

Master Abschlussarbeit

**Dezentrale dynamische Verkehrslenkung im
V2X Szenario**

Tim Bierwirth
Dezember 2018

Gutachter:

Prof. Dr. Katharina Morik

Dr. Thomas Liebig

Technische Universität Dortmund

Fakultät für Informatik

Lehrstuhl für Künstliche Intelligenz

<http://www-ai.cs.uni-dortmund.de/>

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufbau der Arbeit	3
2	Grundlagen und Stand der Forschung	5
2.1	Verkehr	5
2.1.1	Verkehrslenkung und autonomes Fahren	5
2.1.2	V2X Szenario	6
2.2	Verkehrssensoren	7
2.2.1	Messgrößen	7
2.3	Verkehrsfluss	7
2.3.1	Fundamentaldiagramm	9
2.3.2	Makroskopisches Verkehrsfluss Modell	10
2.3.3	Breakdown und kritischer Fluss	11
2.3.4	Bottleneck Problem	11
2.4	Straßennetzmodellierung	12
2.4.1	Kürzeste Routen im Straßennetz mit Dijkstra Algorithmus	13
2.4.2	Dijkstra Rang	14
2.5	Bestärkendes Lernen (Reinforcement Learning)	14
2.5.1	Learning from Bandit Feedback	15
2.5.2	Policy Optimizer for Exponential Models (POEM)	16
2.6	Breakdown Minimization Principle (BMP)	16
2.7	Label Proportionen	17
2.7.1	Definition <i>Learning from label proportions</i> aus [SM11]	18
2.8	Verwandte Arbeiten	18
3	Methodik	21
3.1	Dynamische Verkehrslenkung	21
3.1.1	Dezentrale Verkehrsdaten	22
3.1.2	Feedback der Verkehrsumgebung durch Überfüllungsetiketten	22

3.1.3	Gewichteter Überfüllungsgraph	25
3.1.4	Routenberechnung	25
3.2	Verkehrslenkung mit POEM	27
3.2.1	POEM Voraussetzungen	27
3.2.2	POEM Anwendung	28
3.2.3	Logging von lokalen Features	29
3.2.4	Datensatz mit lokalen Features für bestärkendes Lernen	31
3.3	LLP Vorverarbeitung	31
3.4	LLP Trainingssetting	32
3.4.1	Anwendung von LLP als Feature in Verkehrslenkung mit POEM	34
3.5	Verkehrslenkung mit LLP Labelvorhersage	35
4	Experimente und Evaluation	39
4.1	Experimentenumgebung	39
4.1.1	Datensatz	39
4.2	Evaluationsmetriken	40
4.2.1	Kritischer Fluss Map für LuST	43
4.3	Verkehrslenkungsexperiment mit lokalem RL	45
4.3.1	Vergleich zu Verkehrslenkung auf Basis zentraler Verkehrsdaten	50
4.4	Verkehrslenkungsexperiment mit LLP als RL Feature	54
4.4.1	Generierung der Eingabedaten für den LLP Algorithmus	54
4.4.2	Evaluation der Ergebnisse mit LLP als POEM Feature	54
4.5	Verkehrslenkungsexperiment mit LLP	59
5	Zusammenfassung	63
	Abbildungsverzeichnis	67
	Literaturverzeichnis	69
	Erklärung	75

Kapitel 1

Einleitung

In dieser Arbeit werden Möglichkeiten gezeigt, durch dynamische Verkehrslenkung Einfluss auf den Straßenverkehr zu haben. Im Mittelpunkt steht dabei die Nutzung der Sensordaten in der lokalen Umgebung von Fahrzeugen. Mit den Daten werden durch den Einsatz von lernenden Algorithmen Überfüllungen von Straßen erkannt. Die Verkehrslenkung reagiert auf erkannte Überfüllungen durch das Anpassen der Route eines Fahrzeugs. Es stellt sich dabei die Frage, wie gut eine dezentrale Verkehrslenkung den Verkehrsfluss verbessert. Diese Frage wird anhand der Anwendung in der Simulation der Stadt Luxemburg evaluiert.

1.1 Motivation

Der Straßenverkehr in Deutschland entwickelt sich durch die Bewegung der Menschen in Großstädte zu einem Zustand von überfüllten Straßen, der durch das Verhalten der Fahrer beeinflusst wird. Weltweit kommt es daher gerade in größeren Städten zu einem sehr dichten Straßenverkehr [Tom18]. Dabei nutzen 80% des Personen- und des Güterverkehrs den Weg des Straßenverkehrs [Bun18a] [Bun18b]. Das Problem der Überfüllung von Straßennetzen könnte durch die Erhöhung der Kapazität der Straßennetze verringert werden, jedoch stellt diese Gegenmaßnahme eine der teuersten Lösungen dar [AT10].

In der Forschung beschäftigt man sich immer mehr mit der Minimierung von Fahrzeiten im Straßenverkehr. Dabei wird der auftretende Verkehrsfluss häufig in den Mittelpunkt gestellt, um die Bewegung der Fahrzeuge zu ihrem Ziel nicht zu verlangsamen. Es kann dabei immer zu Staus im Straßenverkehr kommen. Die höhere Nutzung der Straßen hat ebenfalls mehr Baustellen als Ergebnis, die umfahren werden müssen. Prognose ist eine Verschlimmerung bis 2030.

Ein allgemeiner Ansatz zur Verringerung von Überfüllungen im Straßenverkehr sieht eine Verkehrsverlagerung vor [AT10]. Die Anpassung von Fahrzeugrouten durch die Nutzung von wenig befahrenen Straßen ermöglicht eine bessere Verteilung der Fahrzeuge im Straßennetz. Dabei wird ein Weg durch ein Straßennetz gesucht, wie bei einem Fahrzeug

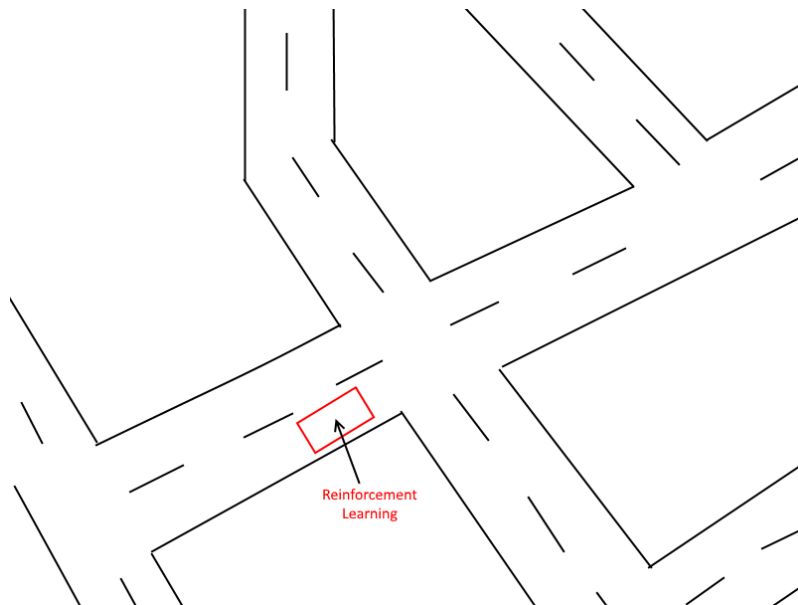


Abbildung 1.1: Ein Fahrzeug befindet sich in einer Verkehrsumgebung und erhält Informationen über umliegende Überfüllungen von Straßen, damit durch eine Verkehrslenkung die Route zum Ziel bestimmt wird.

in Abbildung 1.1. Es gibt auch noch weitere Möglichkeiten Überfüllungen zu verringern. Beispielsweise durch die Weiterentwicklung des Nahverkehrs kann der Verkehr verlagert werden [Bot03] [Hua09]. Weitere Alternativen liegen ebenfalls in der Nutzung umweltfreundlicher Möglichkeiten wie Fahrradfahren. Auch Car Sharing stellt eine Möglichkeit dar [Kat03].

In Abbildung 1.2 wird ein Szenario dargestellt, in dem sich dezentrale dynamische Verkehrslenkung erläutern lässt. In einer Verkehrsumgebung befinden sich Fahrzeuge, die möglichst schnell auf Straßen fahren, um an einen Zielpunkt zu gelangen. Um dies zu ermöglichen kann jedes Fahrzeug im einfachsten Fall den kürzesten Weg zum Ziel abfahren, ohne den Verkehr zu betrachten. Es kann jedoch dazu kommen, dass dabei überfüllte Straßen genutzt werden und somit längere Fahrzeiten entstehen und die Überfüllung auf Straßen durch weitere Fahrzeuge erhöht wird. Daher lohnt sich der Einsatz von dynamischer Verkehrslenkung, damit schnell auf Überfüllungen reagiert werden kann und sich der Verkehr besser in der Verkehrsumgebung verteilt.

Viele intelligente Verkehrslenkungen basieren auf unveränderten Verkehrsdaten, die den aktuellen Zustand einer Verkehrsumgebung darstellen. In dieser Arbeit werden Voraussagen über den Verkehr genutzt, sodass aktuelle Verkehrsinformationen durch zukünftige Verkehrsinformationen erweitert werden. Dies ermöglicht einen wichtigen Nutzen für eine Verkehrslenkung, da entstehende überfüllte Straßen frühzeitig erkannt werden und die Möglichkeit besteht, diese zu umfahren. Das Besondere liegt darin, dass Fahrzeuge durch die Sensoren bereits verarbeitete Informationen über zukünftige Zustände erhalten. Das

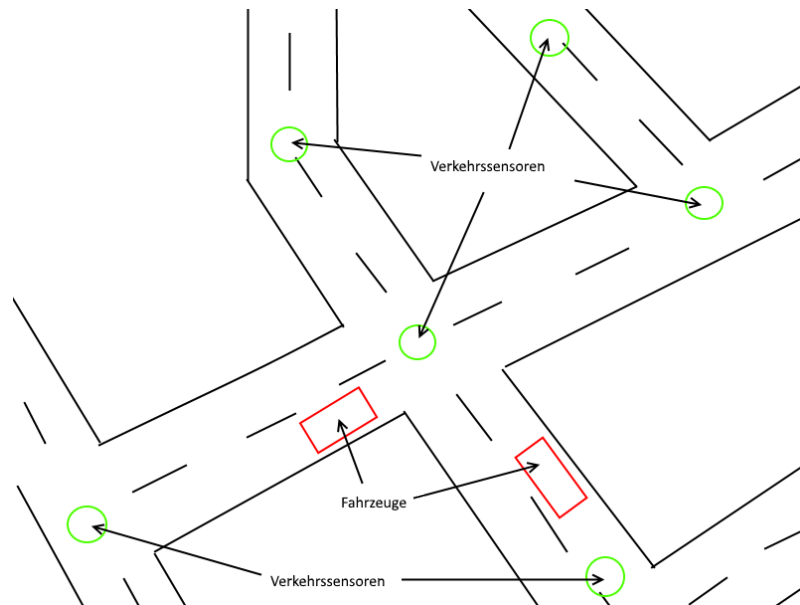


Abbildung 1.2: Beispielhafte Skizze einer Verkehrsumgebung, die mit Verkehrssensoren ausgestattet ist.

Verarbeiten der erweiterten Verkehrsdaten in Kombination mit einer intelligenten Verkehrslenkung resultiert in einem großen Vorteil und hebt sich von vielen Verkehrslenkungsmethoden ab. Das Schwierige an diesem Konzept liegt darin, dass Fahrzeuge zu Beginn ihrer Fahrt noch nicht viele Informationen über die Verkehrsumgebung besitzen, da diese auf der Fahrt in einem individuellen Datensatz gesammelt werden.

1.2 Aufbau der Arbeit

Das zweite Kapitel dieser Arbeit behandelt die thematischen Grundlagen des Verkehrsflusses und der Verkehrslenkung. Ebenfalls werden in diesem Kapitel Algorithmen vorgestellt, die eingesetzt werden können, um einen Einfluss auf die Verkehrslenkung zu haben. In diesem Zusammenhang werden verwandte Arbeiten vorgestellt, die sich mit Verkehrsflüssen und Verkehrslenkung befassen. Im dritten Kapitel werden die methodischen Inhalte vorgestellt, die umgesetzt wurden, um eine dynamische dezentrale Verkehrslenkung zu ermöglichen. Insbesondere werden hier die Einsatzgebiete des bestärkenden Lernens und des Lernens durch Labelproportionen gezeigt. Auf die Methodik folgen im vierten Kapitel die Experimente und deren Evaluierung. Es wird gezeigt, welche Simulationsumgebung benutzt wird sowie herausgestellt, was mit den methodischen Inhalten möglich ist. Abschließend wird in Kapitel fünf eine Zusammenfassung der Ergebnisse dargestellt und mögliche offene Fragen diskutiert.

Kapitel 2

Grundlagen und Stand der Forschung

Dieses Kapitel beschreibt die Grundlagen, die für intelligente Verkehrslenkung nötig sind. Dabei geht es zunächst um Verkehrssensoren und Verkehrsflüsse. Des Weiteren werden grundlegende Konzepte erläutert, die für eine Verkehrslenkung nötig sind. Anschließend werden lernende Algorithmen gezeigt, die die Verkehrslenkung beeinflussen werden.

2.1 Verkehr

Die Verkehrsforschung beschäftigt sich mit Fahrzeugen im Straßenverkehr, die auf bestimmten Straßen in Städten fahren können und bestimmte Routen zurücklegen, um zu ihrem Ziel zu kommen [LPBM14]. Auf dem Weg kommt es zu Interaktionen zwischen Verkehrsteilnehmern, die zu Problemen in Form von Aufstauung und Überfüllung von Straßen führen können. Die Probleme haben zur Folge, dass Fahrzeuge auf Straßen mit reduzierter Geschwindigkeit fahren müssen, sodass die Fahrzeit erhöht wird.

2.1.1 Verkehrslenkung und autonomes Fahren

Am Straßenverkehr teilnehmende Fahrzeuge werden von den Fahrern gefahren und diese greifen in der Regel auf die Kenntnis einer Route zurück, um ein Ziel zu erreichen. Routen, die ein Fahrer wählt, können jedoch eine sehr große Fahrzeit in Anspruch nehmen, da dem Fahrer jegliche Informationen über den aktuellen Verkehr fehlt. Um einen Einfluss auf die Auswahl der genutzten Straßen des Fahrzeugs zu erhalten, gibt es mehrere Möglichkeiten. Es gibt die Verkehrskontrolle mit externer Kommunikation, wobei beispielsweise durch Ampeln bestimmte Straßen priorisiert werden und der Verkehr auf diese Weise gelenkt wird. Dabei werden alle Fahrzeuge beeinflusst, die mit Ampeln versehene Straßen und Kreuzungen nutzen. Eine weitere Möglichkeit stellt individuelle Kommunikation dar. Hierbei wird zum Beispiel über Verkehrsinformationen im Radio Einfluss auf die Verkehrslenkung der Fahrzeuge genommen, sodass Fahrer Informationen über überfüllte Straßen erhalten und diese meiden können. Diese Informationen sind jedoch nicht für jedes Fahrzeug hilfreich

und es kann zusätzlich zu ungewollten Umwegen führen. Deswegen gibt es die individuelle Verkehrslenkung mit individueller Kommunikation. Diese nutzt aktuelle Verkehrsdaten, um für jedes Fahrzeug Routen zu bestimmen, die die Fahrzeiten verringern. Navigationssysteme sind Beispiele für die Nutzung dieser Möglichkeit zur Verkehrslenkung. In diesem Fall nutzen Fahrer individuell für ihren Standort durch das Navigationssystem berechnete Routen, die ebenfalls durch den aktuellen Verkehr beeinflusst sind. Ein weiteres mögliches Anwendungsgebiet der Verkehrslenkung liegt im autonomen Fahren, da die Route des Fahrzeugs automatisch bestimmt wird und der Fahrer diese nutzt, um an sein Ziel zu gelangen [MGLW15]. Verkehrslenkung ermöglicht eine Fortbewegung von Fahrzeugen, die keinen Fahrer benötigt.

2.1.2 V2X Szenario

Es existieren mehrere Möglichkeiten, wie die Kommunikation im Straßenverkehr stattfindet. Am Straßenverkehr nehmen Fahrzeuge teil, die sich auf Straßen befinden. Zusätzlich gibt es auch Fußgänger, die den Bürgersteig nutzen. Das *Vehicle-To-Infrastructure* (V2X) Szenario betrachtet die Kommunikation von Fahrzeugen mit der Verkehrsumgebung. Hierbei wird die Verkehrsumgebung durch Verkehrssensoren dargestellt, die Informationen über den aktuellen Verkehr aufzeichnen und diese verarbeiten können. Es sind ebenfalls Kommunikationen zwischen den Fahrzeugen (*Vehicle-To-Vehicle* V2V) und zwischen Fußgängern und Fahrzeugen (*Vehicle-To-Pedestrian*) möglich. Abbildung 2.1 zeigt diese möglichen Kommunikationen beispielhaft anhand einer Kreuzung.

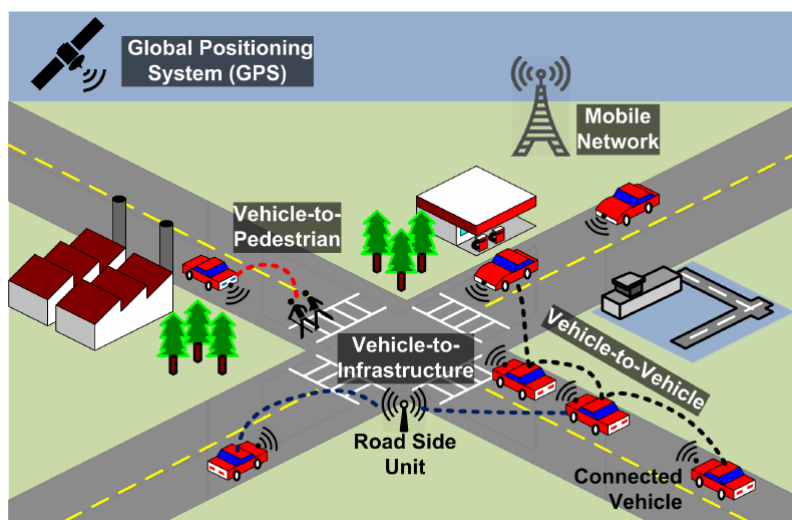


Abbildung 2.1: Verschiedene Kommunikationsmöglichkeiten im Straßenverkehr zwischen Fahrzeugen, Verkehrsumgebung und Fußgängern. Quelle: [HNZ15]

2.2 Verkehrssensoren

Verkehrssensoren können viele verschiedene Daten aufzeichnen und diese der Umgebung zur Verfügung stellen. Es ist ebenfalls möglich, die Daten sogar durch Kommunikation mit benachbarten Verkehrssensoren aufzubereiten. Die Aufzeichnung der Daten kann dabei auf *single-point* Sensoren oder *point-to-point* Sensoren basieren [Sch04].

Bei *single-point* Sensoren handelt es sich um Sensoren, die, wie der Name schon sagt, einen einzelnen Punkt der Straße beschreiben und dadurch nur die Anzahl von Fahrzeugen auf einer Straße aufzeichnen. Es gibt jedoch auch komplexe Sensoren, die Daten über Geschwindigkeiten von Fahrzeugen aufzeichnen können [ORO02]. Es ist ebenfalls möglich, auch die Art und somit die Größe von Fahrzeugen zu erkennen.

Eine höhere Aktualität besitzen *point-to-point* Sensoren. Diese haben das Ziel *source-target* Verteilungen zu erhalten, indem Fahrzeuge an mehreren Punkten auf der Straße überwacht werden. Diese Festdetektoren kommen häufig im Straßenverkehr zum Einsatz. Es werden Zeiten gemessen, die Informationen über die Dichte einer Straße oder die Geschwindigkeit der fahrenden Fahrzeuge betreffen. Es gibt verschiedene Arten von Festdetektoren, wie z.B. Induktionsschleifen, Videokameras, Infrarot-, Radar- oder Ultraschallmessungen [Sch04].

2.2.1 Messgrößen

Es existieren viele verschiedene Messgrößen, die durch Verkehrssensoren aufgezeichnet werden können. Interessant sind Daten über Anzahlen von Fahrzeugen auf Straßen sowie deren Geschwindigkeit. Daraus können auch weitere Informationen über den Verkehr auf einer Straße gewonnen werden, wie die Dichte, die Abstände der Fahrzeuge, Wartezeiten von Fahrzeugen oder die messbare Belegung einer Straße.

Beispielsweise bei Induktionsschleifen, die aus zwei Schleifen bestehen, kann die Geschwindigkeit eines Fahrzeuges sehr einfach herausgefunden werden [Sch04]. Die Geschwindigkeit v_n des n -ten Fahrzeugs lässt sich durch die Zeitpunkte t_{D_1} und t_{D_2} der Detektoren D_1 und D_2 und dem Abstand $d_{D_{1,2}}$ der Detektoren durch

$$v_n = \frac{d_{D_{1,2}}}{t_{D_1} - t_{D_2}} \quad (2.1)$$

bestimmen, wobei implizit angenommen wird, dass sich die Geschwindigkeit im Intervall $[t_{D_1}, t_{D_2}]$ nicht ändert [Sch04].

2.3 Verkehrsfluss

Ein elementarer Bestandteil des Straßenverkehrs ist der Verkehrsfluss. Dieser lässt sich allgemein als Anzahl von Fahrzeugen in einem bestimmten Zeitintervall in einer Verkehrsumgebung beschreiben. Einen großen Einfluss auf den Verkehrsfluss haben die Dichte oder

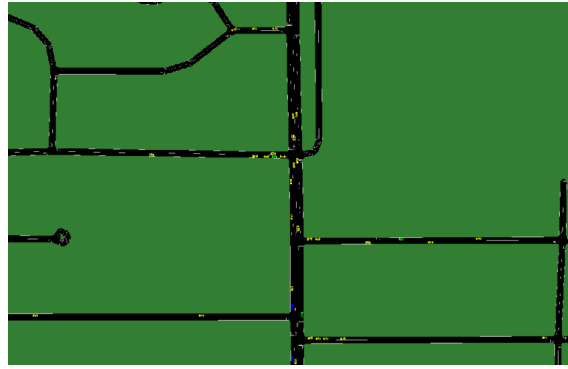


Abbildung 2.2: Beispielhafte Darstellung einer Verkehrsflussimulation. Quelle: [KPF⁺09]

Belegung einer Straße und die Durchschnittsgeschwindigkeiten von Fahrzeugen auf einer Straße. Mit dem Verkehrsfluss lassen sich Aussagen über die hauptsächlich gewählten Routen und Straßen in einer Verkehrsumgebung treffen oder analysieren, ob es zu Problemen in der Verkehrsumgebung kommt.

