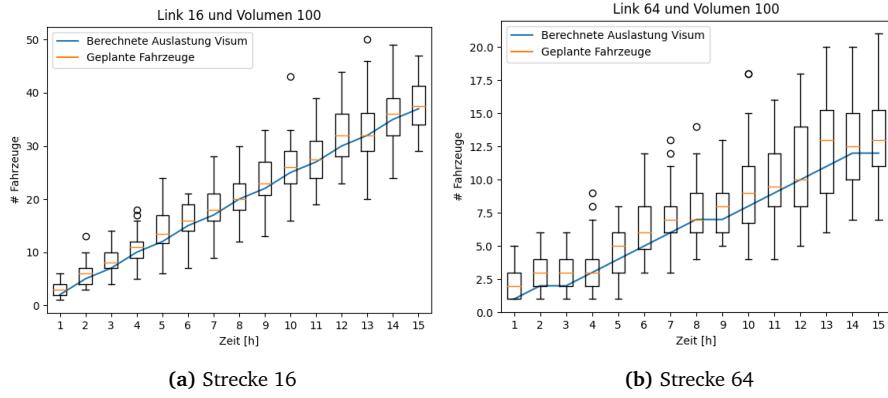
Voraussetzung: VISUM-Netz liegt in .ver-Format vor

1. Netz als .net-Datei aus VISUM exportieren
  - Datei → Datei speichern unter → Netz
  - Namen wählen
  - Bei der Attributsauswahl mindestens die folgenden Attribute markieren:  
Knoten (Nummer, X-Koordinate, Y-Koordinate) und  
Strecken (Nummer, Von-Knoten-Nummer, Nach-Knoten-Nummer, Typ-  
Nummer, VSys-Set, v0 IV)
2. Begriffe in .net-Datei anpassen (Großschreibung nur bei Tabellennamen und Sprache relevant)
  - Tabelle Versionsblock (Z.8): "DEU" statt "ENG"
  - Tabelle Knoten (Z.13): komplett übersetzen  
(z.B. "\$NODE:NO;XCOORD;YCOORD" zu "\$KNOTEN:Nr;XKoord;YKoord"  
ändern)
  - Tabelle Strecken: komplett übersetzen  
(z.B. "\$LINK:NO;FROMNODENO;TONODENO;TYPENO;TSYSSET;CAPPRT;VOPRT"  
zu "\$STRECKE:Nr;VonKnot;NachKnot;Typ;VSysSet;KapIV;v0-IV" ändern)
3. Konvertierung in einer Shell (z.B. CMD) aufrufen:  

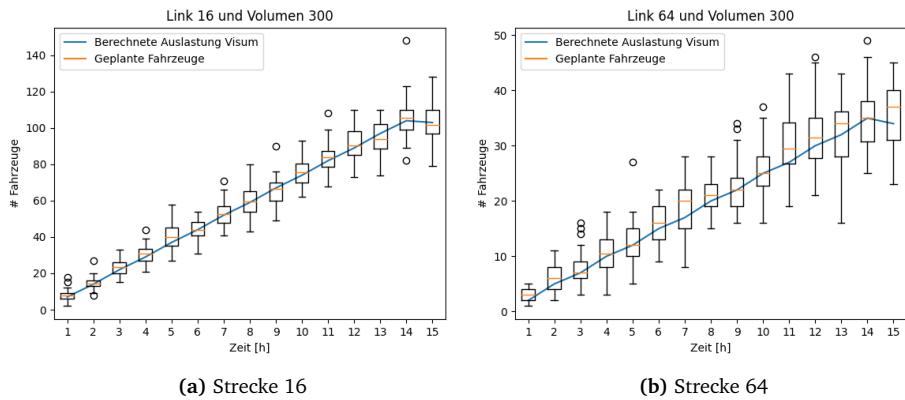
```
netconvert -visum nameInputFile.net -o nameOutputFile.net.xml
```
4. nameOutputFile.net.xml kann nun in netedit bearbeitet werden. Für einen Aufruf aus dem Skript muss es zunächst in der SUMO GUI geöffnet und über File → Save Configuration als .sumocfg-Datei gespeichert werden. Diese Simulationskonfiguration kann anschließend vom Skript gestartet werden.

## Auswertungen der Streckenauslastungen bei der Fahrzeugerzeugung

Im Folgenden sind die Diagramme zu den in Abschnitt 4.1.1.1 weggelassenen Übergangsstrecken dargestellt. Die Graphen der Strecken 32, 34, 36 und 66 sind nicht relevant, da sie im kompletten Analyseintervall keine Auslastung zeigen.



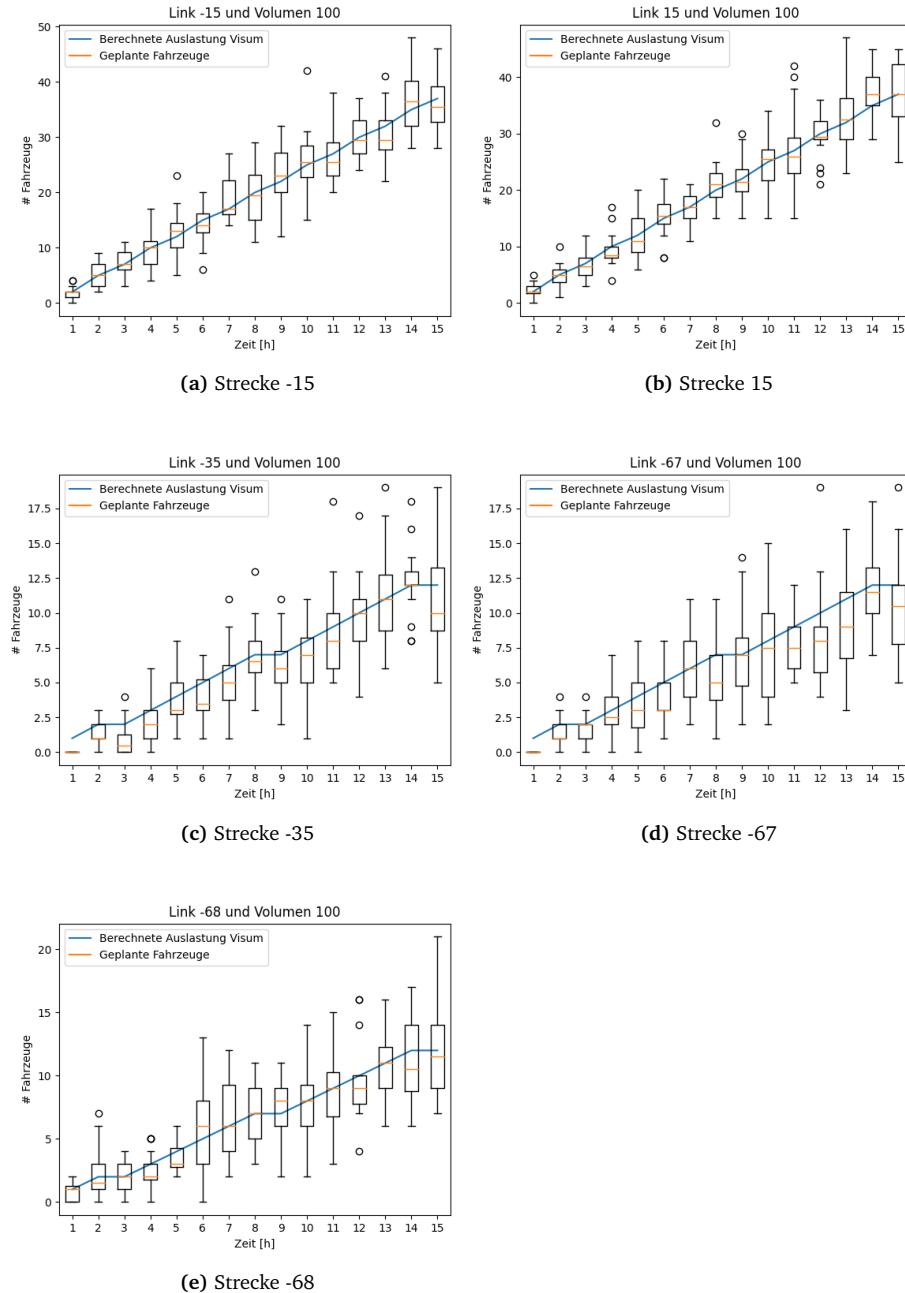
**Abbildung 6.1 – Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (100er OD-Matrix)**