Um den Verkehrsfluss auf verschiedenen Straßen zu beschreiben, wird ein mathematisches Modell gesucht. Dabei gibt es mikroskopische und makroskopische Ansätze. Der mikroskopische Ansatz bezieht sich auf Interaktionen zwischen den Verkehrsteilnehmern auf einer Straße und die Fahrzeuge sind von einander unterscheidbar. Makroskopische Modelle beschreiben den Zustand einer Straße und geben Straßen bestimmte Labels, um diese Zustände zu beschreiben. Es werden Dichten oder durchschnittliche Geschwindigkeiten betrachtet. Mikroskopische Modelle benötigen einen höheren Rechenaufwand als makroskopische Modelle. Die Wahl fällt je nach Anwendung in den meisten Fällen auf makroskopische Modelle.

Verkehrsflussmodelle basieren auf Verkehrssensoren, die den aktuellen Verkehr in Form von Fahrzeuganzahlen oder Fahrzeuggeschwindigkeiten aufzeichnen. Dabei werden Netze aus statischen Sensoren genutzt. Abbildung 2.2 zeigt Fahrzeuge in einer simulierten Verkehrsumgebung. Simulationen basieren auf einem Mikroskopischen Modell. Diese Fahrzeuge simulieren den Verkehrsfluss auf den Straßen. Ein makroskopisches Modell hingegen kennt nicht wie in Abbildung 2.2 die genauen Positionen und Anzahlen pro Straße, sondern macht Aussagen über Zustände der Straßen. Eine Straße kann dabei beispielsweise den Zustand eines Staus annehmen und würde das Label für einen Stau erhalten. Sollte sich der Stau auflösen, ändert sich das Label der Straße im makroskopischen Modell.

Nach [Hä74] muss ein Verkehrsflussmodell die zwei folgenden Anforderungen erfüllen: Als erste Anforderung muss die Anwendung auf ein reales Szenario möglich sein. Weiterhin muss die Möglichkeit bestehen, zwischen mikroskopischen und makroskopischen Modellen hin und her zu wechseln. Beim Wechsel der Modelle dürfen keine kritischen Informationen verloren gehen. Das vorgestellte Verkehrsfluss Modell erfüllt diese Anforderungen, indem reale mikroskopische Sensordaten genutzt werden.

2.3.1 Fundamentaldiagramm

Um Verkehr aussagekräftig charakterisieren zu können, wird das Fundamentaldiagramm genutzt. Das Fundamentaldiagramm in Abbildung 2.3 wird aus Daten über Dichte und Verkehrsfluss über 5 Minuten aus einer empirischen Messung zusammengesetzt. Es ist ein Freiflussast mit positiver Steigung und ein gestauter Ast mit negativer Steigung erkennbar [Sch04]. Diese beiden Eigenschaften des Fundamentaldiagramms sind noch einmal in Abbildung 2.4 getrennt dargestellt. Es werden durch diese beiden Eigenschaften die zwei Phasen des Fundamentaldiagramms beschrieben. Der Verkehrsfluss steigt also zunächst bei höherer Dichte und befindet sich im freien Fluss, sinkt jedoch, sobald Wechselwirkungen zwischen Fahrzeugen entstehen und beispielsweise der Abstand zwischen Fahrzeugen geringer wird oder die Geschwindigkeit der Fahrzeuge von der Wunschgeschwindigkeit abweicht.

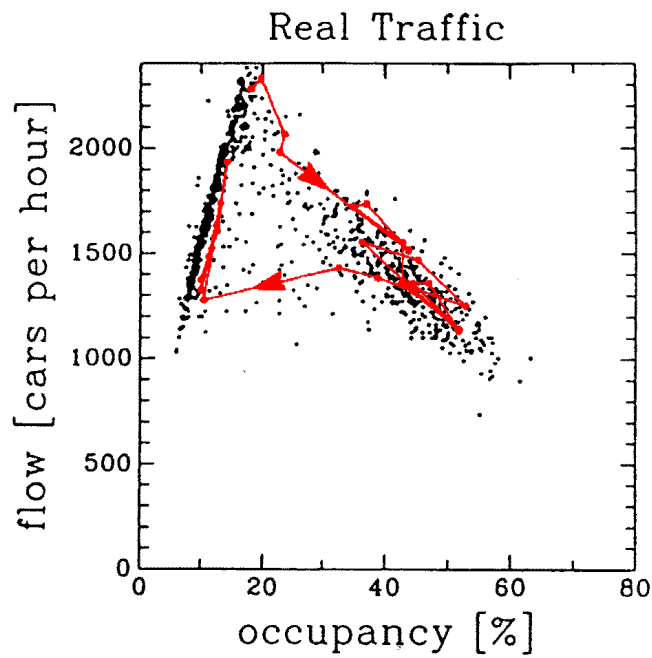


Abbildung 2.3: Fundamentaldiagramm des Verkehrsflusses anhand empirischer Daten. Zusätzlich ist eine Hysterese eingezeichnet. Jeder Punkt entspricht einer Nachricht über 5 Minuten. Quelle: [Sch04]

Die roten Pfeile in Abbildung 2.3 zeigen das Phänomen der Hysterese [Sch04]. Bei einer Hysterese kommt es, nachdem die Ursache geändert wurde, zu einer verzögerten Wechselwirkung. Es wird von einem Zeitpunkt t_1 gestartet, in dem eine bestimmte Dichte und ein bestimmter Verkehrsfluss herrscht. Wird die Dichte erhöht, sinkt der Verkehrsfluss. Bei einer Verringerung der Dichte steigt der Verkehrsfluss nun erneut. Es ist jedoch zu erkennen, dass nicht der Ausgangswert erreicht wird.

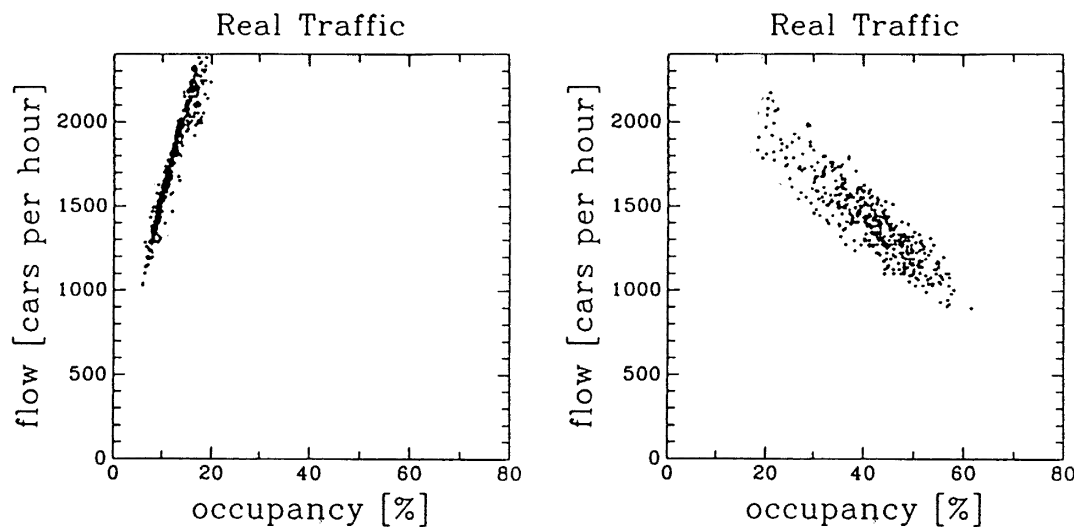


Abbildung 2.4: Freiflussast (links) und gestauter Ast (rechts) des Fundamentaldiagramms. Quelle: [Sch04]

2.3.2 Makroskopisches Verkehrsfluss Modell

Grundsätzlich werden makroskopische Variablen dazu benutzt, um aggregierte Zustände eines Verkehrsnetzes zu repräsentieren [HK13]. Das Makroskopische Verkehrsflussmodell beschreibt Straßen anhand bestimmter Messgrößen wie den zuvor vorgestellten Verkehrsfluss oder die Dichte einer Straße. Der Verkehrsfluss wird dargestellt durch

$$q := \frac{n}{T}. \quad (2.2)$$

Dabei beschreibt n die Anzahl der Fahrzeuge und T das Zeitintervall, in dem die Fahrzeuge durch Verkehrssensoren wie beispielsweise *single-point* Sensoren aufgezeichnet wurden. Eine weitere beschreibende Größe ist die Dichte

$$k := \frac{n}{X} \quad (2.3)$$

mit der Fahrzeuganzahl n auf der Strecke X .

Makroskopisches Bewegungsmodell (Physikalisches Modell)

Das *Physikalische Modell* ist Teil der Makroskopischen Bewegungsmodelle und wird auch als Verkehrsmodell verwendet. Aufgrund der Ähnlichkeit kann das Makroskopische Bewegungsmodell mit einem gaskinetischen Modell verglichen werden. Im Gaskinetischen Modell existieren makroskopische Beobachtungen, die charakteristisch für Bewegung, sind wie Durchschnittsgeschwindigkeit, durchschnittliche Dichte und Zahl von Vorkommen an bestimmten Punkten des Modells [LAA14]. Diese Beobachtungen lassen sich ebenfalls auf ein Verkehrsmodell übertragen. Statt Molekülen, die zu einem bestimmten Zeitpunkt in einem Volumen gezählt werden, werden Fahrzeuge auf Straßen gezählt.

2.3.3 Breakdown und kritischer Fluss

Ein Breakdown in einer Verkehrsumgebung kann auf Straßen oder Kreuzungen auftreten. Damit ist der Zusammenbruch eines fließenden Verkehrs gemeint. Es wird ein Zustand erreicht, bei dem die Kapazität eines freien Verkehrsflusses durch den Breakdown eingeschränkt wird [KKS14]. Weiterhin hängt die Zuverlässigkeit einer Verkehrslenkung davon ab, diese Breakdowns zu vermeiden.

Der kritische Fluss ist ein Wert, der die Entstehung von Breakdowns auf Straßen festlegt. Wenn dieser Wert überschritten wird, entstehen Breakdowns. Das Fundamentaldia-gramm aus Abbildung 2.3 enthält ebenfalls den kritischen Fluss, sobald es zu einem möglichen Zusammenbruch des fließenden Verkehrs kommt. Dieser Breakdown befindet sich im höchsten Punkt des Freiflussastes.

Kritischer Fluss

Der kritische Fluss beschreibt einen Wert für einen Fluss, der den maximalen Fluss einer Straße beschreibt, bevor diese überfüllt ist und es zu einem Stau kommt. Es werden dabei zwei Werte als C_{min} und C_{max} unterschieden. Dabei lassen sich Kreuzungen als Bottleneck bezeichnen, wenn viele Fahrzeugrouten über diese Kreuzung führen und es somit zur Aufstauung des Verkehrs kommen kann. C_{min} bezeichnet einen Wert, der überschritten wird, wenn es zu einem Breakdown vor einer Kreuzung kommen kann [Vra17]. Für einen Wert zwischen C_{min} und C_{max} lassen sich Wahrscheinlichkeiten berechnen, ob es zu einem Breakdown kommt. Ist C_{max} überschritten, wird es einen Breakdown an der Kreuzung geben. Dieser kritische Fluss C_{max} berechnet also ein Label für Überfüllung an einer bestimmten Kreuzung, welches dazu geeignet ist, die Verkehrslenkung von Fahrzeugen zu beeinflussen.

2.3.4 Bottleneck Problem

Oft ist die Staubildung im Straßenverkehr das Resultat eines Bottlenecks. Dabei ist ein Bottleneck eine Inhomogenität auf Grund reproduzierbarer externer Gründe [Sch04]. Es wird häufig vermutet, dass diese durch Fahrbahnverengungen, Abfahrten oder Brücken entstehen. Abbildung 2.5 zeigt ein Straßennetz als Graph, in dem ein Bottleneck auftaucht. Wenn dieses überwunden werden muss, ist die Wahrscheinlichkeit höher, dass Staus entstehen. Abbildung 2.6 zeigt ein Verkehrsnetz mit wahrscheinlich mehreren Bottlenecks, die einen Breakdown auslösen können. Diese können anhand des kritischen Flusses genauer betrachtet werden.

Die Lokalität von Bottlenecks steht im Zusammenhang mit Breakdowns [KKS14]. An einem Bottleneck kommt es zu einer Verkehrsstörung, wenn der freie Verkehrsfluss in einen überfüllten Verkehr übergeht.

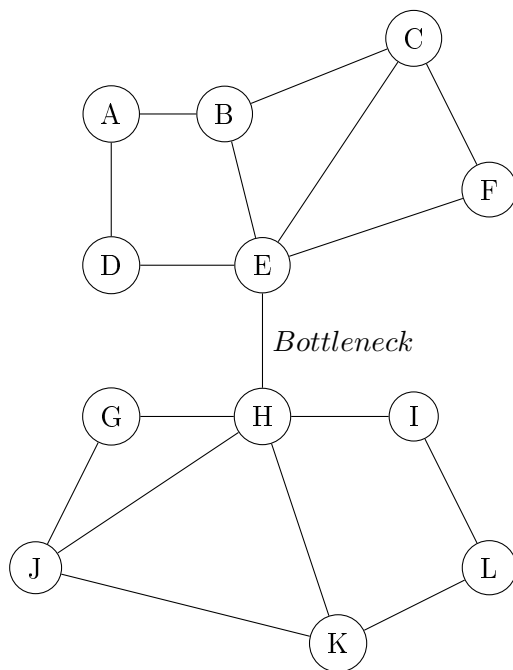


Abbildung 2.5: Bottleneck Problematik im Graphen. Bei der Bestimmung eines Pfades von einem Knoten $s \in \{A, B, C, D, E, F\}$ zu einem Knoten $d \in \{G, H, I, J, K, L\}$ muss die Kante zwischen E und H ausgewählt werden und wird dadurch zu einem *Bottleneck*.

2.4 Straßennetzmodellierung

Ein Netz, das aus Straßen und Kreuzungen besteht, wird als Graph modelliert. Die Straßennetzmodellierung als Graph eignet sich, um den kürzesten Pfad zwischen zwei Punkten zu finden, der eine Route für ein Fahrzeug darstellt.

Straßen lassen sich als Kanten abbilden, die einen Startpunkt und einen Endpunkt besitzen. Diese Start- und Endpunkte beschreiben Kreuzungen oder Knoten mit festgelegten Koordinaten. Um jede Fahrbahn und die entsprechende Richtung abzubilden, wird ein Straßennetz als gerichteter Graph $G = (E, N, c)$ modelliert und besteht damit aus Knoten $n \in N$, Kanten $e \in E$ und Kosten c , um eine Kante mit einem Fahrzeug zu nutzen. Ein Knoten $n \in N$ wird durch eine Position genauer beschrieben, die durch Koordinaten bestimmt ist und besitzt ausgehende und eingehende Kanten. Weiterhin lässt sich die Distanz zwischen zwei Knoten $n, m \in N$ durch die euklidische Norm $\|n - m\|$ berechnen. Diese Strecke zwischen n und m wird durch eine gerichtete Kante $(u, v) \in E \subset \{(p, q) | p, q \in N, p \neq q\}$ dargestellt. Eine Kante $e \in E$ besitzt in einem Straßennetz einerseits die Eigenschaft der Länge $l : E \rightarrow \mathbb{R}_+^*$ und andererseits eine Geschwindigkeit $s : E \rightarrow \mathbb{R}_+^*$, die zum Überqueren der Straße vorgegeben ist. Diese beiden Eigenschaften sind dazu geeignet, die Kosten für

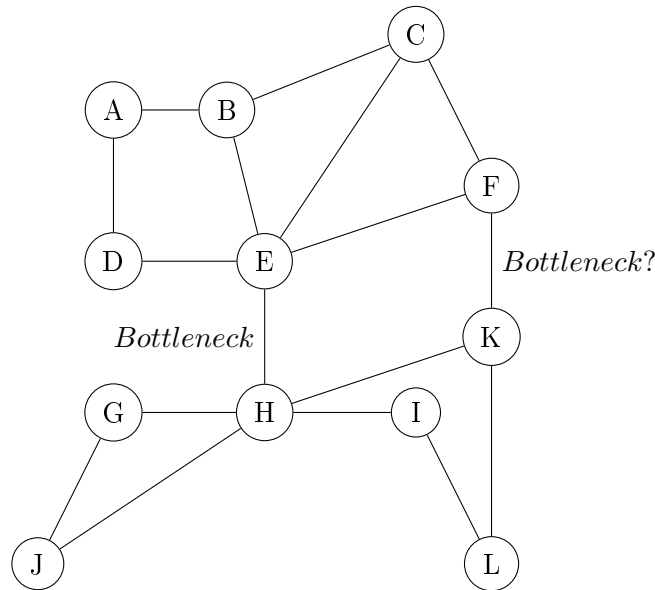


Abbildung 2.6: Beispiel eines unübersichtlichen Bottleneck Problems.

die Überquerung der Straße auszudrücken. Die Fahrzeit eines Fahrzeugs, das eine Straße überqueren möchte, sind die Kosten einer Kante im Straßennetz. Mit

$$c(e) := \frac{l(e)}{s(e)} \quad (2.4)$$

werden die Kosten c berechnet, um die Kante e zu überqueren, wenn ein Fahrzeug die maximal erlaubte Geschwindigkeit fährt.

Das Ziel jedes Fahrzeugs sollte es nun sein, mit bekanntem Startknoten und bekannten Zielknoten den kürzesten Weg zwischen diesen zu fahren. Mit der Modellierung als Graph ist es nun möglich, einen Pfad zu berechnen, der die geringsten Kosten verursacht. Somit wird der Dijkstra Algorithmus [Dij59] eingesetzt, um den kürzesten Pfad zu finden.

2.4.1 Kürzeste Routen im Straßennetz mit Dijkstra Algorithmus

Durch die Modellierung des Straßennetzes als Graphen lässt sich der Dijkstra Algorithmus anwenden. Es wird für ein Fahrzeug eine Route gesucht, welche sich an einem bestimmten Knoten befindet und einen Knoten als Ziel besitzt. Die Grundidee, um den Dijkstra Algorithmus einzusetzen, basiert auf der Suche des kürzesten Pfades zwischen zwei Knoten $u, v \in N$ [Sch12]. Zur Ausführung des Algorithmus muss der Startknoten $s \in N$ bekannt sein, der im gerichteten Graphen $G = (E, N, c)$ vorkommt. Anschließend wird der nachfolgend beschriebene Algorithmus durchgeführt, den kürzesten Pfad im Graphen zu finden. Angewendet auf ein Straßennetz, besteht die Möglichkeit, den aktuellen Verkehr in die Kantengewichte einzurechnen. Ein kürzester Pfad auf Basis der Kantengewichte ohne eingerechnete Verkehrsauslastung führt ein Fahrzeug in der schnellsten Zeit vom Startknoten zum Zielknoten.

Der Dijkstra Algorithmus setzt alle Knotengewichte initial auf ∞ [Sch12]. Das Gewicht des Startknotens s bekommt den Wert 0. In einem iterativen Verfahren werden die Gewichte der Knoten auf die kürzesten Summen von Gewichten aktualisiert. Nicht verwendete Kanten werden entfernt, sodass von s alle anderen Knoten auf dem Pfad mit geringstem Gewicht erreicht werden können. Interessant für eine Verkehrslenkung ist nun der kürzeste Pfad zum Zielknoten.

2.4.2 Dijkstra Rang

Der Dijkstra Rang (*dijkstra rank*) lässt sich durch die Ordnung von Knoten einer bestimmten Distanz zu einem Startpunkt beschreiben [SS12]. Man kann als Interpretation auch eine Lokalität zweier Punkte nutzen. Nachdem der Dijkstra Algorithmus durchgeführt wurde, befinden sich die verwendeten Knoten in einer resultierenden Reihenfolge. Ein festgelegter Startknoten s ist dabei notwendig. Der Dijkstra Rang eines Knotens t bestimmt den Rang in Bezug auf die Reihenfolge der Knoten, die sich durch die Durchführung des Dijkstra Algorithmus ergeben [DSSW09].

Zum Beispiel existiert nach Durchführung des Dijkstra Algorithmus mit Startknoten s ein kürzester Pfad zu einem Zielknoten z . Jeder Knoten d_i , der auf dem Pfad zwischen den Knoten s und z liegt, hat den Dijkstra Rang als Abstand zum Startknoten s . Dadurch wird die Lokalität von einzelnen Knoten beschrieben.

2.5 Bestärkendes Lernen (Reinforcement Learning)

Maschinelles Lernen lässt sich in drei Bereiche unterteilen. Die drei Bereiche sind überwachtes, unüberwachtes und bestärkendes Lernen. Grundsätzlich geht es darum, Wissen aus Daten zu generieren. Überwachtes Lernen benötigt einen Datensatz, der jedem Datenpunkt ein Label zuordnet. Es müssen folglich zusätzliche Informationen vorhanden sein, um lernende überwachte Algorithmen anwenden zu können. Unüberwachtes Lernen benötigt keine weiteren Informationen über den Datensatz und hat das Ziel, unbekannte Strukturen im Datensatz ausfindig zu machen. Bestärkendes Lernen hingegen lernt aus der Interaktion mit einer Umgebung. Es ist also eine Lernmethode, die ihr Wissen durch die Ausführung von Experimenten erhält [SB18].

Bestärkendes Lernen beschäftigt sich mit Agenten in einer Umgebung, die mit dieser interagieren können. Die Hauptkomponenten des bestärkenden Lernens setzen sich aus Agenten, Umgebung, Belohnungen und Aktionen zusammen. Zusätzlich gibt es noch einen Zustand der Umgebung, der sich ändern kann. Beim bestärkenden Lernen geht es grundsätzlich darum, herauszufinden, wie auf Situationen durch Aktionen reagiert wird. Das Lernergebnis wird durch eine Maximierung einer numerischen Belohnung erzielt. Es wird die Aktion ausgewählt, die die beste Belohnung der Umgebung erhält. Abbildung 2.7 zeigt

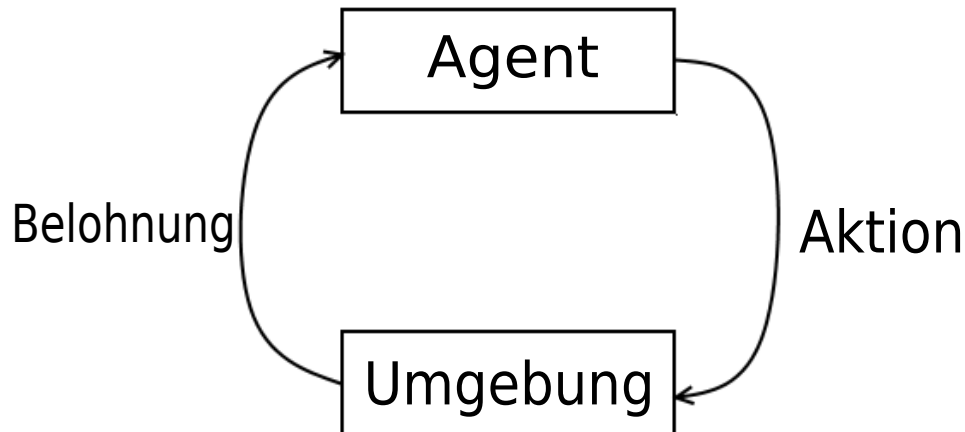


Abbildung 2.7: Eine Übersicht des bestärkenden Lernens. Es gibt Aktionen zwischen Agenten und Umgebung, die eine Belohnung an den Agenten zurückgibt.

den Ablauf beim bestärkenden Lernen. Ein Agent führt Aktionen in einer Umgebung aus und bekommt von der Umgebung eine Belohnung für diese Aktion.

Der Vorgang der Belohnung an den Agenten durch die Umgebung wird immer nur dann durchgeführt, sobald eine Aktion ausgewählt wurde. Ziel ist es, eine möglichst hohe Belohnung zu erhalten. Deswegen wird der Vorgang mehrfach wiederholt. Man nennt dieses Problem auch Ausbeutung gegen Erkundung [SB18] und ermöglicht einen entscheidenden Faktor für den Erfolg von bestärkendem Lernen. Um eine aussagekräftige Strategie zu erhalten, sollten mehrere Aktionen mit Belohnungen versehen werden. Wenn die Erkundung ausreichend Ergebnisse geliefert hat, werden Aktionen auf Belohnungen geprüft.

2.5.1 Learning from Bandit Feedback

Das sogenannte *k-armed Bandit Problem* beschreibt eine Methode, die auf das bestärkende Lernen zurückzuführen ist. Anschaulich lässt sich das Problem so erläutern, dass man in wiederholender Weise eine Auswahl an Aktionen hat, die ausgeführt werden können. Je nach Auswahl, erhält man eine numerische Belohnung auf Basis der Auswahl.

Dabei wird eine Aktion, die zum Zeitpunkt t ausgewählt wurde als A_t bezeichnet und die korrespondierende Belohnung als R_t . Der Wert einer beliebigen Aktion a ist durch

$$q_*(a) = \mathbb{E}[R_t | A_t = a] \quad (2.5)$$

gegeben und drückt die erwartete Belohnung dafür aus, dass a ausgewählt wird [SB18].

2.5.2 Policy Optimizer for Exponential Models (POEM)

Der Algorithmus *Policy Optimizer for Exponential Models* wird eingesetzt, um ein bestärkendes Lernen für strukturierte Vorhersagen durchzuführen. Der POEM Algorithmus nutzt das *Counterfactual Risk Minimization* (CRM) Prinzip, um Batch Lernmethoden mit *Bandit Feedback* zu ermöglichen [SJ15]. Mit dem CRM Prinzip sind Vorhersagen möglich.

POEM funktioniert mit einer Datengrundlage, die die Form eines geloggtten *Bandit Feedbacks* hat. Die Anwendung des Algorithmus führt eine Klassifikation von einzelnen Labels durch. POEM hat das Ziel, eine Strategie h zu optimieren. Diese Strategie existiert initial durch h_0 und wird aus den geloggtten Eingabedaten abgeleitet und optimiert.

Sei $X \subset \mathbb{R}^m$, $m \in \mathbb{N}$ der Eingaberaum und $Y \subset \{(0), (1)\}$ der Ausgaberaum. Dabei gilt für den Eingaberaum X , dass dies alle aufgezeichneten und geloggtten Eingabevariablen enthält und $\mathbf{x} \in X$ stellt einen Eingabevektor dar. Der Ausgaberaum Y repräsentiert ein Label, das entweder (0) oder (1) sein kann. Hierbei ist $y \in Y$ dann das vorhergesagte Ausgabelabel für den Eingabevektor \mathbf{x} . Die existierende Strategie $h_0(Y|\mathbf{x})$ ist eine Wahrscheinlichkeitsverteilung über dem Ausgaberaum Y und Vorhersagen werden durch ein Ziehen aus $y \sim h_0(Y|\mathbf{x})$ durchgeführt. Also ist $h_0(y|\mathbf{x})$ die Wahrscheinlichkeit, die bei Eingabe \mathbf{x} entsteht, dass y ausgewählt wird. Zusätzlich sei $\delta : X \times Y \rightarrow \mathbb{R}$ die Belohnung des bestärkenden Lernproblems als kardinaler Verlust, der sich aus allen beobachteten Paaren (\mathbf{x}, y) zusammensetzt. Es gilt, dass kleinere Werte auf eine höheres Eintreffen von y für \mathbf{x} hindeuten. Um nun die Strategie h_0 zu optimieren, wird für POEM der Datensatz

$$\mathcal{D} = \{(\mathbf{x}_1, y_1, \delta_1, p_1), \dots, (\mathbf{x}_n, y_n, \delta_n, p_n)\} \quad (2.6)$$

mit $i \in \mathbb{N}_{\leq n}$ genutzt, wobei $\delta_i = \delta((\mathbf{x}_i, y_i))$ und $p_i = h_0(y_i|\mathbf{x}_i)$ gelten. Als nächsten Schritt in POEM wird der Hypothesenraum \mathcal{H}_{lin} nach einer Strategie h_w durchsucht, die das maximale Eintreffen erfüllt. Die Hypothese $h(Y|\mathbf{x})$ definiert hierbei eine Wahrscheinlichkeitsverteilung und wird durch den Einsatz eines Gradientenabstiegs auf dem zu erwartenden und geschätzten Verlust als empirische Standardabweichung erhalten. Bei der Klassifizierung von einzelnen Klassen zieht die Strategie $h_w \in \mathcal{H}_{lin}$ y mit

$$h_w(y|\mathbf{x}) = \frac{\exp(y_1 * w^T \mathbf{x})}{(1 - \exp(w^T \mathbf{x}))}, \quad (2.7)$$

wobei $w \in \mathcal{R}^m$ der gelernte Vektor durch POEM ist und y_1 das Label aus y darstellt.

2.6 Breakdown Minimization Principle (BMP)

Das *Breakdown Minimization Principle* (BMP) kann dazu eingesetzt werden, Verkehr zu lenken. Die Anzahl der *Breakdowns* an Kreuzungen wird dabei minimiert. *Breakdowns* sind Übergänge vom frei fahrenden Verkehr zu einem synchronisierten Verkehr [Vra17]. Das *Breakdown Minimization Principle* setzt auf der Tatsache auf, dass sich Fahrzeuge

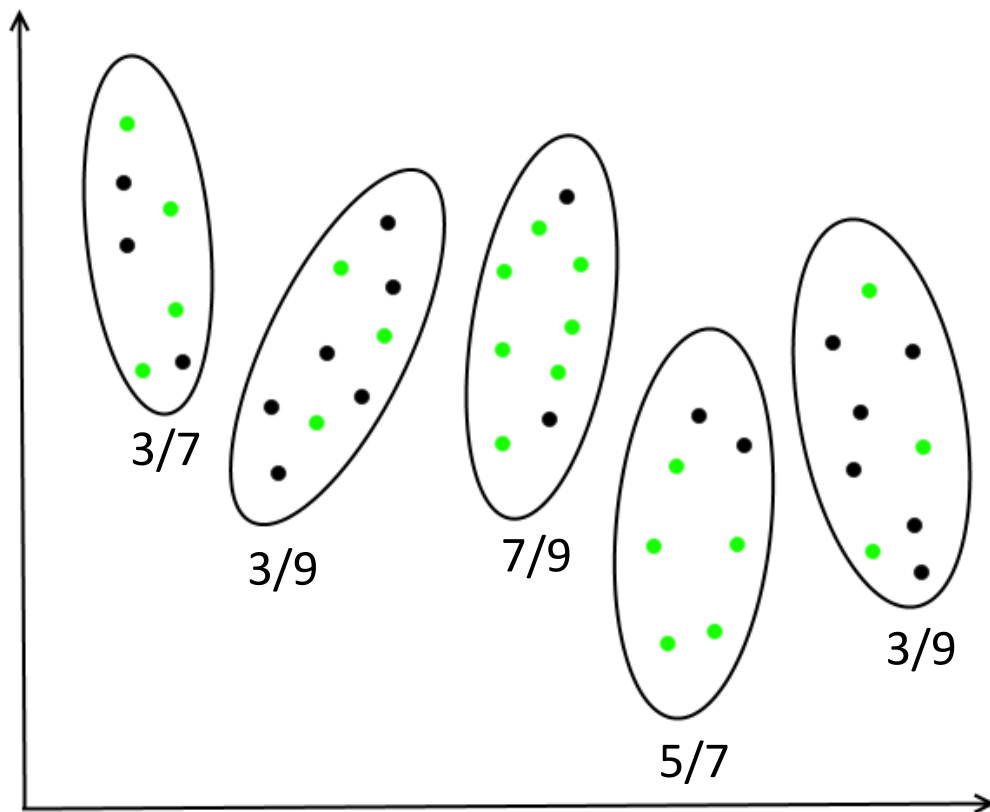


Abbildung 2.8: LLP Bags, die grüne oder schwarze Instanzen enthalten.

sehr schnell an Bottlenecks anhäufen und somit zu Staus an diesen Engstellen führen. An Bottlenecks existiert ein kritischer Fluss C , der bei Überschreitung darauf hindeutet, dass es zu einem Breakdown kommt. Es ist dann eine Wahrscheinlichkeit $P(q)$ berechenbar, die das Eintreffen eines Einbruchs des Verkehrsflusses in einem Zeitfenster T beschreibt [Vra17]. Das Ziel des *Breakdown Minimization Principle* liegt darin, Staus zu vermeiden, indem an keinem Bottleneck der Zustand $q^k < C^k$ eintritt.

2.7 Label Proportionen

Learning from Label Proportions (LLP) ist ein Maschinenlernproblem, das Trainingsdaten in Form von *Bags* von Instanzen nutzt [AC16]. Nur das Klassen Label jeder *Bag* ist bekannt. Die Anzahl der Gruppen ist mindestens so groß wie die Anzahl an Klassen [QSCL09]. Es ist ziemlich aufwändig *Label Proportions* zu lernen, da es sich um ein NP-Schweres Problem handelt [FR17].

Abbildung 2.8 zeigt ein Beispiel mit verschiedenen Instanzen in verschiedenen Bags, die unterschiedlich eingefärbt sind. Dabei sind grüne Punkte positiv und schwarze Punkte negativ. Es kann nun für jede Bag die Proportion berechnet werden. Die Abbildung enthält

den Wert der Label Proportion an jeder Bag. Die Bags werden nun nicht einer Klasse zugeordnet, sondern geben nur den Wert anhand der Vorkommen zurück.

2.7.1 Definition *Learning from label proportions* aus [SM11]

Sei X eine Instanz einer Menge von Features $\{X_1 \times \dots \times X_m\}$ und sei $Y = \{y_1, \dots, y_m\}$ eine Menge von Klassenlabels. $P(X, Y)$ stellt eine unbekannte Verteilung von Beobachtungen mit dem jeweiligen Klassenlabel dar. Es ist eine ungelabelte Beobachtung $U = \{x_1, \dots, x_n\} \subset X$ gegeben mit einer Partitionierung in h disjunkte Gruppen G_1, \dots, G_h . Weiterhin sind Proportionen $\pi_{ij} \in [0, 1]$ gegeben mit Label y_j aus Gruppe G_i für jede Gruppe und jedes Label. Das Ziel ist eine Modellfunktion $g : X \rightarrow Y$, um eine Vorhersage eines Labels $y \in Y$ für eine Beobachtung $x \in X$ zu erhalten [SM11]. Im allgemeinen Fall minimiert der Fehler

$$Err_P = E[L(Y, g(x))] \quad (2.8)$$

die Verlustfunktion $L(Y, g)$ bei Anwendung auf P . Die Label Proportionen π_{ij} werden als $h \times l$ Matrix $\Pi = \pi_{ij}$ dargestellt, wobei sich die Summe der Werte $\Pi_{i,\cdot} = (\pi_{i1}, \dots, \pi_{il})$ ergibt. Somit lässt sich die Proportion eines Labels y_j zusammen mit Π durch

$$\eta(\Pi, y_j) = \frac{1}{n} \sum_{i=1}^h |G_i| \cdot \pi_{ij} \quad (2.9)$$

berechnen [SM11]. Die Multiplikation von π_{ij} mit der jeweiligen Gruppengröße $|G_i|$ ergibt die Häufigkeiten μ_{ij} der Beobachtungen mit Label $y_j \in Y$ der Gruppe G_i .

Im folgenden Abschnitt werden Ergebnisse von Arbeiten vorgestellt, die sich mit Verkehrsflüssen und Verkehrslenkung beschäftigen und dabei POEM und LLP nutzen.

2.8 Verwandte Arbeiten

In Großstädten kommt es zu immer häufigerer Aufstauung von Fahrzeugen, sodass diese durch Verkehrslenkung vermindert werden muss, da sich die Anzahl der Fahrzeuge jährlich erhöht. In diesem Abschnitt werden ähnliche Ansätze der dynamischen Verkehrslenkung und Vorhersage von Verkehrsflüssen behandelt.

Es existieren bereits mehrere Arbeiten, die sich mit intelligentem Verkehrsrouting befassen. Ziel ist es, durch Auswahl von Routen einen Verlust zu minimieren, der für die Verkehrsteilnehmer entsteht [LPBM14]. Der Verlust ist durch die Zeit definiert, die durch den Abstand der Zeit einer nicht-optimalen Route zur Zeit einer optimalen Route entsteht.

Die Arbeit [LS17] beschreibt eine Möglichkeit, um auf Basis von zentralen Verkehrsdaten den Verkehr zu lenken. Dabei wird ein bestärkendes Lernen verwendet. *Bandit Learning* ist eine Methode des bestärkendes Lernens, wobei das Verhalten eines *Bandits* unter der Berücksichtigung einer Belohnung gelernt wird [LS17]. Fahrzeuge sind die Agenten, die

sich in einem Straßennetz aufhalten. Aktionen beschreiben die Wahl einer Straße, um eine Route vom Startpunkt zum Zielpunkt zu erhalten. Sobald eine Straße gewählt ist, erhält der Agent eine Belohnung.

In der Arbeit [LSM15] wird eine Methode vorgestellt, um Verkehrsfluss Label von verteilten Sensoren vorherzusagen. Grundsätzlich wird in dieser Arbeit eine verteilte Lernmethode vorgestellt, die *Spatio-Temporal Local Models* nutzt [LSM15]. Als Ausgabe werden Labels von in der Zukunft liegenden Zeitpunkten bestimmt, die zugehörig zu einem Knoten sind. Dabei können die gesamten Sensordaten genutzt werden oder nur die Daten einer Teilmenge von Sensoren.

LLP ist eine Möglichkeit für Kommunikation von Sensoren, die sich dadurch auszeichnet, möglichst kleine Nachrichten effizient versenden zu können. Diese Kommunikationseffizienz ist ein wichtiger Grund, LLP zu verwenden. Alternativ gibt es beispielsweise die kNN Methode. Ein Nachteil zur kNN Methode liegt darin, dass die Genauigkeit von LLP schlechter ist, jedoch werden mit der LLP Methode weniger Übertragungsbits verwendet als bei kNN.

Die Arbeit *Untersuchung von Strategien zur Beeinflussung von Fahrzeugen zur Verkehrslenkung* [Vra17] nutzt mehrere verschiedene Methoden, um einen Einfluss auf die Lenkung von Fahrzeugen zu haben. In einem erfolgreichen Experiment wird das *Break-down Minimization Principle* (BMP) benutzt. Hierbei handelt es sich ebenfalls um ein Verfahren, das den Verkehr lenkt, um Staus zu vermeiden.

In der Arbeit [ARI⁺15] wurden Vorhersagen von Verkehrsflüssen auf der Basis von begrenzten Daten untersucht. Dabei wurden große Verkehrsnetze untersucht, wie gut Verkehrsflüsse erkannt wurden, wenn nur ein kurzer Zeitraum beobachtet wird. Die Ergebnisse dabei sind eine gute Genauigkeit bei der Anwendung einer Vorhersage. Eine Erweiterung durch die Interaktion von Fahrzeugen zur Verkehrsumgebung wurde dabei nicht untersucht, kann jedoch zu einer Verbesserung beitragen.

Eine weitere Arbeit [dSYM⁺16] CHIMERA nutzt das *Intelligent Transportation System* (ITS) [FJM⁺01]. Dabei werden keinen festen Verkehrssensoren benutzt, sondern mehrere *road side units* (RSU), um den Straßenverkehr aufzuzeichnen. Der Unterschied liegt darin, dass RSU lokale Messungen von Daten nutzt, die durch Sensoren in Fahrzeugen aufgezeichnet werden. Es werden Überfüllungen anhand des k-Nächsten-Nachbar Algorithmus vorhergesagt und Fahrzeuge durch eine Verkehrslenkung beeinflusst.

In Abgrenzung zu den genannten Arbeiten wird in dieser Arbeit eine dezentrale dynamische Verkehrslenkung eingesetzt. Dieser Ansatz nutzt aktuelle Verkehrsinformationen und verbindet diese mit vorhergesagten Verkehrsflusslabels, um Überfüllungen in Verkehrsnetzen zu erkennen.

Im nachfolgenden Kapitel wird die Methodik gezeigt, um durch POEM und LLP Einfluss auf eine dezentrale dynamische Verkehrslenkung zu haben.

Kapitel 3

Methodik

Dieses Kapitel behandelt die methodischen Inhalte dieser Arbeit. Dabei wird das Vorgehen der intelligenten Verkehrslenkung auf Basis lokaler Features mit Informationen von lernenden Sensoren beschrieben. Diese lokal extrahierten Features bilden die Grundlage für ein bestärkendes Lernen, das überfüllte Straßen und Staus vorhersagt, sodass die Verkehrslenkung darauf reagiert. Lernende Sensoren steuern die Verkehrslenkung durch die Vorhersage von Verkehrslabels einzelner Straßen in der Umgebung eines Fahrzeugs.

3.1 Dynamische Verkehrslenkung

In *Smart Cities* und intelligenten Verkehrssystemen wird der Verkehr auf den Straßen durch Sensoren aufgezeichnet, wie in Abschnitt 2.2 näher erklärt wird. Es besteht die Möglichkeit, aktuelle Stausituationen zu erkennen, um diese durch dynamisch generierte Routen zu umfahren [LS17]. Dabei ist es wichtig, dass aktuelle Informationen über den Verkehr genutzt und analysiert werden. Letztendlich werden auf Basis der aktuellen Verkehrsdaten Routen generiert, die Fahrzeuge in einer Verkehrsumgebung lenken. Diese Routen sind auf den dynamischen Verkehr angepasst und können sich entweder zu verschiedenen Uhrzeiten unterscheiden oder auch auf plötzlich auftretende Veränderungen des Verkehrs reagieren.

Das Besondere an dynamischer Verkehrslenkung im Vergleich zu statischer Verkehrslenkung besteht darin, dass bei dynamischer Verkehrslenkung schnell auf mögliche Überfüllungen auf Straßen reagiert werden kann. Statische Verkehrslenkung nutzt nur die Situation des Startzeitpunktes und ermöglicht keine Reaktion auf den aufkommenden Verkehr. Ist ein Fahrzeug auf einer Straße, kann dieses seine Route jedoch nur an der nächsten Kreuzung ändern. Somit lässt sich dynamische Verkehrslenkung auf eine Aktualisierung der Routen reduzieren, die kurz bevor Kreuzungen angefahren werden, durchgeführt werden.

Um ebenfalls die Performanz zu erhöhen, sollen Verkehrslenkungsentscheidungen nur getroffen werden, wenn diese notwendig sind. Das bedeutet, dass Fahrzeuge ausschließlich beim Anfahren von Kreuzungen die Möglichkeit haben, eine bessere Route zu wählen. Die

Auswahl einer besseren Route entsteht auf Basis der gelernten Modelle aus beobachteten Daten und wird durch die Routenberechnung in 3.1.4 bestimmt. Kreuzungen sind mit Sensoren ausgestattet. Sobald ein Fahrzeug einen Sensor anfährt, befindet sich das Fahrzeug an einer Kreuzung.

Die dynamische Verkehrslenkungsentscheidung wird durch Daten beeinflusst, die insbesondere in der lokalen Umgebung eines Fahrzeugs auftreten. Es besteht daher eine geringe Notwendigkeit, Zugriff auf alle Verkehrsdaten einer Verkehrsumgebung zu haben. Bevor eine Kreuzung angefahren wird, entscheidet sich, welche zukünftige Route von dieser Position aus genutzt wird, um zum Zielpunkt zu gelangen.

3.1.1 Dezentrale Verkehrsdaten

Dezentrale Verkehrsdaten bezeichnen die eingeschränkte Möglichkeit, als Fahrzeug Verkehrsdaten von der Verkehrsumgebung zu erhalten. In diesem Fall sind Verkehrsdaten dann nur beim Anfahren von Sensoren vorhanden. Fahrzeuge können auf ihrem Weg Verkehrsdaten aufzeichnen und speichern. Insbesondere an Kreuzungen erhalten Fahrzeuge die Möglichkeit, mehrere Sensoren anzusprechen und die jeweiligen Verkehrsdaten verarbeiten zu können. Im Vergleich zu zentralen Verkehrsdaten, die beispielsweise durch Anfrage eines Servers erhalten werden können, stellen dezentrale Verkehrsdaten eine beschränkte Datengrundlage dar. Bei dezentraler Datengrundlage gibt es für den aktuellen Zeitpunkt nur sehr wenige Datenpunkte. Jedoch skalieren dezentrale Verkehrsdaten besser und sind nicht so fehleranfällig wie zentrale Verkehrsdaten. In diesem Ansatz beschreiben die dezentralen Verkehrsdaten jedoch nicht nur die aktuelle Verkehrslage, sondern geben eine Vorhersage der Verkehrsdaten, die für die zukünftige Entwicklung berechnet werden. Diese Informationen sind sehr aussagekräftig, da für eine Verkehrslenkung, die den nächsten Schritt plant, zukünftige Verkehrsdaten die Zeitpunkte beschreiben, die die Ausführung des nächsten Schrittes beeinflussen.