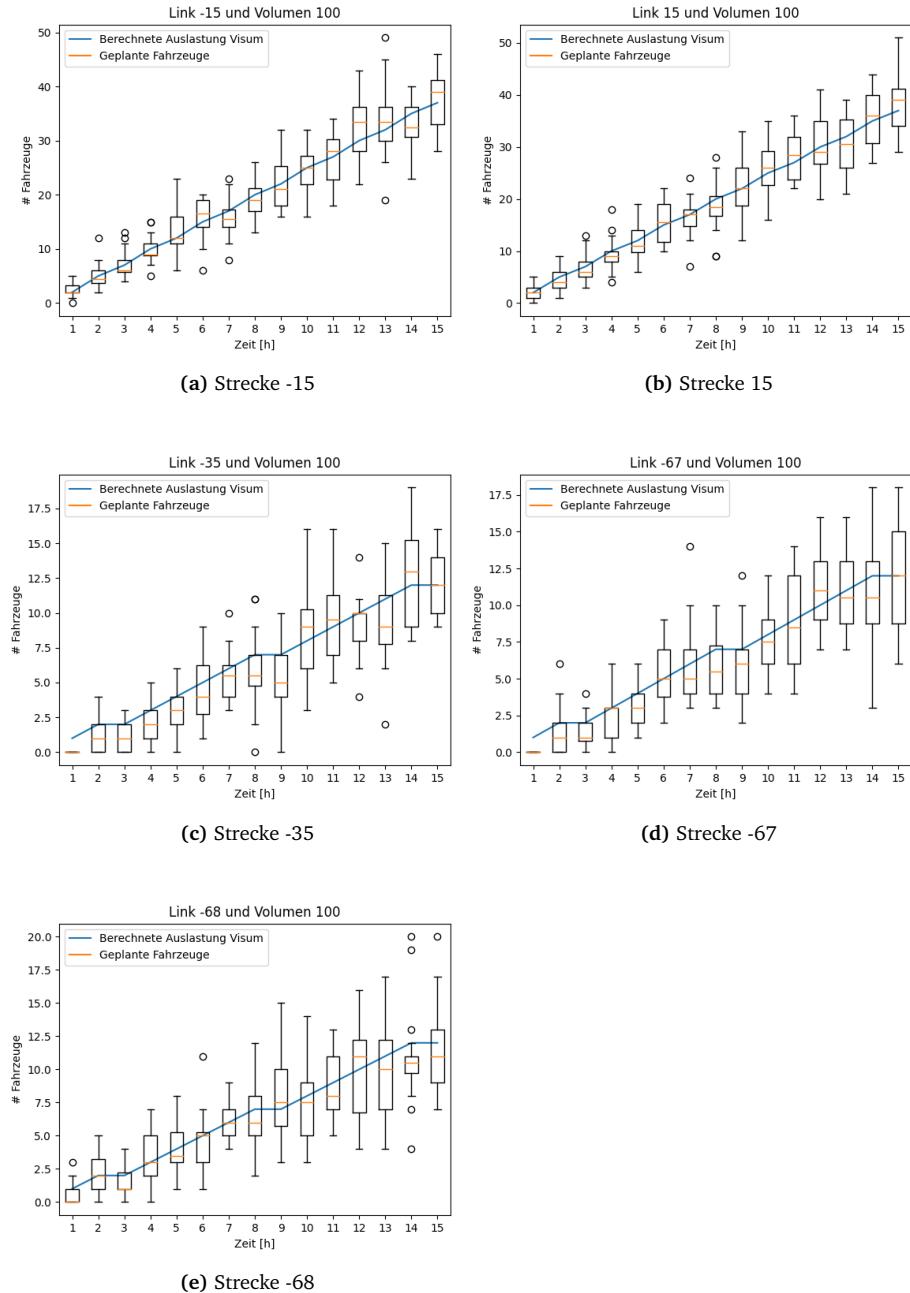
**Abbildung 6.2 – Auswertung der Fahrzeugerzeugung mittels Exponentialverteilung (300er OD-Matrix)**