Ein Fahrzeug, das seine Verkehrslenkung verbessern will, hat die Möglichkeit, aufgezeichnete Daten zu nutzen und diese mit Daten der aktuell angefahrenen Sensoren zu vergleichen. Es ist also zunächst eine Datengrundlage vorhanden, die vor der angetretenen Fahrt bereits existiert und dann auf dem Weg erweitert wird. Mit jedem Anfahren eines Sensors wird der Datenpunkt zur Datengrundlage hinzugefügt.

3.1.2 Feedback der Verkehrsumgebung durch Überfüllungsetiketten

Auf Straßen kann immer der Zustand der Überfüllung erreicht werden, wenn Fahrzeuge sich auf der gleichen Straße häufen. Diese Überfüllung kann einerseits durch die Wahl der gleichen Routen von Fahrzeugen zur gleichen Zeit auftreten oder auch durch zu langsames Überqueren von Straßen, sodass sich dort Fahrzeuge aufstauen. Die langsame Überquerung oder Reduzierung der Durchschnittsgeschwindigkeit steht im Zusammenhang mit dem Ver-

kehrfluss, der durch Überfüllung verringert wird, wie in Abschnitt 2.3 gezeigt wird. Um einen Indikator für Überfüllung auf Straßen zu erhalten, gibt es viele Methoden. Dabei werden folgend drei Methoden vorgestellt, die auf einem Schwellwertproblem basieren und eine Bewertung einer Verkehrsumgebung generieren können, welche auf Überfüllung hinweist. Im Kontext des bestärkenden Lernens werden diese Indikatoren als mögliche Belohnungen der Verkehrsumgebung verwendet.

Um einer Straße ein bestimmtes Label zuzuordnen und dieses Label zu erkennen, wird zunächst mit $\delta_{\text{Überfüllung}}$ unterschieden, ob eine Straße überfüllt ist oder nicht. $\delta_{\text{Überfüllung}}$ wird aus den Verkehrsdaten einer Straße berechnet. Es ist ebenfalls möglich, den Zustand von Staus durch δ_{Stau} auf Straßen abzubilden, der aus einer Überfüllung resultiert, jedoch macht es mehr Sinn, bereits überfüllte Straße zu umfahren und nicht nur Staus, da bereits eine Überfüllung einer Straße dazu führen kann, dass sich die Fahrzeit eines Fahrzeugs erhöht. Es werden also keine Stau Label in diesem Anwendungsfall verwendet, da eine erkannte Überfüllung ausreichend ist, um die Entscheidung zu treffen, eine Straße nicht zu befahren. Um folgend einen Indikator für eine Überfüllung zu geben, werden δ_{Dichte} , $\delta_{\text{Geschwindigkeit}}$ und $\delta_{C_{max}}$ verwendet.

Es existiert eine kritische Dichte, die überschritten werden muss, um $\delta_{\text{Überfüllung}}$ zu berechnen, um eine überfüllte Straße zu erkennen. Dabei wird die maximale Kapazität von Straßen erreicht, wodurch eine Überfüllung entstehen wird. δ_{Dichte} kann hieraus abgeleitet werden, indem die Dichte auf einer Straße betrachtet wird und mit der kritischen Dichte verglichen wird. Als kritische Dichte wird ein Wert bezeichnet, bei dem eine Überfüllung entstehen wird. Berechnen lässt sich die kritische Dichte durch Erfahrungswerte. Dabei spielen Fahrzeuglänge und Bremswege eine Rolle, um einen Wert dafür zu berechnen, wie viele Autos auf eine Straße passen. Dieser Wert lässt sich allgemein durch ein Siebtel der Dichte eines Staus bestimmen [Cha14], sollten alle Fahrzeuge auf einer Straße mit kleinem Sicherheitsabstand hintereinander stehen. Das Label für δ_{Dichte} bestimmt sich folglich aus der Überschreitung dieser kritischen Dichte.

Die Geschwindigkeit von Fahrzeugen gibt ebenfalls Informationen über mögliche Überfüllung, wie in Studien gezeigt wurde [RBD⁺10]. Dabei wird geringe Geschwindigkeit als Indikator für Überfüllung genutzt. Geringe Geschwindigkeit impliziert hier erneut einen Stau, der durch diese Methode ebenfalls abgefangen wird. Es reicht bereits aus, wenn von der erlaubten Geschwindigkeit im Durchschnitt eine Abweichung von 10 km/h erreicht wird, sodass $\delta_{\text{Geschwindigkeit}}$ berechnet werden kann. Hierbei handelt es sich ebenfalls um ein Schwellwertproblem mit der Durchschnittsgeschwindigkeit d aller Fahrzeuge auf einer Straße s und einer erlaubten Geschwindigkeit e , sodass $\delta_{\text{Geschwindigkeit}} \in \{(0), (1)\}$ ein Label (0) für nicht überfüllt oder ein Label (1) für überfüllt annimmt mit

$$\delta_{\text{Geschwindigkeit}} = \begin{cases} (0), & d_s \geq (e_s - 10) \\ (1), & d_s < (e_s - 10) \end{cases} . \quad (3.1)$$

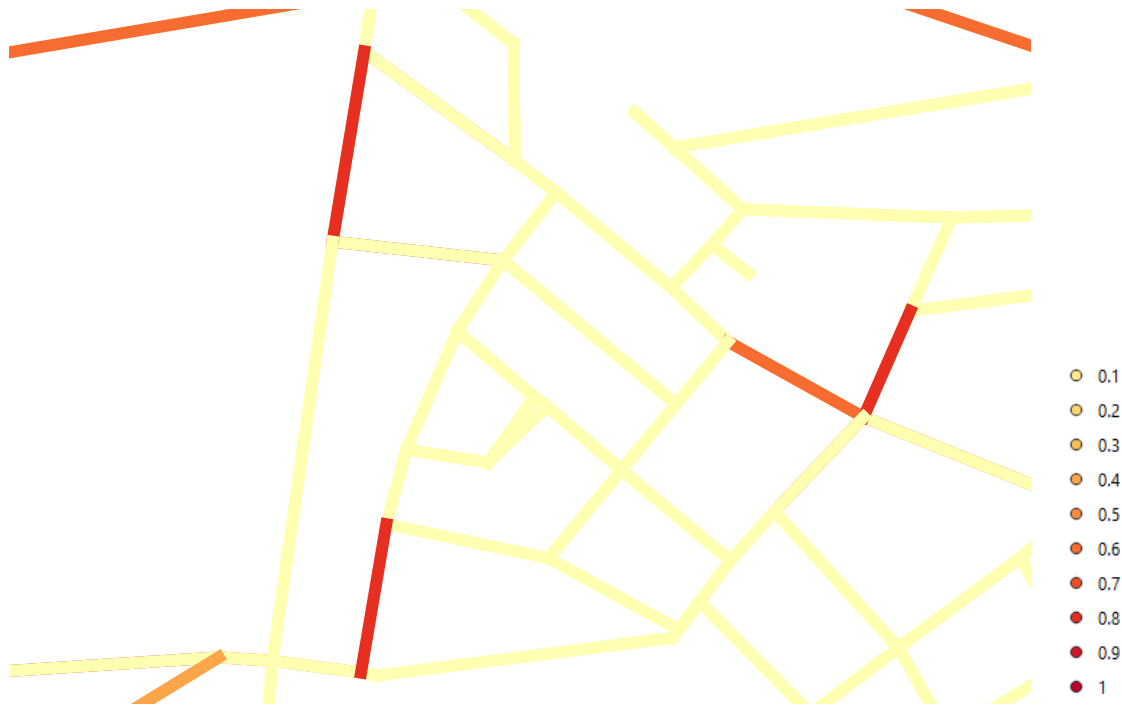


Abbildung 3.1: Ein Ausschnitt aus der kritischen Fluss Map. Dabei wird der Wert für den kritischen Fluss jeder Straße anhand der Skala dargestellt. Ein roter Wert deutet dabei auf einen geringen kritischen Fluss hin, der bei Erreichen einen Breakdown auslöst und schnell erreicht wird.

Dieser Indikator wird häufig in Simulationen verwendet, da es zu guten Ergebnissen führt.

Mit der Durchschnittsgeschwindigkeit von Fahrzeugen kann nicht nur ein $\delta_{Geschwindigkeit}$ berechnet werden, das eine Abweichung der erlaubten Geschwindigkeit auf Straßen nutzt, sondern eine Staugeschwindigkeit v_{Stau} . Diese ist unabhängig von äußeren Umständen, wie zum Beispiel Wetterverhältnissen oder Straßenzuständen, und lässt sich auf etwa $v_{Stau} \approx 15\text{km/h}$ festlegen [Sch04]. Durch diese Konstante wird ein Indikator für einen Stau auf einer Straße gegeben.

Eine weitere Methode stellt die Überschreitung des kritischen Flusses dar, der Straßen als überfüllt kennzeichnen kann. Diese Methode wird in 2.3.3 ausführlich beschrieben. $\delta_{C_{max}}$ erkennt eine Straße als überfüllt, wenn der Schwellwert C_{max} überschritten ist.

Der kritische Fluss kann im Gegensatz zur Dichte oder zur Durchschnittsgeschwindigkeit nicht direkt aus den aktuellen Verkehrsdaten berechnet werden. Es wird für jede Straße, die gerichtet auf eine Kreuzung zuführt, ein Wert C_{max} gemessen. Dieser Wert wird für jede Straße aus einer Simulation berechnet, indem betrachtet wird, ab welchem Fluss einer Straße Fahrzeuge langsamer werden oder stehen bleiben. Abbildung 3.1 zeigt Kreuzungen und die unterschiedlichen Werte von Straßen, die auf Kreuzungen zuführen. Sollte dieser Wert beim Anfahren der Kreuzung durch die aktuellen Verkehrsdaten in Form von aktuellem Verkehrsfluss überschritten werden, gibt $\delta_{C_{max}}$ der Straße eine Einordnung

als überfüllt. Die Überfüllung entsteht, da es durch Überschreiten des kritischen Flusses zu einem Breakdown kommt und somit kein freier Fluss mehr möglich ist.

3.1.3 Gewichteter Überfüllungsgraph

Die hauptsächlich beeinflussende Datenstruktur für die Berechnung der Route für ein Fahrzeug stellt ein gewichteter Graph $\mathcal{G}(E, N, c)$ dar. Dieser wird zunächst auf Basis des Straßennetzes mit $G(E, N, c)$ generiert. Dabei existieren Kanten $e \in E$ und Knoten $n \in N$. Kanten besitzen eine Gewichtung, die initial aus der möglichen Fahrzeit berechnet wird. Das Ziel eines Überfüllungsgraphen liegt darin, die Gewichtungen der Kanten im Graphen durch Informationen über Verkehrsverengungen zu aktualisieren und zu erhöhen. Wie die Überfüllung erkannt wird, unterscheidet sich in unterschiedlichen Verfahren. Überfüllte Kanten werden durch eine Funktion c' höher gewichtet. In diesem Verfahren kommen nun zwei Komponenten zusammen:

1. POEM Überfüllungsvorhersage
2. LLP Verkehrsfluss Vorhersagelabels umgebener Sensoren

Ein Fahrzeug, das sich an einer Kreuzung befindet, ist durch die aktuelle Kante $u \in E$ im Graphen lokalisiert. Eine Kreuzung besteht dabei aus einem Knoten $n \in N$, der die Verbindung zwischen der eingehenden, aktuellen Kante und mindestens zwei ausgehenden anderen Kanten $v_1, v_2, \dots, v_m \in V$ mit $m \in \mathbb{N}_{\geq 2}$ gewährleistet. Für alle ausgehenden Kanten werden nun mit der Funktion c' neue Kantenkosten durch Überfüllungen und Verkehrsflusslabels bestimmt. Der Überfüllungsgraph \mathcal{G} wird anhand der berechneten Überfüllungen und Verkehrsflusslabels aktualisiert, sodass die Basis für die Routenberechnung mit dem Dijkstra Algorithmus geschaffen ist.

3.1.4 Routenberechnung

Der aktualisierte, gewichtete Überfüllungsgraph \mathcal{G} besitzt nun alle Informationen über die lokale Umgebung des Fahrzeugs. Durch Anwendung des Dijkstra Algorithmus wird eine Route vom aktuellen Punkt zum Zielpunkt berechnet, sodass diese Route dem minimal gewichteten Pfad entspricht. Eine Route ist das wichtigste Werkzeug, um einen Einfluss auf die Verkehrslenkung eines Fahrzeugs zu haben.

Eine Route lässt sich durch eine Liste von Straßen definieren, die von einem Fahrzeug abgefahren werden können. Abstrakt entspricht dies bei Anwendung eines Graphen G auf die Verkehrsumgebung als Straßennetz eines Pfades von Kanten $e_i \in E$. Um die Verkehrslenkung zu beeinflussen, wird der aktualisierte Überfüllungsgraph $\mathcal{G}(E, N, c)$ ausgewählt. Zum Beispiel stellen die Kanten $u, v, w \in E$ durch

$$r_{veh} = [u, v, w] \tag{3.2}$$

eine Route r für ein Fahrzeug veh dar. Dabei sind mehrere Einschränkungen zu beachten, damit eine Route als gültig eingestuft wird. Zum einen müssen zwei Kanten des Straßennetzes mit einander verbunden sein, sodass sie sich einen Knoten teilen, der jeweils Startknoten der einen Kante und Endknoten der anderen Kante ist. Weiterhin darf es sich bei einem Kantenpaar nicht um die gegenüberliegenden Straßenseiten handeln, da dies zu einem zu hohen Aufwand des Straßenwechsels kommen würde. Die Verkehrsregeln wie insbesondere Richtungsvorgaben sind mit der Routenberechnung einzuhalten. Die Position des Fahrzeugs muss sich auf der Kante befinden, die an der ersten Stelle der Route steht. Wichtig ist ebenfalls, dass das Ziel des Fahrzeugs durch die Kante erreicht werden kann, die an der letzten Stelle in der Route auftaucht. Im Beispiel befindet sich das Fahrzeug veh aktuell auf der Kante u und hat ein Ziel, das durch die Kante w erreichbar ist. Abbildung 3.2 zeigt eine mögliche Anwendung der Routenberechnung im Anwendungsgebiet der dynamischen Verkehrslenkung. Dort wird eine Kreuzung von einem Fahrzeug angefahren, sodass eine gültige Route benötigt wird.

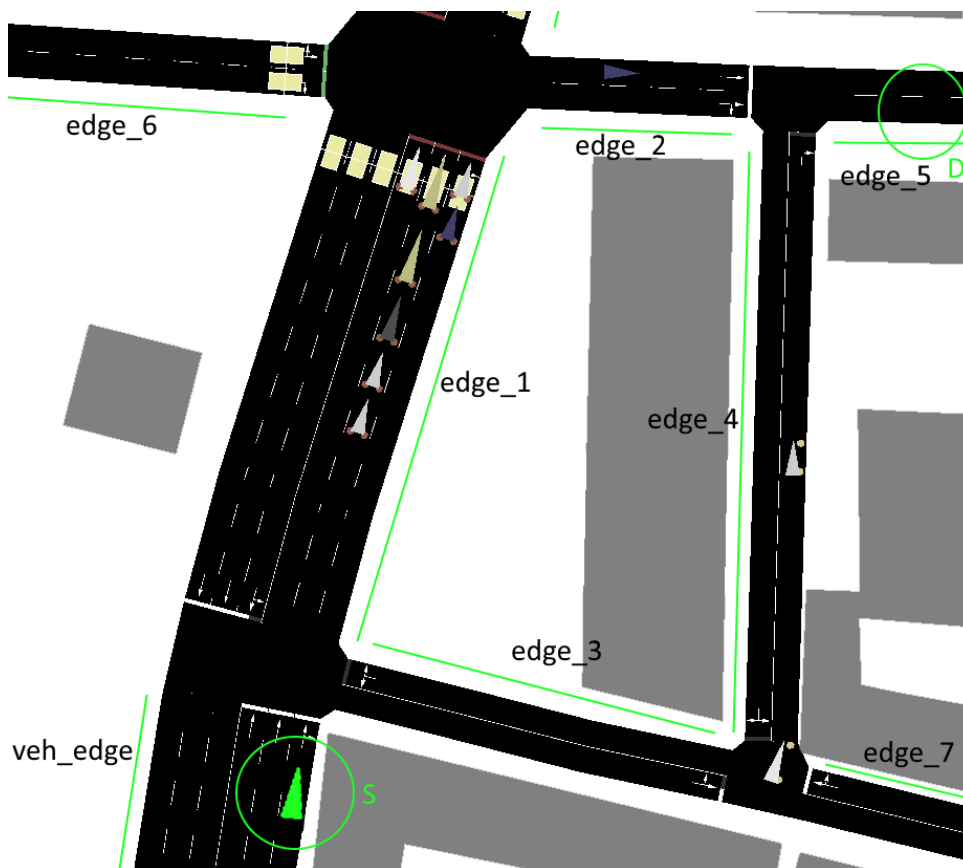


Abbildung 3.2: Ein Fahrzeug befindet sich an Position S und muss eine Route berechnen, um zu Position D zu fahren. Der lokale Eingriff auf die Routenberechnung untersucht nun die Labels der ausgehenden Kanten der nächsten Kreuzung: $edge_1$ und $edge_3$. Beispielsweise kann die Route $[veh_edge, edge_1, edge_2, edge_5]$ ausgewählt werden. Durch die Überfüllung auf $edge_1$ würde jedoch $[veh_edge, edge_3, edge_4, edge_5]$ die erkannte Überfüllung umfahren.

Bei der dynamischen Berechnung von Routen sollte die Problematik von Kreisen innerhalb des Pfades ausgeschlossen sein. Die Anwendung des kürzesten Pfades zum Ziel gewährleistet dies. Durch eine neue Berechnung einer Route kann es unter Umständen jedoch dazu kommen, dass ein Fahrzeug sein Ziel nicht erreicht, da der zurückgelegte Pfad unabhängig von seiner aktuellen Route wiederkehrende Kreise enthält. Kommt ein Fahrzeug in die unmittelbare Nähe des Zielpunktes, wird der kürzeste Pfad ohne Einfluss von Verkehrsinformationen berechnet, um Kreise definitiv zu vermeiden.

3.2 Verkehrslenkung mit POEM

Die Verkehrslenkung besteht aus Fahrzeugen als Agenten in einer Verkehrsumgebung, die mit Routen von ihren Startpunkten zu ihren Zielpunkten geführt werden. Hierbei lässt sich der Verkehr als ein bestärkendes Lernproblem modellieren, um beispielsweise Verkehrsverengungen zu erkennen. Eine Aktion des bestärkenden Lernens beschreibt die Auswahl einer Straße an einer Kreuzung, um gegebenenfalls von der aktuellen Route abzuweichen, falls diese durch ein aktuelles Verkehrsaufkommen beeinträchtigt ist. Sobald eine Straße ausgewählt wurde, erhält die ausgewählte Straße eine Belohnung, welche die Belohnung der Verkehrsumgebung darstellt. Alle anderen Straßen, die hätten ausgewählt werden können, behalten unbekannte Belohnungen [Sot16]. Dieses Szenario wird ebenfalls als *Multi-Armed Bandit Problem* bezeichnet.

In Anbetracht dieser Eigenschaften fällt die Wahl auf den POEM Lernalgorithmus [SJ15], um lokale Features zur Erkennung von Überfüllungen auf Straßen zu nutzen.

3.2.1 POEM Voraussetzungen

Um POEM einsetzen zu können, wird eine existierende Strategie h_0 benötigt, die später optimiert werden kann, um aus der durch POEM resultierenden Strategie eine Labelvorhersage treffen zu können. Dabei wird eine strukturierte Ausgabe auf Basis einer zufälligen Eingabe gebildet, indem die Wahrscheinlichkeit berechnet wird, dass die Ausgabe zutrifft. Speziell für die Anwendung von POEM auf die Vorhersage von Überfüllungen von Straßen benötigt man nun eine geeignete initiale Strategie h_0 sowie einen Eingaberaum X und einen Ausgaberaum Y . Zusätzlich kommt der kardinale Verlust δ zum Einsatz, um ein Qualitätsmaß für alle Eingabe/Ausgabe Kombinationen zu besitzen.

Für den Eingaberaum gibt es keine Einschränkungen, obwohl eine Wahl in geeigneter Weise das Lernen vereinfacht. Hier sei nun der Eingaberaum $X := [0, 1]^m, m \in \mathbb{N}$ mit $\mathbf{x} = (x_1, x_2, \dots, x_m)^T \in X$ als Featurevektor. Dieser Featurevektor \mathbf{x} kann sich aus unterschiedlichen, durch Sensoren aufgezeichnete Werte zusammensetzen, wie zum Beispiel die Dichte einer Straße, die Belegung, die durchschnittliche Geschwindigkeit einer Straße, die Anzahl der Fahrzeuge oder auch durch die Wartezeit von Fahrzeugen auf einer Stra-

ße. Weiterhin lassen sich auch weitere Features einbringen, die im weiteren Verlauf näher erläutert werden. Die Sensorwerte müssen skaliert werden.

Ein Ausgaberaum, der von POEM genutzt wird, muss strukturierte Ausgaben enthalten. In diesen Fall, der Erkennung von überfüllten Straßen, lässt sich der Ausgaberaum auf das einzige Label beschränken, das eine Aussage darüber gibt, ob eine Straße überfüllt ist oder nicht. Daher fällt die Wahl des Ausgaberaums auf $Y := \{(0), (1)\}$. Hierbei stellt das Label (0) eine Ausgabe für eine nicht überfüllte Straße dar.

Die Strategie $h_0(Y|\mathbf{x})$ beschreibt die Wahrscheinlichkeitsverteilung über dem Ausgaberaum. Da diese letztendlich durch POEM optimiert wird, fällt die Wahl standardmäßig auf eine Gleichverteilung, da eine Aussage über das Verhältnis von Eingabe und Ausgabe zunächst kaum möglich ist. Dies ist ein Vorteil bei der Anwendung von POEM, da sehr einfach eine initiale Strategie gewählt werden kann. Folglich sei $h_0(y|\mathbf{x}) := 0,5$ sodass es zunächst gleich wahrscheinlich ist, dass eins der beiden Label ausgewählt wird.

Um die Voraussetzungen für POEM zu vervollständigen, wird ein kardinaler Verlust $\delta : X \times Y \rightarrow \mathbb{R}$ benötigt. Die Erkennung von Überfüllung auf Straßen könnte durch primitive Metriken gewährleistet werden. Beispielsweise die durchschnittliche Geschwindigkeit mit einem Schwellwert, der unterschritten werden kann oder die Dichte einer Straße, die ebenfalls als Schwellwert verwendet werden kann, können eingesetzt werden. Auch der kritische Fluss eines Bottlenecks kann durch den Ersatz als Schwellwertproblem genutzt werden. Somit würde sich der Wertebereich reduzieren: $\delta : X \times Y \rightarrow \{-1, 0\}$.

3.2.2 POEM Anwendung

Es gibt zwei Einsatzgebiete für POEM, um Einfluss auf das Verkehrsrouting zu haben. Einerseits können Label für Straßen bestimmt werden, die aus einer Strategie h abgeleitet werden und es kann ein zukünftiges Überfüllungsgewicht einer Straße berechnet werden, indem Kanten im Überfüllungsgraphen betrachtet werden.

Zunächst wird für die Anwendung von POEM ein Straßengraph $G = (N, E, c)$ benötigt. Jedes Fahrzeug besitzt eine eigene Strategie und lernt diese auf dem Weg, indem ein Log aufgebaut wird. Ein größerer Log bedeutet eine größere Datengrundlage für POEM. Initial erhält jedes Fahrzeug eine Datengrundlage aus der Logging-Phase vor der Anwendung von POEM.

Die Strategie h setzt sich dabei aus $h \in \mathcal{H}_{lim} \cup \{h_0\}$ zusammen, wobei h_0 mit dem Wert 0.5 initialisiert wird. Als Features werden dabei die Features der aktuellen Route, Umgebungsfeatures, Kritischer Fluss und der Dijkstra Rang verwendet. Diese werden in 3.2.3 genauer beschrieben.

Durch eine Funktion c' lässt sich nun aus den Wahrscheinlichkeiten für Überfüllung, die mit der Strategie h berechnet werden, das zukünftige Gewicht einer Kante berechnen. Dabei werden zwei Kanten e_1 und e_2 betrachtet, sodass sich

$$c' : E^2 \rightarrow \mathbb{R}_+, (e_1, e_2) \mapsto \frac{c(e_2)}{h((0)|\frac{e_1}{2}) + h((0)|\frac{e_2}{2})} \quad (3.3)$$

berechnen lässt.

Anhand der berechneten Strategie zur Vorhersage von Überfüllungslabels können nun Gewichtungen von Straßen mit c' berechnet werden. Ein Fahrzeug muss an jeder Kreuzung entscheiden, wo es lang fahren wird. Die umgebenden Straßen erhalten jeweils eine neue Gewichtung. Mit diesen Gewichtungen wird nun ein Überfüllungsgraph \mathcal{G} aktualisiert, sodass eine erneute Anwendung von Dijkstra durchgeführt werden kann. Als Ergebnis erhält jedes Fahrzeug eine eventuell neue Route, sodass gegebenenfalls überfüllte Straßen umfahren werden.

3.2.3 Logging von lokalen Features

Sobald ein Fahrzeug auf seiner Route zum Zielpunkt eine Kreuzung anfährt, kommt es zum Austausch mit den Verkehrssensoren. Hieraus lassen sich viele Informationen und insbesondere lokale Features berechnen. Diese lokalen Features bilden zusammen einen Vektor aus Datenpunkten, der für jedes Fahrzeug auf seiner Route geloggt wird. Jedes Fahrzeug beginnt seine Route mit einem Datensatz aus bereits geloggt Datenpunkten anderer Fahrzeuge und wird erweitert durch Anfahren weiterer Kreuzungen auf einer Route eines Fahrzeugs.

Der angewendete Lernalgorithmus POEM ist speziell dafür gedacht, geloggte Daten zu verwenden. Um im vordefinierten Setting zu funktionieren, wird ein Datensatz D mit der Größe n genutzt:

$$D := \{(\mathbf{x}_i, y_i, \delta_i, p_i) | i \in \mathbb{N}_{\leq n}\}, p_i = h(y_i | \mathbf{x}_i) \quad (3.4)$$

Der Featurevektor \mathbf{x}_i setzt sich aus mehrere lokalen Features zusammen und ist erweiterbar. Hier findet die Anwendung der Features der aktuellen Route, Umgebungsfeatures, kritischem Fluss und dem Dijkstra Rang statt. Nachdem der Featurevektor zusammengesetzt wurde, wird das Label y_i bestimmt. Das entsprechende Label y_i wird mit der dazugehörigen Wahrscheinlichkeit p_i mit der folgenden Gleichung bestimmt:

$$y_i = \begin{cases} (0), h((0)|x_i) > 0,5 \\ (1), h((1)|x_i) > 0,5 \\ random((0), (1)), sonst \end{cases} \quad (3.5)$$

mit $random((0), (1))$ als zufällige Wahl eines Labels. Weiterhin wird die Belohnung als δ_{Dichte} , $\delta_{Geschwindigkeit}$ oder $\delta_{C_{max}}$ hinzugefügt.

Nachdem der Datensatz in der Logging-Phase generiert wurde, können alle Kanten auf Überfüllung untersucht werden, die im gewichteten Graphen auf Basis des Verkehrsnetzes aktualisiert werden. Nun kann durch die Implementierung von Dijkstra eine Route für jedes Fahrzeug berechnet werden, die den kürzesten Weg zum Zielpunkt darstellt. In POEM wird initial die initiale Policy h_0 genutzt und alle Gewichtungen um einen Faktor 2 erhöht, um das Lernen der zuvor gesammelten Daten zu vereinfachen. Dabei wird Dijkstra nicht beeinflusst und es entstehen keine Änderungen an den berechneten Verkehrslenkungsentscheidungen.

Feature der aktuellen Route

Die aktuelle Route und somit die bisher befahrenen Straßen der Route besitzen Informationen über ihre aktuelle Verkehrsdaten und es lässt sich somit pro angefahrenem Sensoren ein neuer Datenpunkt generieren. Die Informationen bilden ein Feature, um an einer Kreuzung zu entscheiden, welcher Weg ausgewählt wird. Sei beispielsweise die bisher ausgefahrene Route $r = [r1, r2, r3, r4, r5]$ und die Sensordaten der einzelnen Kanten bekannt, dann bilden

$$routedata_m = [Sensordata_{r1}, Sensordata_{r2}, Sensordata_{r3}, Sensordata_{r4}, Sensordata_{r5}] \quad (3.6)$$

die Datenpunkte der aktuellen Route.

Umgebungsfeature

Um an einer Kreuzung zu entscheiden, welcher Weg zu wählen ist, spielt die Umgebung eine wichtige Rolle. In diesem Feature werden die Verkehrsinformationen der ausgehenden Kanten an einer Kreuzung zusammengefasst. Besondere Gewichtung dieser Kanten erhalten diese durch mögliche Routen zum Zielpunkt. Das Feature wird durch eine Sortierung der Kanten erweitert, um Vergleichbarkeit von Kreuzungen zu erhalten. Ähnlich zum Feature der aktuellen Route werden die Messungen der Sensoren der umgebenden Kanten $n1$, $n2$ und $n3$ durch

$$arounddata_m = [Sensordata_{n1}, Sensordata_{n2}, Sensordata_{n3}] \quad (3.7)$$

als Umgebungsfeature verwendet werden.

Kritischer Fluss

Der kritische Fluss C_{max} ist ein Schwellwertproblem, das für jede Straße, die auf eine Kreuzung trifft, berechnet werden kann. Somit erhält jede Kante ein Label, ob C_{max} überschritten ist oder nicht. Hierbei werden alle Kanten als *Bottlenecks* gesehen und durch

Überschreitung des Schwellwertes als nicht befahrbar gekennzeichnet. Es existieren ebenfalls Kanten, die keinen kritischen Fluss besitzen, da bei der Messung kein Breakdown aufgetreten ist. Dies wird abgefangen, indem aus der Simulation ein Wert berechnet wird, der höher liegt als der maximal gemessene Fluss dieser Kante.

Dijkstra Rang

Durch den Dijkstra Rang lässt sich die Vielseitigkeit einer Kreuzung ausdrücken, die ein Fahrzeug angefahren hat. Das Dijkstra Rang Feature lässt sich an jeder Kreuzung berechnen. Durch dieses Feature werden Informationen über die Lokalität eines Datenpunktes ermöglicht, da ein Bezug zur aktuellen Position hergestellt wird.

Der Dijkstra Rang nutzt den initialen Startpunkt des Fahrzeugs sowie das Ziel des Fahrzeugs. Er kann nun als Feature auf der Fahrt eingesetzt werden, da sich das Fahrzeug von Startpunkt entfernt und sich dem Zielpunkt nähert. Im Verlauf der Fahrt wird folglich ein höherer Dijkstra Rang erwartet. Der Dijkstra Rang beschreibt dabei die umliegenden Kanten.

3.2.4 Datensatz mit lokalen Features für bestärkendes Lernen

Für das bestärkende Lernen mit POEM entwickelt sich durch die dezentrale Verkehrslenkung ein Datensatz. Dieser setzt sich pro Datenpunkt aus einer gewählten Aktion und den vorliegenden Features zusammen. In diesem Fall werden zunächst aus einer Logging-Phase Daten erfasst und auf dem Weg pro angefahrenem Sensor erweitert. Datenpunkte der Sensoren bestehen dabei aus der Dichte, Belegung, Durchschnittsgeschwindigkeit und Wartezeit.

Bei der dezentralen Verkehrslenkung mit POEM sieht der Datensatz zunächst so aus wie in Tabelle 3.1. Es werden die Features der aktuellen Route, die Umgebungsfeatures, der kritische Fluss und der Dijkstra Rang mit der gewählten Aktion verknüpft.

	Route	Around	C_{max}	DijkstraRang
Action₁	<i>routedata₁</i>	<i>arounddata₁</i>	C_{max_1}	<i>DijkstraRang₁</i>
...
Action_n	<i>routedata_n</i>	<i>arounddata_n</i>	C_{max_n}	<i>DijkstraRang_n</i>

Tabelle 3.1: Datentabelle von n Datenpunkten zu unterschiedlichen Zeitpunkten für POEM. Eine Aktion wird mit lokalen Features verknüpft.

3.3 LLP Vorverarbeitung

Um den LLP Algorithmus anwenden zu können, sind einige Voraussetzungen notwendig. Grundsätzlich existiert für die Anwendung in einer Verkehrsumgebung ein verteiltes Netz

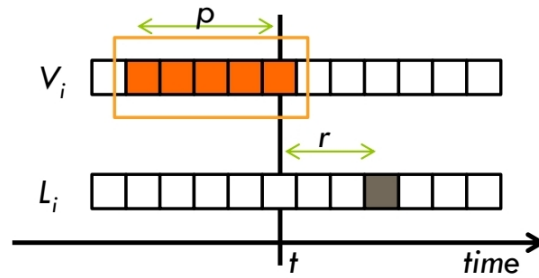


Abbildung 3.3: LLP Datenvorverarbeitung. Quelle: [LSM15]

aus Verkehrssensoren. Diese Verkehrssensoren haben eine bestimmte und feste Position und kennen ihre Verbindungen zu benachbarten Verkehrssensoren.

Zusätzlich zu den Verkehrssensoren gibt es Zeitpunkte t und Knoten P stellen Sensoren im Sensorenetz dar. Ein bestimmter Knoten P_i ist umgeben von mehreren Nachbarknoten. Dabei ist

$$N^{(i)} = \{n_1^{(i)}, \dots, n_c^{(i)}\} \quad (3.8)$$

eine Menge von c Nachbarknoten um P_i . Es sind zu jedem Knoten alle Nachbarknoten bekannt und diese haben die Möglichkeit, miteinander zu kommunizieren. Jeder Knoten besitzt Messungen V_i . In Abbildung 3.3 wird gezeigt, wie in der Vorverarbeitung aus den Messungen V_i eines Sensors ein Fenster x_t mit der Größe p erstellt wird. Jedem Fenster $x_t^{(i)}$ wird nun ein Label $y_{t+r}^{(i)}$ zugeordnet.

3.4 LLP Trainingssetting

Aus der Vorverarbeitung für LLP existieren Datensätze D_i und zugehörige Label L_i . Basierend auf diesen Datensätzen D_i und Labels L_i wird ein lokales Modell $f^{(i)}$ für den Knoten P_i gelernt. Im nächsten Schritt werden die Informationen über Label von P_i an Nachbarknoten versendet. Dies ermöglicht folglich, dass Modelle $f_0^{(i)}$ bei P_0 und $f_{n_1}^{(i)}, \dots, f_{n_c}^{(i)}$ bei den Nachbarknoten trainiert werden. Das Modell $f_{n_e}^{(i)}$ wird im Knoten $P_{n_e}^{(i)}$ gelernt.

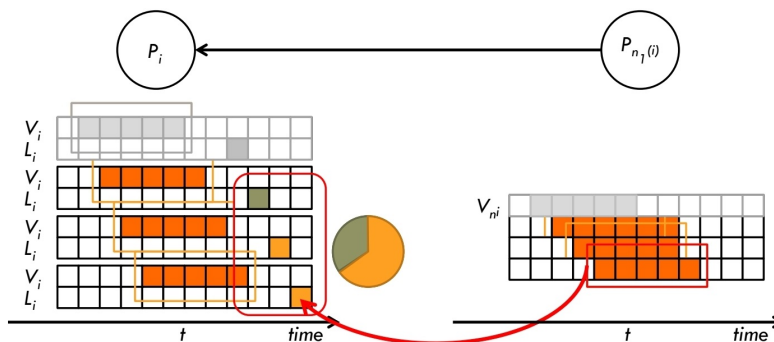


Abbildung 3.4: Aggregation von Labels. Quelle: [LSM15]

Jeder Knoten nutzt seine eigenen Messungen und Label Proportionen. Das gelernte Modell sagt das Label $y_{t+r}^{(i)}$ des Knotens korrekt für den in der Zukunft liegenden Zeitpunkt $t + r$ voraus. Abbildung 3.4 zeigt dabei, wie Label aggregiert werden.

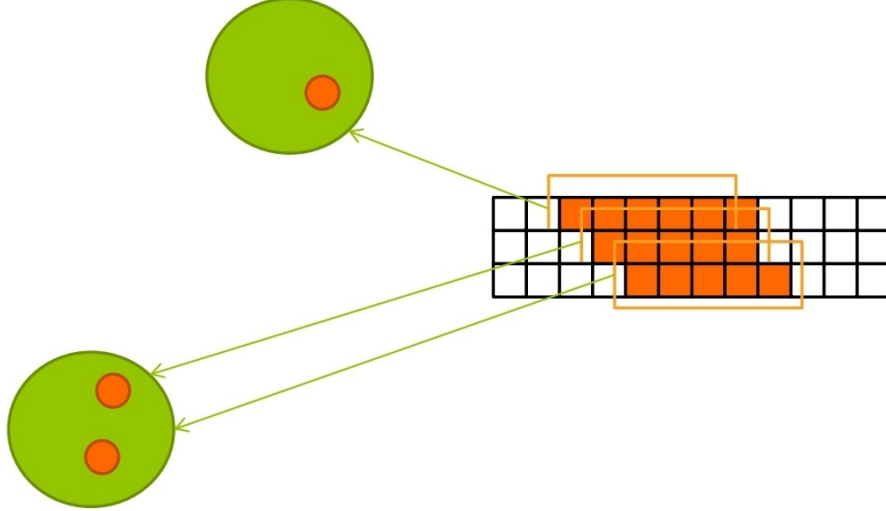


Abbildung 3.5: Cluster Beobachtungen. Quelle: [LSM15]

Das Lernen aus Labelproportionen basiert auf einem Clustering der Label. Beispielhafte Cluster Beobachtungen sind in Abbildung 3.5 dargestellt. Es kommt zu einer Zuordnung der Beobachtungen $x \in X$ zu jedem Cluster C_1, \dots, C_k . Die Zuordnung wird durch $\mu : X \rightarrow \mathcal{C}$ durchgeführt. Dabei wird jede Beobachtung x mit den Clusterzentren durch

$$\mu(x) = \arg \min_{C_k \in \mathcal{C}} \|x - c_k\|^2 \quad (3.9)$$

vergleichen. Zusammen mit $l : \mathcal{C} \rightarrow Y$ mappt $\lambda \in Y$ zu $C \in \mathcal{C}$. Somit ist f eine Komposition $l \circ \mu$. B_i bezeichnet einen Batch $B_i^{n_e^{(i)}}$ und $\Pi^{(i)}$ eine Label Proportion Matrix. Die Einträge γ_{jd} einer Modell-basierten Matrix $\Gamma_f = (\gamma_{jd})$ lassen sich durch

$$\gamma_{jd} = \frac{1}{|B_j|} \sum_{x \in B_j} \mathcal{I}(f(x), Y_d), \mathcal{I} = \begin{cases} 1 : f(x) = Y_d \\ 0 : f(x) \neq Y_d \end{cases} \quad (3.10)$$

berechnen. Letztendlich minimiert LLP den $MSE(\Pi, \Gamma_f)$. Abbildung 3.6 zeigt die Anwendung der mittleren quadratischen Abweichung (MSE) auf eine Darstellung der Label Proportionen Matrix und der Modell-basierten Matrix [LSM15].

Mit LLP ist es nun möglich von jedem Sensor eine Voraussage über das Verkehrsflusslabel zu erhalten. Dabei können verschiedene Zeitpunkte, die in der Zukunft liegen, betrachten werden.

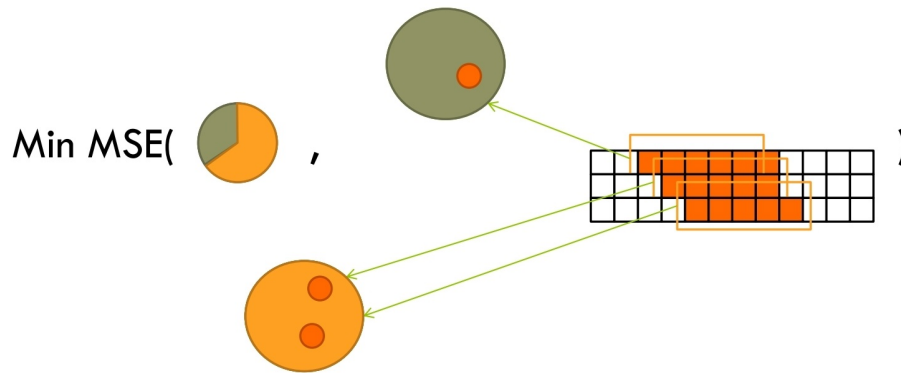


Abbildung 3.6: Label werden zu Clustern zugeordnet. Quelle: [LSM15]

3.4.1 Anwendung von LLP als Feature in Verkehrslenkung mit POEM

Mit der Hilfe von LLP wird die Verkehrslenkung mit POEM durch ein weiteres Feature erweitert. Dabei wird wie in Abbildung 3.7 in der Verkehrsumgebung ein Datenaustausch zwischen Sensor und Fahrzeug ermöglicht, sodass Informationen über den Verkehrsfluss für das Fahrzeug existieren. Der große Vorteil liegt darin, dass durch das LLP Verkehrsflusslabel eine Vorhersage in die Verkehrslenkung mit aufgenommen wird. In Verkehrsumgebungen ist es besonders wertvoll zukünftige Informationen zu verarbeiten, da durch die Bewegung von Fahrzeugen immer wieder neue Zustände eintreten. Das Bewerten einer Straße auf Basis der aktuellen Daten kann zu einem späteren Zeitpunkt nicht mehr nützlich sein.

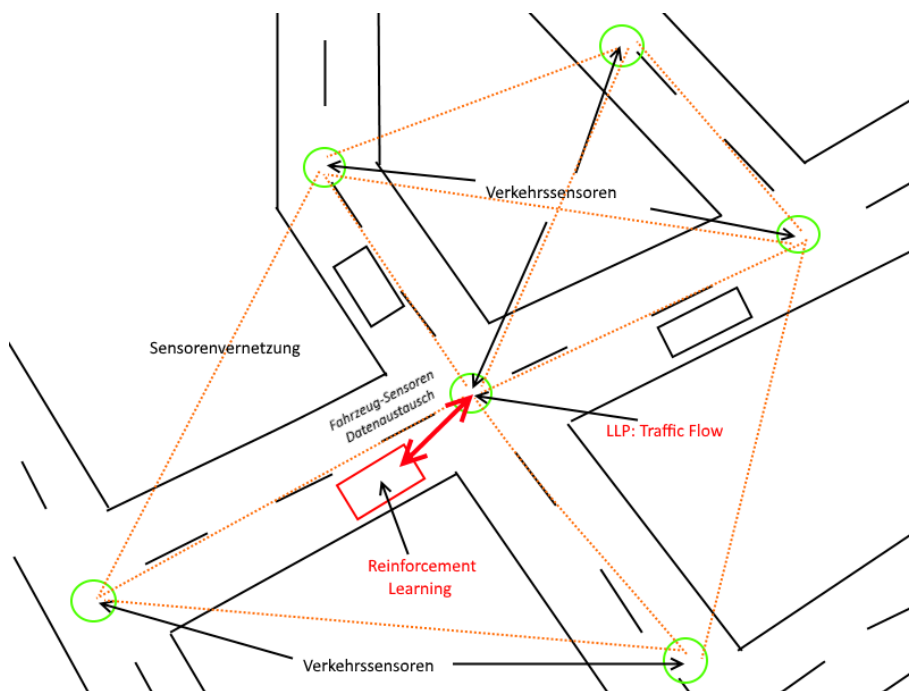


Abbildung 3.7: LLP als weiteres Feature zur Verkehrslenkung.

Eine Möglichkeit die Verkehrsflusslabels, die durch LLP vorhergesagt werden, einzubinden sieht das LLP Verkehrsflusslabel als weiteres Feature für die Verkehrslenkung mit POEM vor. Dabei wird wie in Tabelle 3.2 eine weitere Spalte zum Datensatz hinzugefügt und als zusätzliches Feature verwendet. Somit erhält POEM ein weiteres Feature, das zukünftige Verkehrsflusslabel mit einbezieht. Erkannte Überfüllungen beziehen sich also auf die Entwicklung, die im Straßenverkehr mit der fortlaufenden Zeit vorhanden ist. Somit können entstehende Überfüllungen erkannt werden bevor diese existieren.