## Auswertungen der Streckenauslastungen bei der Routenplanung

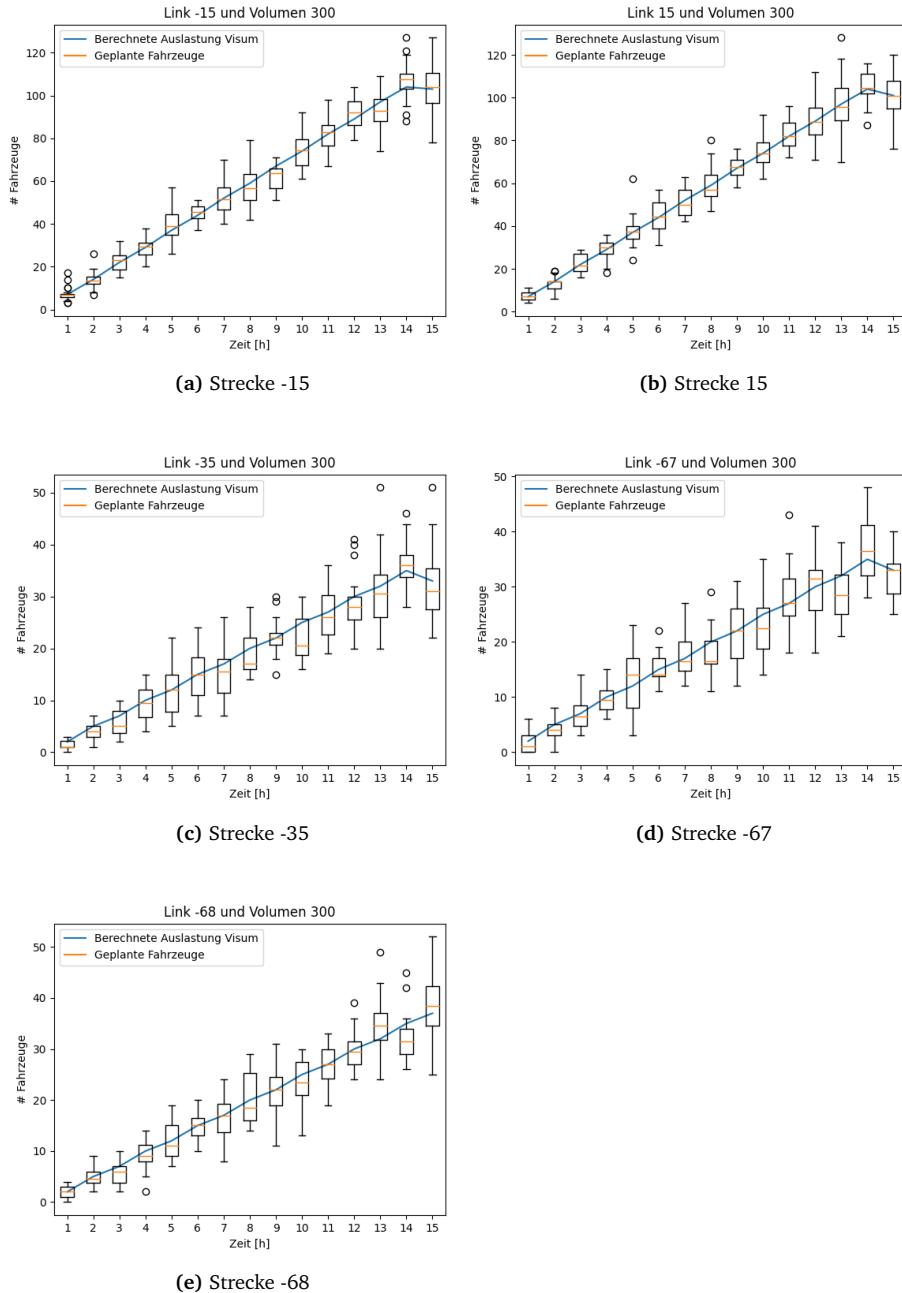
Auf den folgenden Seiten sind die Diagramme zu den in Abschnitt 4.1.1.2 nicht gezeigten Strecken dargestellt. Die Graphen der Straßen 35 und 67 haben keine Auslastung und werden daher auch an dieser Stelle weggelassen. Es wird dabei auf jeder der nächsten Seiten eine der Kombinationen aus 100er beziehungsweise 300er OD-Matrix und entweder Variante 1 oder 2 der Routenplanung dargestellt.