	Route	Around	C_{max}	DijkstraRang	LLP-Label
Action₁	$routedata_1$	$arounddata_1$	C_{max_1}	$DijkstraRang_1$	$LLP - Label_1$
...
Action_n	$routedata_n$	$arounddata_n$	C_{max_n}	$DijkstraRang_n$	$LLP - Label_n$

Tabelle 3.2: Der Datensatz kann durch ein $LLP - Label \in \{0, 1, 2, 3, 4\}$ erweitert werden.

Im Gegensatz zur Labelvorhersage mit POEM mit Ausgaberaum $Y_{POEM} = \{0, 1\}$ besteht der Ausgaberaum Y_{LLP} bei der Anwendung von LLP aus $Y_{LLP} = \{0, 1, 2, 3, 4\}$. Also aus fünf Labeln, die den Verkehrsfluss beschreiben, im Gegensatz zum Ausgaberaum von POEM, der aus zwei Labeln besteht. Mit der Labelerkennung durch POEM wird eine Straße als überfüllt oder nicht überfüllt eingestuft.

Es lassen sich im lokalen Ansatz mit POEM nur die umgebenden Straßen der aktuellen Kreuzung betrachten. Dies wird durch den Einsatz von LLP erweitert, indem auch Labelinformationen von Nachbarknoten vorhanden sind, die die Zukunft beschreiben. Anhand des zusätzlichen Features wird nun erneut die Verkehrslenkung mit POEM durchgeführt. Die Verkehrslenkung mit POEM bezieht nun auch zukünftige Verkehrsdaten mit in die Berechnung der Strategie zur Erkennung von Überfüllung mit ein.

3.5 Verkehrslenkung mit LLP Labelvorhersage

LLP kann nicht nur als weiteres Feature bei der Verkehrslenkung verwendet werden. Eine zusätzliche Möglichkeit LLP zur Verkehrslenkung einzusetzen nutzt die Vorhersage der Labelinformation, um diese zur Gewichtung einer Kante im Überfüllungsgraph zu verwenden. Einerseits können Label für Straßen bestimmt werden, die aus der LLP Vorhersage bestimmt werden und es kann ein zukünftiges Überfüllungsgewicht einer Straße berechnet werden.

Es wird für die Anwendung von LLP ein Straßengraph $G = (V, E, c)$ benötigt. Jedes Fahrzeug erhält durch Anfahren eines Sensoren durch das LLP Modell eine Vorhersage

über das Label des angefahrenen Sensors. Diese Vorhersagen werden eingesetzt, um die dynamische Verkehrslenkung direkt zu beeinflussen. Die benötigte Funktion c' wird mit

$$c' : E \rightarrow \mathbb{R}_+, (e) \mapsto c(e) + (c(e) \cdot \frac{LLP(e)}{\#Y_{LLP}}) \quad (3.11)$$

berechnet und kann an jeder Kreuzung jeder Kante ein neues Gewicht zuweisen. Zum Beispiel würde eine Kante e mit erkanntem Verkehrsflusslabel mit $LLP(e) = 2$ und vorheriger Gewichtung $c(e) = 50$ die neue Gewichtung

$$c'(e) = 50 + (50 \cdot \frac{2}{5}) = 70 \quad (3.12)$$

erhalten und somit ein erhöhtes angepasstes Gewicht anhand einer Vorhersage mit LLP besitzen. Die restlichen Gewichte der Kanten werden nicht betrachtet, da diese nicht für das Fahrzeug erreichbar sind. Dadurch wird in der Verkehrslenkung die Entscheidung getroffen, welche nächste Straße ausgewählt wird. Durch die Routenberechnung aus Abschnitt 3.1.4 wird eine neue Route auf Basis der erkannten Überfüllungen berechnet.

Ein Fahrzeug muss an jeder Kreuzung entscheiden, wie die nächste Route aussieht. Die umgebenden Straßen erhalten jeweils eine neue Gewichtung mit c' . Mit diesen Gewichtungen wird nun ein Überfüllungsgraph aktualisiert, sodass eine erneute Anwendung von Dijkstra durchgeführt werden kann. Als Ergebnis erhält jedes Fahrzeug eine neue Route, sodass gegebenenfalls überfüllte Straßen umfahren werden.

Die Funktion c' nutzt die Abstufung eines Verkehrsflusslabels, das durch LLP vorhergesagt wird. Der Ausgaberaum Y_{LLP} besteht aus der Menge $\{0, 1, 2, 3, 4\}$. Das Label (0) wird die Kantengewichtung nicht erhöhen. Erhält die Straße durch die LLP Vorhersage sogar das Label (4), wird die Kantengewichtung verdoppelt, sodass diese Straße durch die Routenberechnung umfahren wird. Es wird also nicht nur erkannt, ob eine Straße überfüllt oder nicht überfüllt sein wird, sondern zusätzlich eine Abstufung mit hereingebracht.

Im Vergleich zur Verkehrslenkung mit POEM werden hier gelernte Verkehrsflusslabel genutzt, die durch das Anfahren von Sensoren erreichbar sind. Es handelt sich ebenfalls um eine Form der dezentralen dynamischen Verkehrslenkung, die das Ziel hat Überfüllungen zu umfahren. Es wird ebenfalls versucht einen Vorteil durch die Vorhersage mit LLP zu erzielen. Die Vorhersagen der Verkehrsflusslabel sind der einzige Einfluss auf die dynamische Verkehrslenkung. Im Gegensatz zur Verkehrslenkung mit POEM müssen keine Daten auf dem Weg geloggt werden.

Es besteht ebenfalls die Möglichkeit diese alternative Nutzung von LLP als Labelerkennung mit der Labelerkennung durch POEM zu erweitern. Es ist ein Versuch die beiden erkannten Label zu kombinieren. Dabei wird eine Erkennung auf Basis aktueller Verkehrsdaten mit einer Vorhersage kombiniert. Dazu wird erneut eine andere Generierung einer Funktion c' benötigt. Betrachte eine Kante e und verwende die Kombination aus einer La-

belvorhersage $LLP(e)$ mit LLP und eine Labelvorhersage $POEM(e)$ mit POEM, damit

$$c' : E \rightarrow \mathbb{R}_+, (e) \mapsto c(e) + \frac{(c(e) \cdot \frac{LLP(e)}{\#Y_{LLP}}) + (c(e) \cdot POEM(e))}{2} \quad (3.13)$$

das zukünftige Gewicht der Kante e bestimmt.

Das nachfolgende Kapitel evaluiert und vergleicht die vorgestellten Möglichkeiten der Verkehrslenkung in Experimenten miteinander.

Kapitel 4

Experimente und Evaluation

Dieses Kapitel behandelt die durchgeführten Experimente im Rahmen dieser Arbeit. Dabei werden die Ergebnisse evaluiert und miteinander sowie mit Experimentenergebnissen eines auf einem zentralen Ansatz basierenden Konzepts verglichen.

4.1 Experimentenumgebung

Als Experimentenumgebung fungiert der Verkehrssimulator *Simulation of Urban Mobility* (SUMO) [BBEK11]. Hierbei handelt es sich um einen Verkehrsnetzsimulator, der durch das Deutsche Institut für Verkehrssystemtechnik entwickelt wurde. Es wird ein Simulator eingesetzt, um Wiederholbarkeit mit gleichen Einstellungen und Voraussetzungen zu erhalten. Dadurch werden Fehler in Experimenten minimiert. Dabei gibt es ein Traffic Control Interface (TraCI) [WPR⁺08], um Einfluss auf die Simulation zu haben. TraCI hat Zugriff auf alle aktiven Fahrzeuge der Simulation und kann viele Informationen über die Straßen eines Szenarios auslesen. Die Straßen werden dabei als *Edges* bezeichnet. Es können Werte von Objekten der Simulation ausgelesen und manipuliert werden. Insbesondere lassen sich Routen von Fahrzeugen anpassen.

Für die Anwendung von Verkehrslenkung in SUMO ist interessant, dass es in Szenarien individuelle Fahrzeuge gibt, die jeweils Routen besitzen. Diese Routen existieren bereits vor der Simulation.

4.1.1 Datensatz

Als Verkehrsdatensatz für SUMO wird die Karte der Stadt Luxemburg genutzt. Dieser Datensatz wird als *Luxembourg SUMO Traffic Scenario* LuST [CFFE17] bereitgestellt und basiert auf einer Nachbildung der Stadt Luxemburg. LuST steht als Open Simulation Szenario zur Verfügung. Es ist möglich, das Verkehrsverhalten von 24 Stunden Mobilität in der Stadt zu reproduzieren. Die Verwendung des LuST Datensatzes ermöglicht einen Vergleich von Experimenten aus [Sot16], da das selbe Szenario genutzt wurde. Abbildung

4.1 zeigt das in SUMO geöffnete Szenario der Stadt Luxemburg. Im LuST Verkehrsnetz gibt es nach Umwandlung in einen Graphen 2373 Knoten, die durch 5969 Kanten verbunden sind. Dadurch entstehen 1571 Kilometer an fahrbaren Straßen auf einer Fläche von 155 km².

In den Experimenten wird das LuST Szenario verwendet und startet um 7:45 für ungefähr 35 Minuten, welche 2048 Sekunden oder Simulationsschritte entsprechen. Die Auswahl dieses Zeitintervalls für die Experimente resultiert aus der Forderung, möglichst viele Fahrzeuge am Straßenverkehr teilnehmen zu lassen. Deswegen wird diese *Rush Hour*, die sich durch erhöhtes Verkehrsaufkommen auszeichnet, ausgewählt.

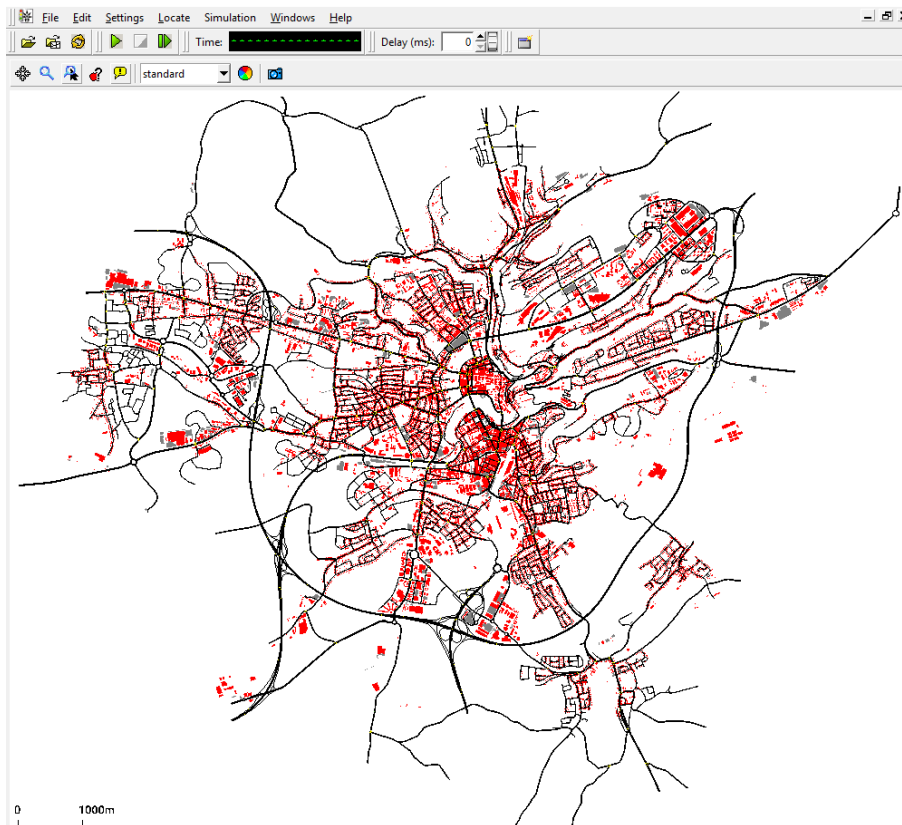


Abbildung 4.1: Simulation der Stadt Luxemburg als SUMO Szenario.

4.2 Evaluationsmetriken

Es sind mehrere Evaluationsmetriken nötig, um die Experimente miteinander vergleichen zu können. Dafür werden Evaluationsmetriken ausgewählt, die Auswirkungen der Anwendung von Verkehrslenkung auf den Verkehr möglichst gut darstellen können. Ein besonderes Augenmerk hat die Verbesserung des Verkehrsflusses im Straßenverkehrs durch eine bessere Verteilung von Fahrzeugen im Verkehrsnetz. Weiterhin soll gemessen werden, dass weniger

überfüllte Straßen in einem Experiment auftreten, als in unbeeinflussten Durchführen des Szenarios.

Edge Wartezeit: Arithmetisches Mittel

Die Edge Waiting Time beschreibt die Wartezeit auf jeder Straße der Simulation. Also die Sekunden, die Fahrzeuge auf den einzelnen Straßen stehen ohne zu fahren. Als Metrik wird der arithmetische Mittelwert aller Straßen während der Experimentendauer gebildet und auf einen Tag hochgerechnet. Diese Messgröße zeigt für den Verkehrsfluss den schlechtesten Fall an, der vermieden werden sollte. Wartezeiten sind ein direkter Hinweis darauf, dass der Verkehrsfluss gestört ist. Denn eine zu hohe Wartezeit hat den Zustand eines Staus zufolge. Somit sind Bewertungen des gesamten Verkehrsnetzes möglich.

Edge Durchschnittsgeschwindigkeit: Harmonisches Mittel

Die Durchschnittsgeschwindigkeit von Fahrzeugen auf den Straßen sagt viel über den Verkehrsfluss aus. Der Indikator $\delta_{Geschwindigkeit}$ als Belohnung für den POEM Lernalgorithmus nutzt genau diese Durchschnittsgeschwindigkeiten von Straßen, um die Bewertung der Verkehrsumgebung zu berechnen. Um einen aussagekräftigen Vergleich zu erhalten, wird die Verteilung der Durchschnittsgeschwindigkeit verwendet.

Das harmonische Mittel \bar{v}_{harm} wird häufig beim Vergleich von Durchschnittsgeschwindigkeiten verwendet, da eine Gewichtung anhand der Straßenlänge möglich ist. In diesen Fall lassen sich die Durchschnittsgeschwindigkeiten v_i von verschiedenen Straßen i , die eine Länge l_i besitzen, vergleichen mit dem anhand der Länge einer Straße gewichteten harmonischen Mittel

$$\bar{v}_{harm} = \frac{\sum_{i=1}^n l_i}{\sum_{i=1}^n \frac{l_i}{v_i}} = \frac{l_1 + \dots + l_n}{\frac{l_1}{v_1} + \dots + \frac{l_n}{v_n}}. \quad (4.1)$$

Edge Dichte: Verteilung

Jede Edge, auf der Fahrzeuge fahren, besitzt eine bestimmte Dichte der Straße. Das Beobachten der Dichteverteilung kann Informationen darüber geben, wie gut Fahrzeuge im Straßennetz verteilt sind und somit ein besserer Verkehrsfluss ermöglicht wird. Bei einer höheren Anzahl von Fahrzeugen in der gesamten Verkehrsumgebung wird sich die Dichte jeder Straße erhöhen. Folglich ist die Betrachtung einer durchschnittlichen Dichte nicht besonders Aussagekräftig. Eine Verteilung der Edge Dichten wird dafür verwendet, um eine Bewertung des Verkehrsverhaltens durch die eingesetzte Verkehrslenkung zu erhalten.

Edge Travel Time: Verteilung

Die Travel Time jeder Edge ist die Zeit, die ein Fahrzeug benötigt, um von Startknoten der Edge zum Zielknoten zu gelangen. Diese Zeit erhöht sich durch die Verkehrsbelastung

auf dieser Straße. Die Travel Times variieren durch die Schwankungen der Fahrzeuggeschwindigkeiten aufgrund von Überfüllungen [Kim17]. Diese Metrik wurde gewählt, um eine Ähnlichkeit zu einer Kostenfunktion c' zu haben, die im Überfüllungsgraphen einen Einfluss auf die Routenberechnung hat.

Fahrzeugfahrzeiten

Die Minimierung der Fahrzeugfahrzeiten stellt ein sehr anschauliches Maß für Verkehrs-routing dar. Jeder Verkehrsteilnehmer hat das Interesse, möglichst schnell zu seinem Ziel zu gelangen. Jedoch ist es in Anbetracht der Verbesserung des Verkehrsflusses nicht ausschließlich das Ziel, die Fahrzeugfahrzeiten zu verringern. Doch umso geringer die Fahrzeugfahrzeiten ausfallen, desto besser funktioniert eine Anwendung von Verkehrslenkung.

Fahrzeugdurchsatz

Der Fahrzeugdurchsatz gibt Aufschluss darüber, wie gut die Verkehrslenkung funktioniert. Das Ziel der besseren Verteilung von Fahrzeugen im Verkehrsnetz ist jedoch ein Konzept, das gegen eine Erhöhung des Durchsatzes spricht. SUMO ist in der Lage, weitere Fahrzeuge an bestimmten Punkten zu generieren, wenn ausreichend Platz vorhanden ist. Umso besser Fahrzeuge von ihrem Startpunkt zum Ziel gelangen, desto höher wird der Durchsatz des Experimentes sein.

UCS und NASH

Zunächst existieren jeweils alle Daten des Simulationsszenarios ohne Eingriff auf die Verkehrslenkung. Ein Vergleichsexperiment berechnet für jedes Fahrzeug zu Beginn eine Route zum Ziel, ohne den Einfluss des Verkehrsaufkommens zu betrachten. Es wird ein *uniform cost search* (UCS) bei der Routenberechnung benutzt, sodass ein Graph genutzt wird, der alle unveränderten Kosten für Straßen im Verkehrsnetz nutzt. Hierdurch wird eine Vergleichsbasis geschaffen als eine untere Schranke. Fahrzeuge würden hier davon profitieren, wenn auf allen Straßen ein freier Verkehrsfluss möglich ist und somit die kürzeste Strecke auch die geringste Zeit benötigt.

Die Spieltheorie *Nash equilibrium* (NASH) [Nas51] kann auf ein Verkehrsnetz angewendet werden und wurde in der Arbeit [Sot16] durchgeführt. Dabei handelt es sich um ein lokales Optimum, wobei Fahrzeuge keinen Vorteil durch das Ändern ihrer Route erhalten können. Als globales Optimum kann das Ändern der Route eines Fahrzeugs in der Verbesserung der Netzwerkperformanz resultieren. Um eine obere Schranke für Experimentenergebnisse zu erhalten, wird NASH [Nas51] verwendet. Es kann somit ein Vergleich zu einem möglichst guten Ergebnis gezogen werden, das in diesem Experiment erreichbar ist. Für NASH existieren jedoch keine Werte für die Durchschnittsgeschwindigkeiten.

In den Experimenten werden die einzelnen Verkehrslenkungen, die auf verschiedenen Konzepten basieren, durchgeführt und die einzelnen Metriken aufgezeichnet und berechnet. Das erste Experiment nutzt eine Verkehrslenkung mit lokalem bestärktem Lernen. Im zweiten Experiment wird das lokale bestärkte Lernen durch Daten der einzelnen Sensoren erweitert, die durch LLP generiert wurden. Ein weiteres Experiment nutzt eine alternative Möglichkeit, um LLP zur Verkehrslenkung einzusetzen. Die Durchschnittswerte werden in einfachen Ergebnisplots miteinander verglichen. Bei den Verteilungen werden Boxplots [MTL78] verwendet. Die Box repräsentiert dabei das untere, mittlere und obere Quartil. Mit den *Whiskers* wird das zweite und 98. Perzentil dargestellt.

4.2.1 Kritischer Fluss Map für LuST

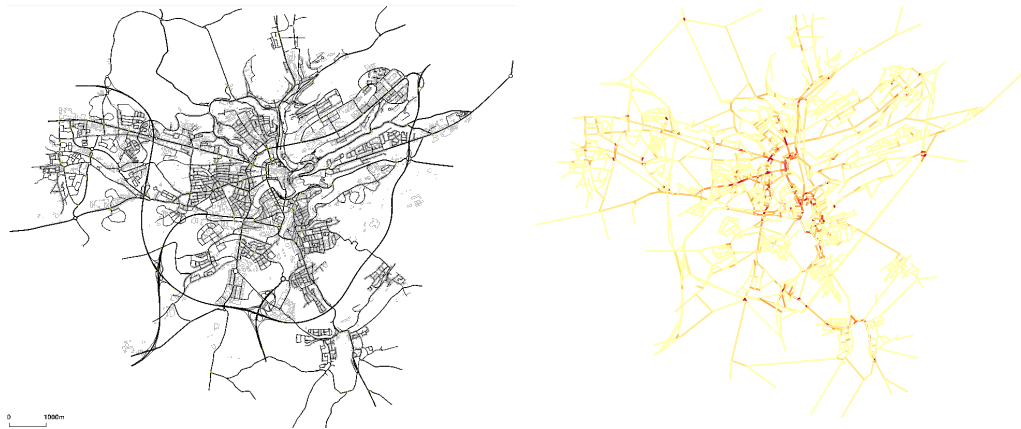


Abbildung 4.2: Links die Stadt Luxemburg aufgerufen in SUMO GUI und rechts eine Map aller Straßen mit Einfärbung der gemessenen kritischen Dichte.

In der Implementierung der Features für POEM wird ein kritischer Fluss C_{max} benötigt. Dieser ist genauer beschrieben in 2.3.3 und 3.2.3. Ein kritischer Fluss ist für jede Straße unterschiedlich, die auf eine Kreuzung trifft. Da der kritische Fluss in der Lage ist, Bottlenecks zu identifizieren, deutet ein geringer Wert für den kritischen Fluss auf eine Einordnung einer Straße als Bottleneck hin. Ein geringer Wert ist durch die Überfüllung einer Straße schneller erreicht als ein hoher Wert. In Abbildung 4.2 sind auf der linken Seite zunächst alle Straßen der Stadt Luxemburg vorhanden. Es ist sehr schwierig, direkt Bottlenecks zu erkennen.

Der kritische Fluss kann nicht berechnet werden, sondern nur durch Messung des Verkehrsflusses von Straßen auf einen bestimmten Zeitraum. Bei der Simulation der Stadt Luxemburg können eben genau diese Messungen durchgeführt werden, sodass C_{max} für jede Straße gemessen wird.

Bei der Messung des kritischen Flusses kommt das Problem auf, dass dieser nur Gemessen werden kann, wenn eine Überfüllung auf einer Straße auftritt. Das bedeutet, dass

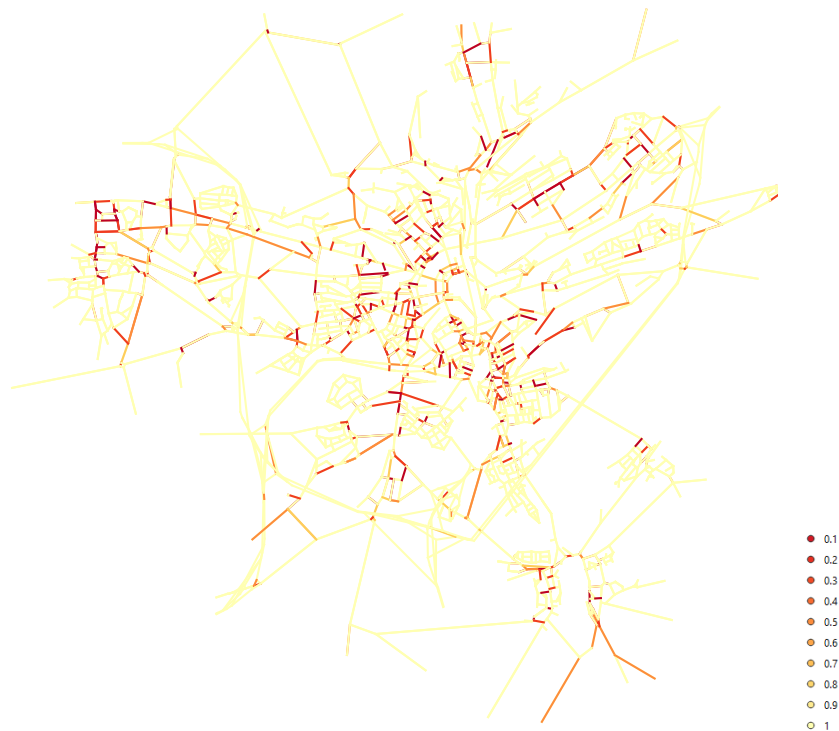


Abbildung 4.3: Map des kritischen Flusses der Straßen in der Stadt Luxemburg. Die Skala normalisiert den kritischen Fluss. Gelb bedeutet einen hohen kritischen Fluss, sodass viele Fahrzeuge pro Zeiteinheit eine Überfüllung auslösen. Rot eingefärbte Straßen besitzen einen geringen kritischen Fluss. Teilweise ist bei gelb eingefärbten Straßen auch kein kritischer Fluss messbar gewesen.

Straßen, die nie überfüllt sind, keinen kritischen Fluss besitzen. In Anbetracht von Verkehrslenkung ist dies jedoch kein großes Problem, da Straßen ohne auftretende Überfüllung auch kein Hindernis zum Zielpunkt darstellen. Im Ergebnis werden nur Straßen abgebildet, für die ein kritischer Fluss berechenbar war.

Abbildung 4.3 zeigt das Ergebnis anschaulich als Map. Es sind anhand der Skala die einzelnen Werte der Straßen eingefärbt. Dabei ist zu erkennen, dass es viele unterschiedliche Ausprägungen gibt und kein Zusammenhang zur Lokalität einer Straße besteht, da jede Straße bei der Messung isoliert betrachtet wurde. Es sind einzelne Bottlenecks vorhanden, die jedoch das Potenzial besitzen, umfahren zu werden. Diese Erkenntnis lässt sich besonders gut durch dezentrale dynamische Verkehrslenkung anwenden. Das Anfahren eines Bottlenecks würde in der Verkehrslenkungsentscheidung einen Weg drumherum wählen und somit ebenfalls ein Bottleneck entlasten.

Ein weiteres Anwendungsgebiet des kritischen Flusses ist $\delta_{C_{max}}$. Diese kann einen Indikator dafür geben, ob eine Straße als überfüllt einzustufen ist oder nicht.

4.3 Verkehrslenkungsexperiment mit lokalen bestärktem Lernen

Die genutzte Verkehrslenkung ist genauer in 3.2 beschrieben und nutzt den POEM Algorithmus. Hierbei handelt es sich um eine Verkehrslenkung, die dynamisch auf Basis von lokalen Daten über die Route eines Fahrzeugs entscheidet. Dabei erhält jedes Fahrzeug Verkehrsdaten von angefahrenen Sensoren in der Verkehrsumgebung. Der POEM Algorithmus wird auf Basis der Daten ausgeführt, die jedes Fahrzeug auf seinem Weg sammelt. Die Belohnung wird dabei durch eine primitive Indikation durch die Durchschnittsgeschwindigkeiten auf Straßen berechnet. Im Experiment werden anhand der Evaluationsmetriken Ergebnisse generiert.

In Abbildung 4.4 ist ein sehr wichtiges Ergebnis für die Verbesserung von Verkehrsfluss durch dynamische Verkehrslenkung erkennbar. Die Wartezeit von Fahrzeugen, die sich auf den Straßen befinden, verringert sich um 200 Sekunden Wartezeit im Experiment. Dies deutet darauf hin, dass sich der Verkehr besser im Verkehrsnetz aufteilt. Die in diesem Experiment gemessene Durchschnittsgeschwindigkeit in Abbildung 4.5 erhöht sich im Vergleich zu UCS. Dieses Ergebnis deutet ebenfalls darauf hin, dass die dynamische Verkehrslenkung einen positiven Einfluss auf die Verteilung des Verkehrs im Verkehrsnetz hat. Die Wartezeit von Fahrzeugen auf den Straßen steht im Zusammenhang mit der Durchschnittsgeschwindigkeit von Fahrzeugen auf diesen. Das Erkennen und Umfahren von überfüllten Straßen hat zu Folge, dass es zu einer besseren Verteilung des Verkehrs im Verkehrsnetz kommt. Überfüllungen lassen sich durch die Dichte von Straßen analysieren. Die Verteilung der Dichte in Abbildung 4.6 zeigt, dass durch die Anwendung der dynamischen Verkehrslenkung weniger Straßen eine hohe Überfüllung erreichen. Die Abbildung 4.6 zeigt einen geringeren Wert für den oberen Whiskas und somit genau die Eigenschaft, dass weniger hoch überfüllte Straßen im Experiment vorkommen. Erkannte Überfüllung wird folglich an den richtigen Kreuzungen erkannt und in der Verkehrslenkung genutzt, um diese zu umfahren.

Ein weiterer Effekt, der durch diese Form der dynamischen Verkehrslenkung auftreten kann, ist die höhere Nutzung von Seitenstraßen und insbesondere von Straßen, die selten benutzt werden. Dieser Effekt ist in Abbildung 4.6 ablesbar, da die Dichten der Straßen zwischen dem oberen und unteren Quartil niedriger sind. Dies ist darauf zurückzuführen, dass Fahrzeuge durch die dynamische Verkehrslenkung ähnliche Routen erhalten. Fahrzeuge, die sich an Kreuzungen befinden, werden bei Überfüllung den gleichen Weg um die Überfüllung herum wählen und somit eine in Teilen ähnliche Route erhalten. Dieser Weg an der Überfüllung vorbei wird so lange eingeschlagen, bis die Überfüllung aufgelöst ist. Festzuhalten ist jedoch, dass sich durch diese Anwendung keine Überfüllungen verlagern, da sich die Fahrzeuge auf die Straßen aufteilen.

Jede Straße besitzt eine bestimmte Travel Time, die abhängig von der aktuellen Verkehrslast ist. Hierbei gibt es ebenfalls sehr lange und kurze Straßen, die ihrer Länge entsprechende Travel Times besitzen. Abbildung 4.7 zeigt die Verteilung der Travel Times auf den Straßen im Laufe des Experiments. Dabei ist zunächst auffällig, dass insgesamt eine kleinere Spanne gemessen wurde.

Die Fahrzeugfahrzeiten und der Durchsatz stehen in einem Zusammenhang, da bei niedriger durchschnittlicher Fahrzeugfahrzeit auch mehr Fahrzeuge ihr Ziel erreichen und somit der Durchsatz steigt. Die eingesetzte Verkehrslenkung mit POEM ist besser als UCS. In den Abbildungen 4.8 und 4.9 ist zu erkennen, dass das Ergebnis des Experiments eine Verbesserung des Verkehrsverhaltens zu Folge hat, aber nicht an das Ergebnis von NASH herankommt. Offensichtlich funktioniert die Verkehrslenkung nicht so gut wie NASH, da dynamische Verkehrslenkung erst auf Überfüllung reagiert, wenn diese auftritt. Dabei wird die dezentrale Erkennung von Überfüllung genutzt.

Jedes Fahrzeug erweitert seinen Datensatz auf der aktuellen Route. Das bedeutet, dass zu Beginn des Experimentes noch nicht ausreichend Daten vorhanden sind, um Überfüllungen zu erkennen. Gerade beim Start einer Route wird durch den kürzesten Weg von vielen Fahrzeugen eine ähnliche Route ausgewählt, um zum jeweiligen Zielpunkt zu gelangen. Das führt beim Start des Experimentes dazu, dass sich Straßen schnell überfüllen. Jedes Fahrzeug erzeugt einen Datensatz, der schnell viele Eigenschaften von Überfüllungen über die verwendeten Features sammelt. Sobald Überfüllungen erkannt werden, wird dynamisch darauf reagiert.

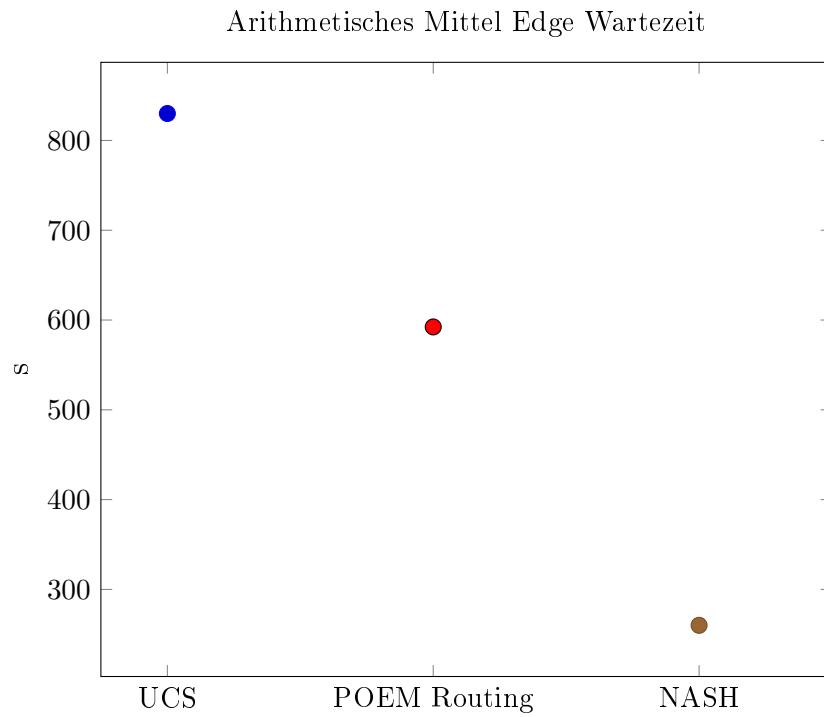


Abbildung 4.4: Ergebnisse der Edge Wartezeit als arithmetisches Mittel.

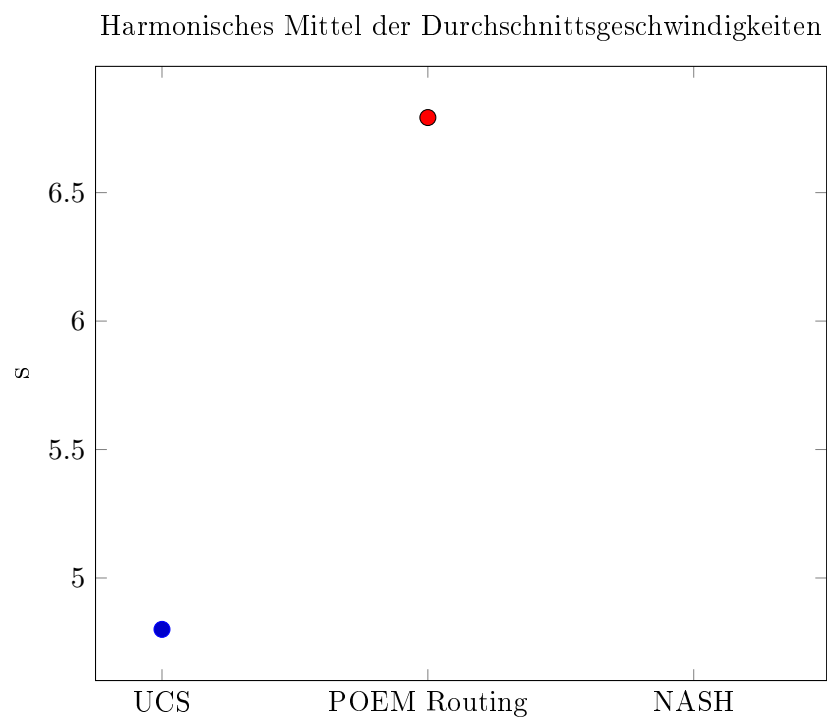


Abbildung 4.5: Ergebnisse der Durchschnittsgeschwindigkeiten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

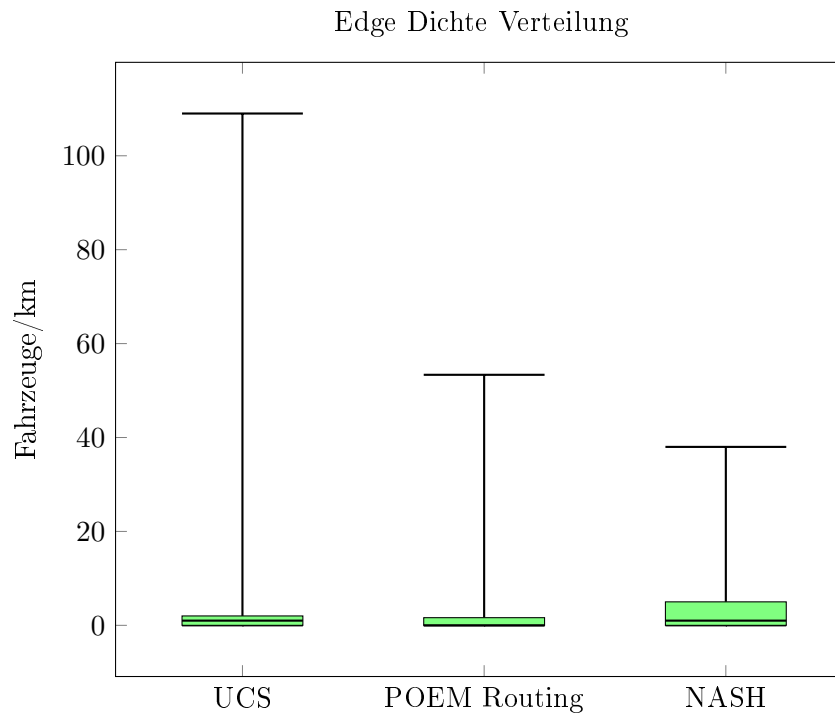


Abbildung 4.6: Ergebnisse der Durchschnittsdichten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

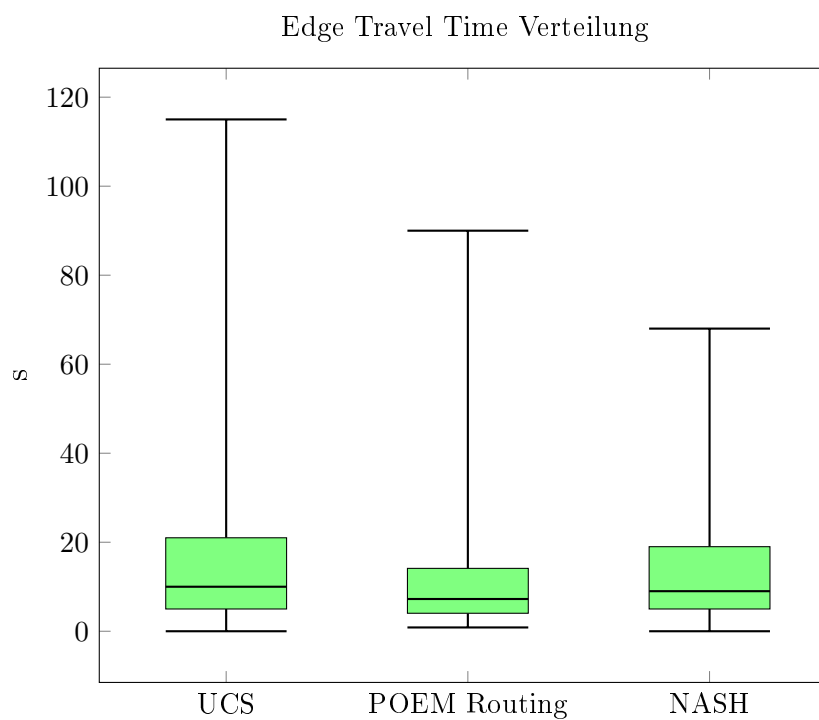


Abbildung 4.7: Ergebnisse der Edge Travel Time Verteilung.

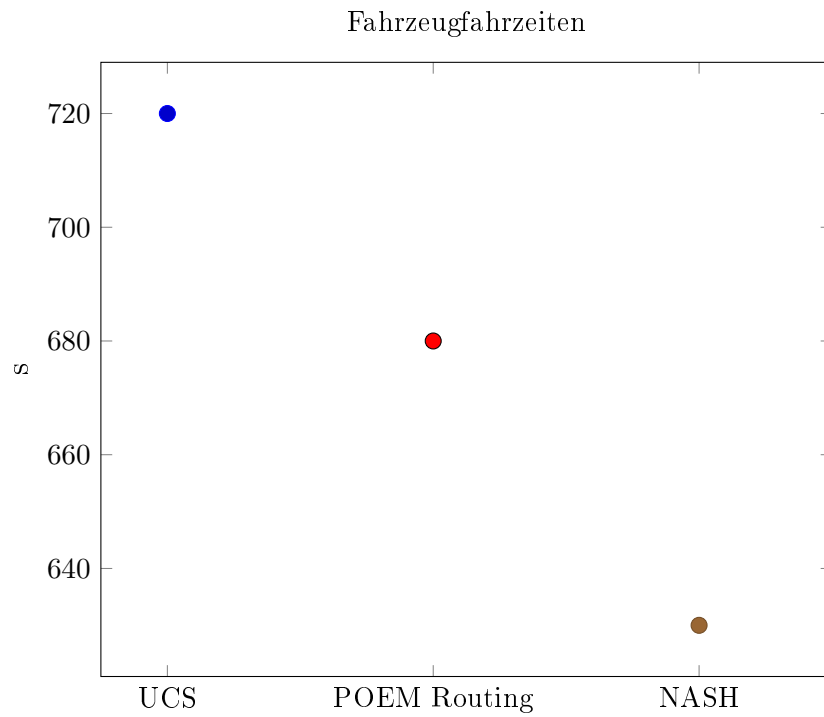


Abbildung 4.8: Ergebnisse der Fahrzeugfahrzeiten, die Fahrzeuge durchschnittlich für ihre Routen benötigen.

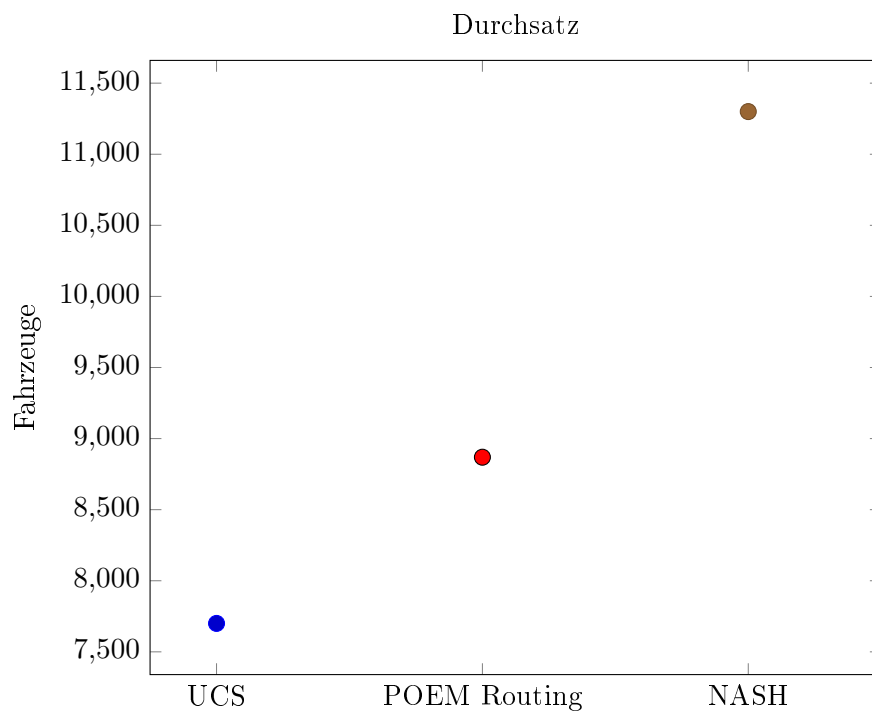


Abbildung 4.9: Ergebnisse des Durchsatzes des Verkehrsnetzes.

4.3.1 Vergleich zu Verkehrslenkung auf Basis zentraler Verkehrsdaten

Die eingesetzte lokale Verkehrslenkung lässt sich mit der Verkehrslenkung, beschrieben in der Arbeit [Sot16], vergleichen. Im Folgenden werden zunächst die Ergebnisse der Verkehrslenkung auf Basis von zentralen Daten vorgestellt und mit den Ergebnissen aus 4.3 verglichen. In 3.2.4 werden die hier verwendeten Features genauer vorgestellt.

In den vorgestellten Metriken erhält die zentrale Verkehrslenkung bessere Ergebnisse als die dezentrale Verkehrslenkung. Der Unterschied besteht darin, dass bei der zentralen Verkehrslenkung die vollständigen Informationen der Verkehrsumgebung zu den einzelnen Zeitpunkten zur Verfügung stehen. Somit sind einerseits mehr Daten für den POEM Lernalgorithmus vorhanden und es können Straßen betrachtet werden, die weit vom Fahrzeug entfernt liegen. Die dezentrale Verkehrslenkung hat hier nur die Möglichkeit, die Informationen der lokal erreichbaren Sensoren zu erhalten.