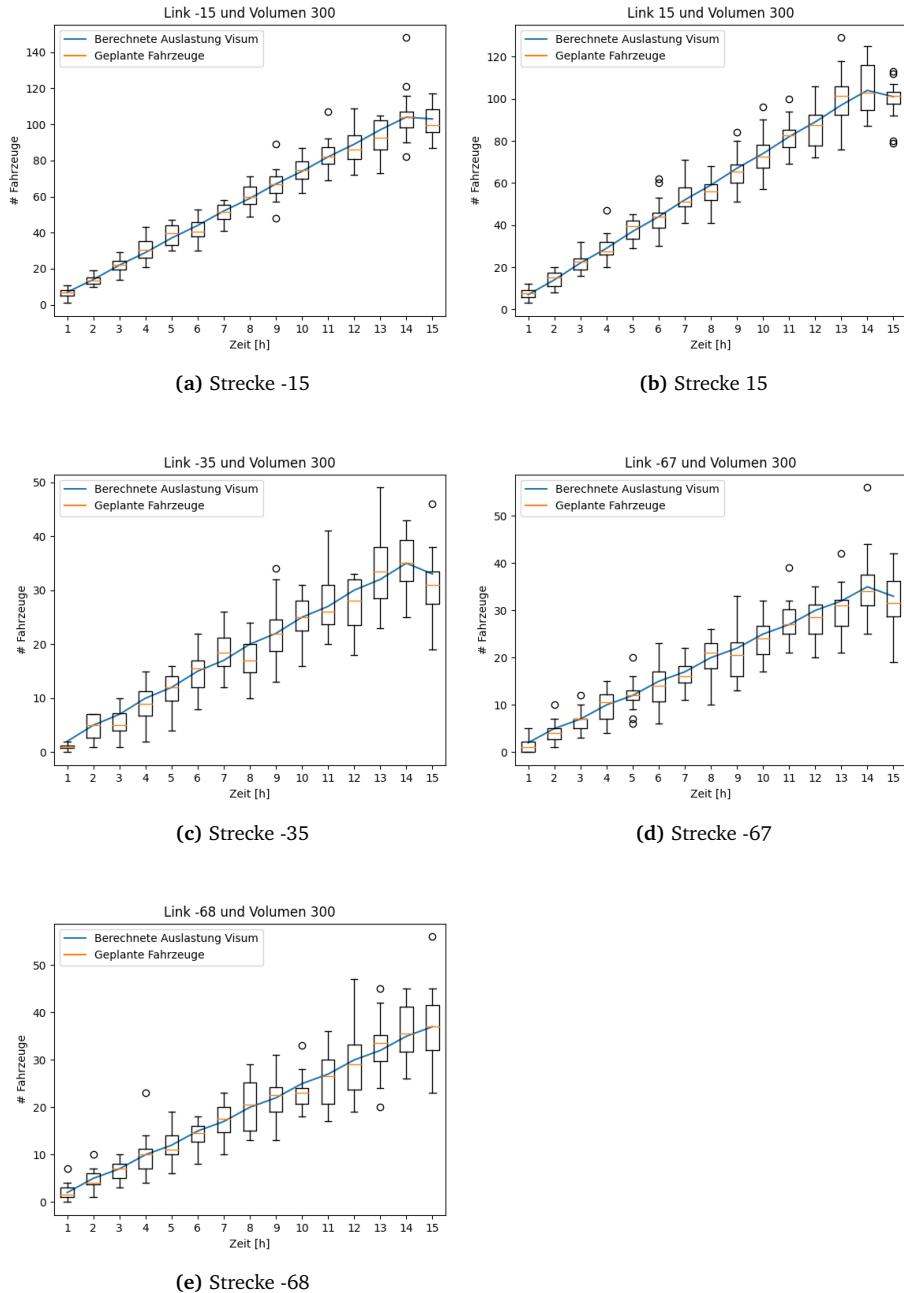
**Abbildung 6.3 – Auswertung der Routenplanung bei Variante 1 (100er OD-Matrix)**



**Abbildung 6.4 – Auswertung der Routenplanung bei Variante 2 (100er OD-Matrix)**



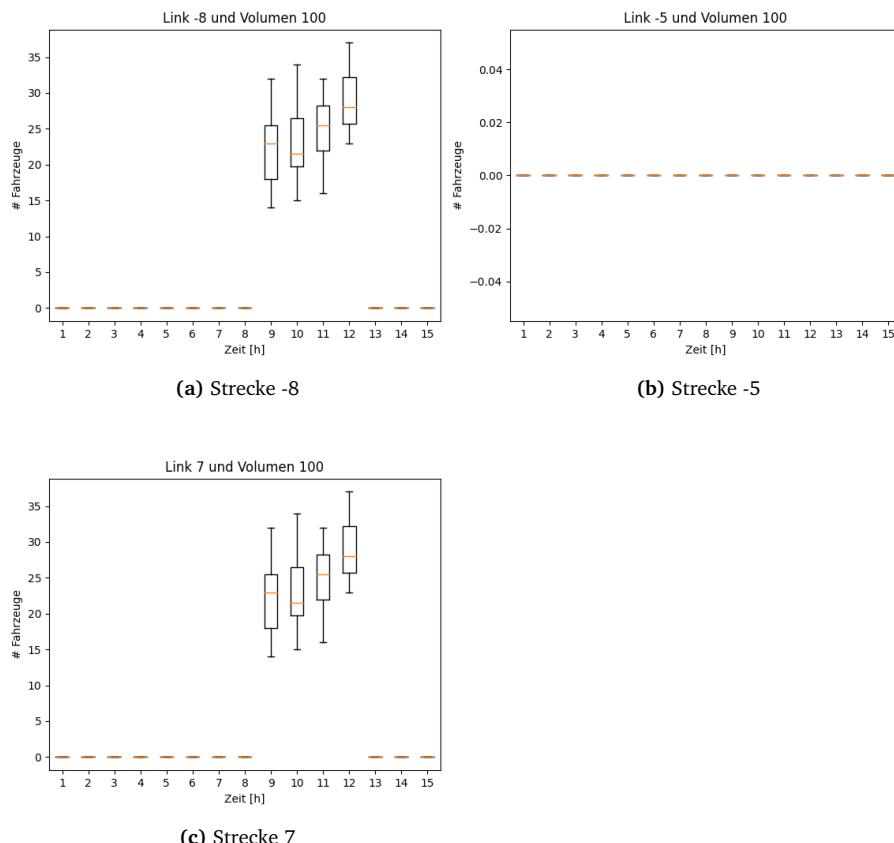
**Abbildung 6.5 – Auswertung der Routenplanung bei Variante 1 (300er OD-Matrix)**



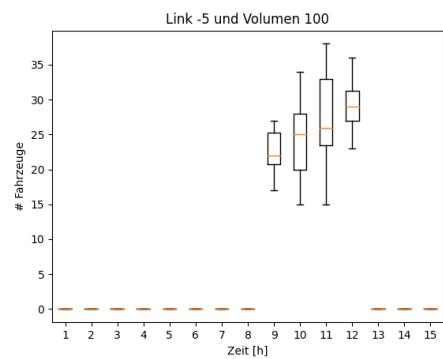
**Abbildung 6.6 – Auswertung der Routenplanung bei Variante 2 (300er OD-Matrix)**

# Auswertungen des Verhaltens bei einer Streckensper- rung

In Abbildung 6.7 sind die Diagramme der Auslastungen der restlichen Strecken gezeigt, die in Abschnitt 4.1.2 weggelassen wurden. Abbildung 6.8 zeigt anschließend die Auslastungen, wenn die obere Alternativroute für Verkehr gesperrt wird und es zur Festlegung von Kapazitätsbeschränkungen und einer neuen Umlegung kommt.



**Abbildung 6.7** – Auswertung der Strecken bei einer Sperrung, mit Alternativroute



**Abbildung 6.8** – Auswertung der Strecke -5 bei einer Sperrung, ohne Alternativeroute