Die Abbildungen 4.10 und 4.11 zeigen, dass die zentrale Verkehrslenkung ebenfalls Wartezeit verringert und Durchschnittsgeschwindigkeit erhöht, sodass auch hier eine bessere Verteilung der Fahrzeuge im Verkehrsnetz vorhanden ist. Die Verteilung der Dichte in Abbildung 4.12 ist dem Gesamtausschlag nach sehr ähnlich, jedoch sind bei der zentralen Verkehrslenkung mehr Fahrzeuge vorhanden. Abbildung 4.13 zeigt die Travel Time Verteilungen und hat das Ergebnis, dass beim zentralen Ansatz insgesamt eine geringere Spanne gemessen wurde und somit weniger Überfüllungen auftreten. Bei den Fahrzeugfahrzeiten in Abbildung 4.14 und dem Durchsatz in Abbildung 4.15 ist zu erkennen, dass durch zentrale Verkehrslenkung die individuellen Fahrzeuge schneller an ihr Ziel gelangen.

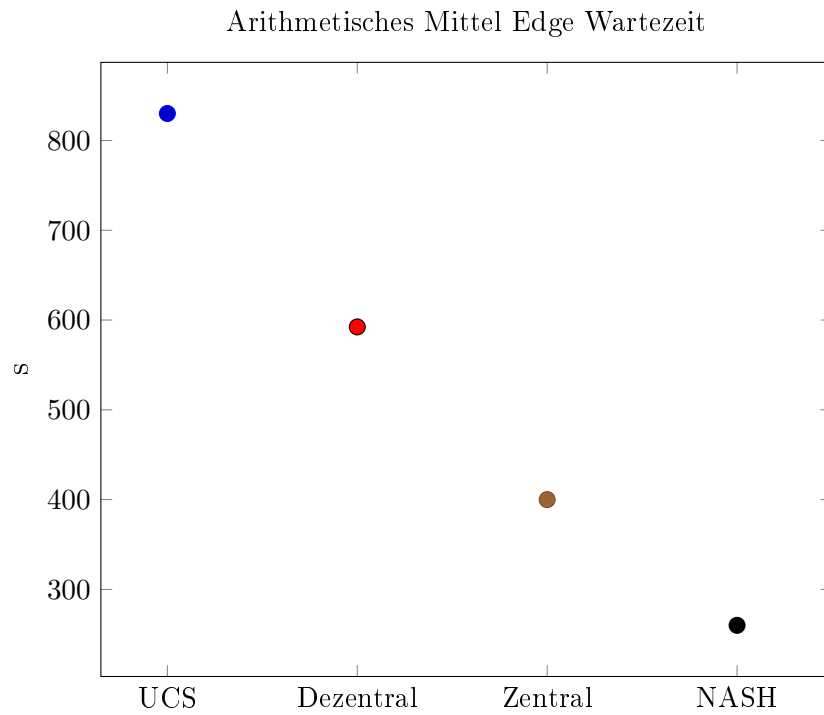


Abbildung 4.10: Vergleich der Edge Wartezeit als arithmetisches Mittel.

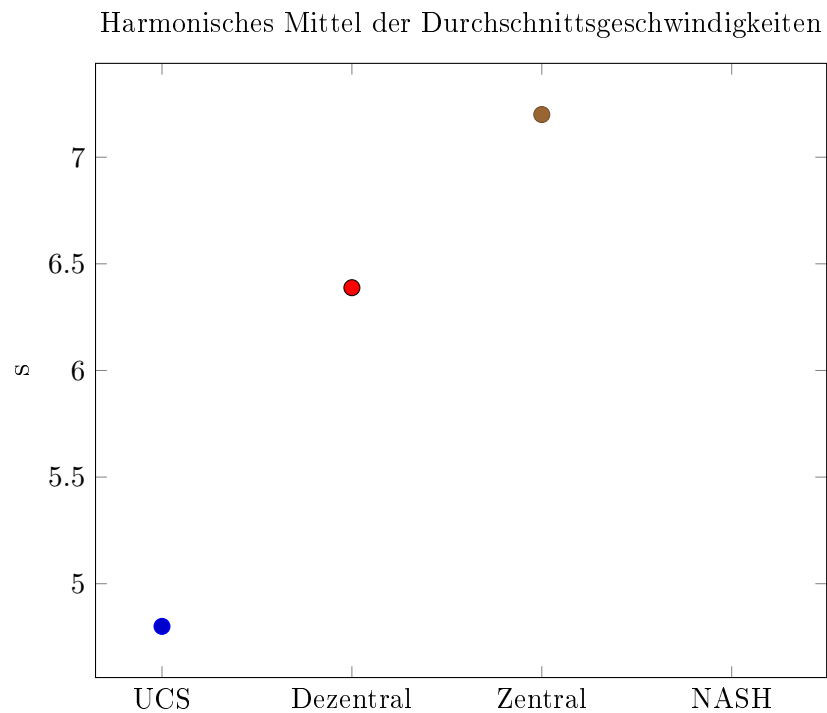


Abbildung 4.11: Vergleich der Durchschnittsgeschwindigkeiten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

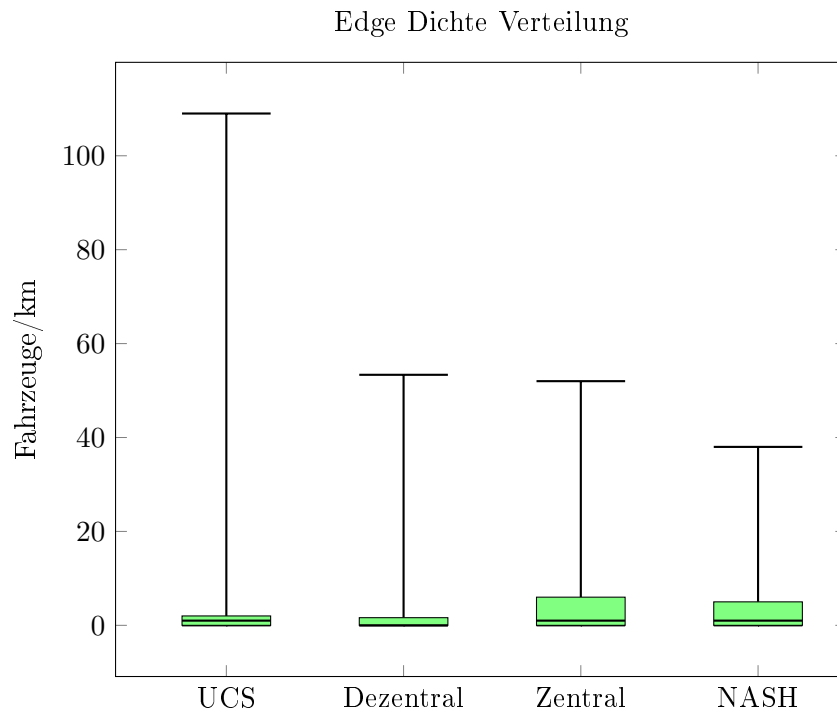


Abbildung 4.12: Vergleich der Durchschnittsdichten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

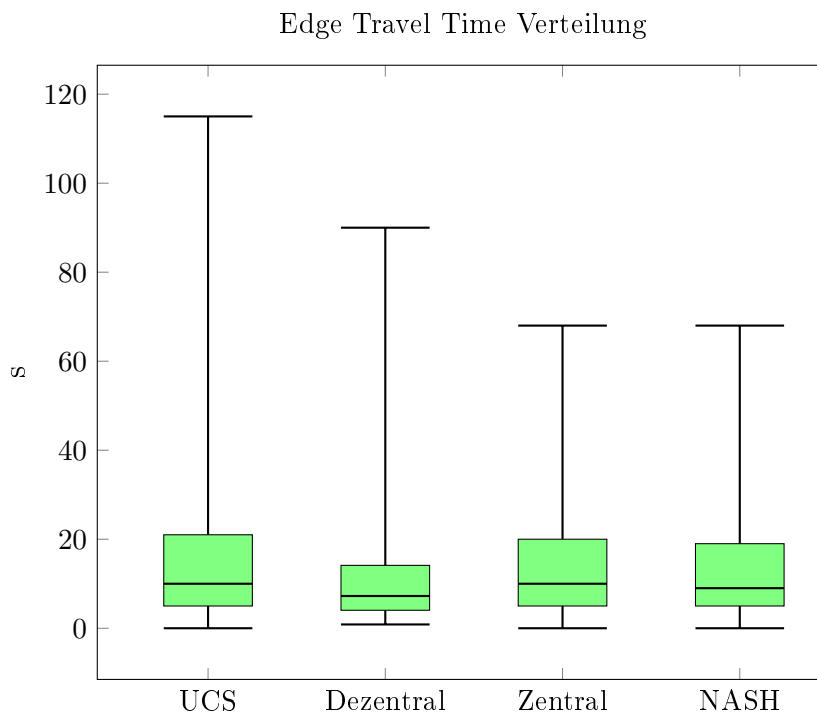


Abbildung 4.13: Vergleich der Edge Travel Time Verteilung.

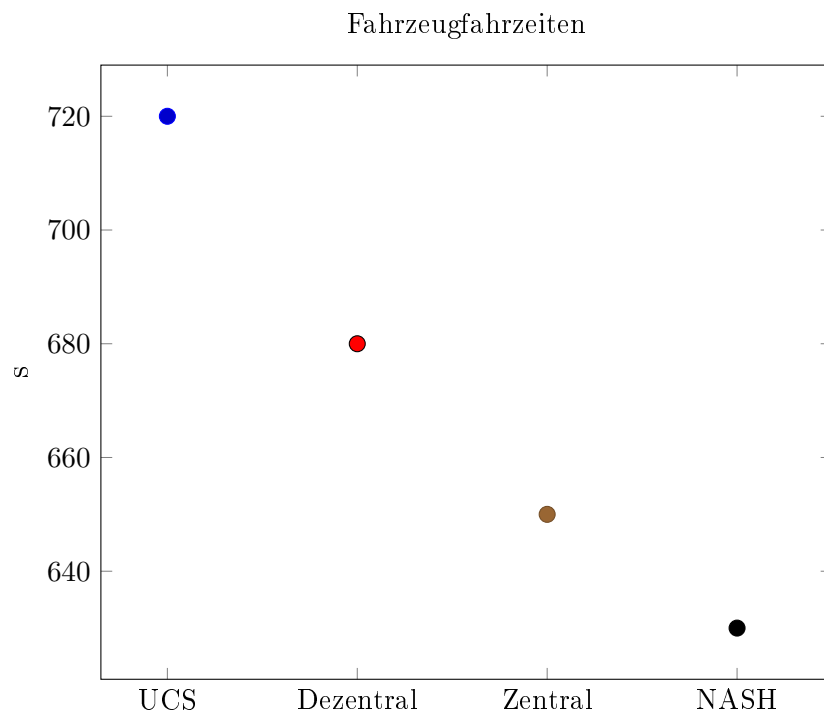


Abbildung 4.14: Vergleich der Fahrzeugfahrzeiten, die Fahrzeuge durchschnittlich für ihre Routen benötigen.

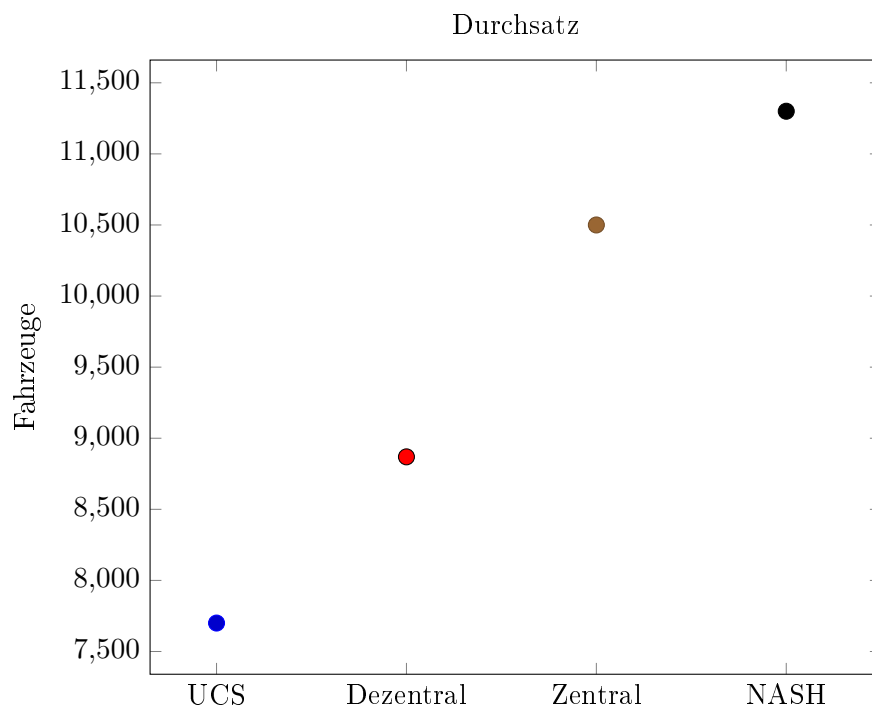


Abbildung 4.15: Vergleich des Durchsatzes des Verkehrsnetzes.

4.4 Verkehrslenkungsexperiment mit lokalen bestärkendem Lernen und zusätzlichem LLP Feature

In diesem Experiment wird das Experiment aus 4.3 mit einem zusätzlichen Feature erweitert und wird in 3.4.1 beschrieben. Dieses Feature hat den großen Vorteil, dass zusätzlich zukünftige Verkehrsflusslabel mit in die Verkehrslenkung aufgenommen werden. Das hat zu Folge, dass überfüllte Straßen erkannt werden können, bevor diese überfüllt sind.

Für die Anwendung von LLP müssen zunächst Sensoren in der Verkehrsumgebung festgelegt werden. Anschließend können Eingabedaten für den LLP Algorithmus generiert werden.

4.4.1 Generierung der Eingabedaten für den LLP Algorithmus

Die Anwendung des LLP Algorithmus benötigt eine Datengrundlage D_i , um aus den Verkehrsdaten Vorhersagen zu generieren. Der Trainingsdatensatz hat die Form:

Zeitpunkte t_i mit $i \in I$ pro Sensor s_j :

$$\{t_i, \text{Sensor-ID}, \#\text{Sensoren}, x\text{-Koordinate}, y\text{-Koordinate}, \text{Stadt}\} \quad (4.2)$$

$$D_i = \{[s_j]_{t_i}, \text{Sensor-ID}, \#\text{Sensoren}, x\text{-Koordinate}, y\text{-Koordinate}, \text{Stadt}\} \quad (4.3)$$

Wie beispielsweise:

$$D_{199} = \{ \underbrace{0; 0; \dots; 0.0464; 0.0463; 0;}_{s_j \text{ pro } t_i} \underbrace{-32370}_{\text{Sensor-ID}} ; \underbrace{1466}_{\#\text{Sensoren}} ; \underbrace{6716.21}_{x\text{-Koordinate}} ; \underbrace{6826.51}_{y\text{-Koordinate}} ; \underbrace{lux}_{\text{Stadt}} \} \quad (4.4)$$

Es existieren folglich für jeden Zeitpunkt des LuST Szenarios und jeden Sensor verschiedene Verkehrsdaten. Anhand der Dichte, die an den Sensoren gemessen wird, existieren zu den Datensätzen D_i auch alle Label L_i .

Für selten angefahrene Sensoren ist eine Anpassung notwendig, da nicht ausreichend Datenpunkte für das Clustering vorhanden sind. Bei einer höheren Anzahl an Clusterzentren als Datenpunkten eines Sensors wird keine LLP Vorhersage vorgenommen.

4.4.2 Evaluation der Ergebnisse mit LLP als POEM Feature

In diesem Experiment kommt es zu der Erweiterung einer dynamischen Verkehrslenkung durch lernende Sensoren. Es wird für jede Straße ein Verkehrsfluss Label vorhergesagt und als Feature im POEM Lernalgorithmus genutzt. Im Vergleich zur dezentralen Verkehrslenkung ohne LLP Feature werden nun Überfüllungen auf der Basis von aktuellen Verkehrsdaten in Kombination mit vorhergesagten Verkehrsdaten erkannt.

In den Abbildungen ist zu erkennen, dass die Verkehrslenkung positiv beeinflusst wird. Insbesondere die Wartezeit sinkt durch die Anwendung des zusätzlichen LLP Features. Dies ist in Abbildung 4.16 zu erkennen. Es kommt zu einem besseren Ergebnis durch die Zunahme von LLP als Feature. Also werden mehr Überfüllungen erkannt und es ist ein positiver Einfluss durch vorhergesagte Verkehrsflüsse zu erkennen.

Die Durchschnittsgeschwindigkeit in Abbildung 4.17 erhöht sich nur leicht und kommt ebenfalls nicht an das Ergebnis der zentralen Verkehrslenkung heran. Die Abbildung 4.18 mit der Verteilung der Dichte deutet auf eine bessere Verteilung der Edge Dichten bei Nutzung des LLP Features hin. Der höchste Ausschlag bei der Nutzung von LLP als Feature ist etwas geringer. Abbildung 4.19 mit der Verteilung der Travel Time zeigt kaum Änderungen.

Die Fahrzeugfahrzeiten der angekommenen Fahrzeuge ist geringer als bei der Verkehrslenkung ohne LLP Feature. Abbildung 4.20 zeigt, dass diese sogar fast so gering sind, wie die Fahrzeugfahrzeiten des zentralen Ansatzes. Dies steht jedoch im Zusammenhang mit dem Durchsatz, der im Vergleich zur zentralen Verkehrslenkung geringer ausfällt. Der Durchsatz in Abbildung 4.21 ist durch die Hinzunahme von LLP als Feature gestiegen.

Grundsätzlich führt die Verkehrslenkung mit bestärkendem Lernen mit zusätzlichem LLP Feature zu einem besseren Ergebnis als ohne LLP Feature. Das Ergebnis hebt sich jedoch nicht soweit ab, dass das Ergebnis einer zentralen Verkehrslenkung erreicht wird. Fahrzeuge werden besser im Verkehrsnetz aufgeteilt, da erkannte Überfüllungen umfahren werden.

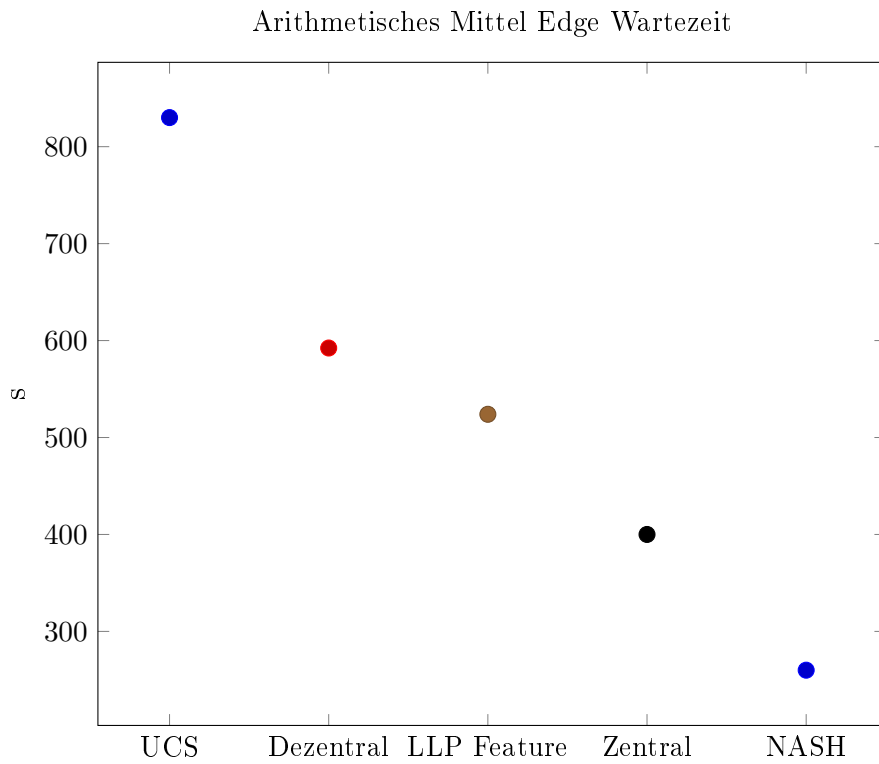


Abbildung 4.16: Ergebnis der Edge Wartezeit als arithmetisches Mittel.

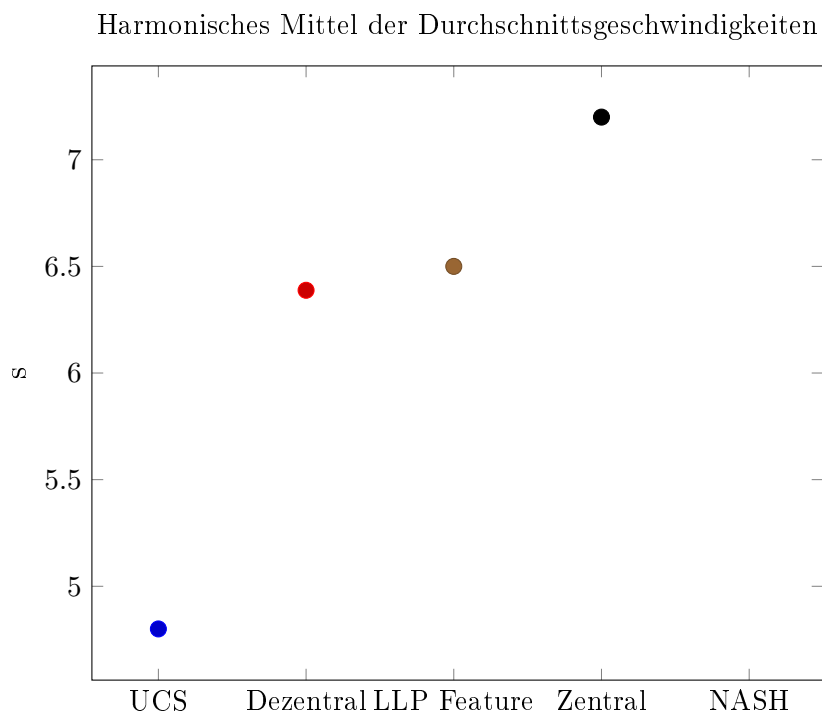


Abbildung 4.17: Ergebnis der Durchschnittsgeschwindigkeiten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

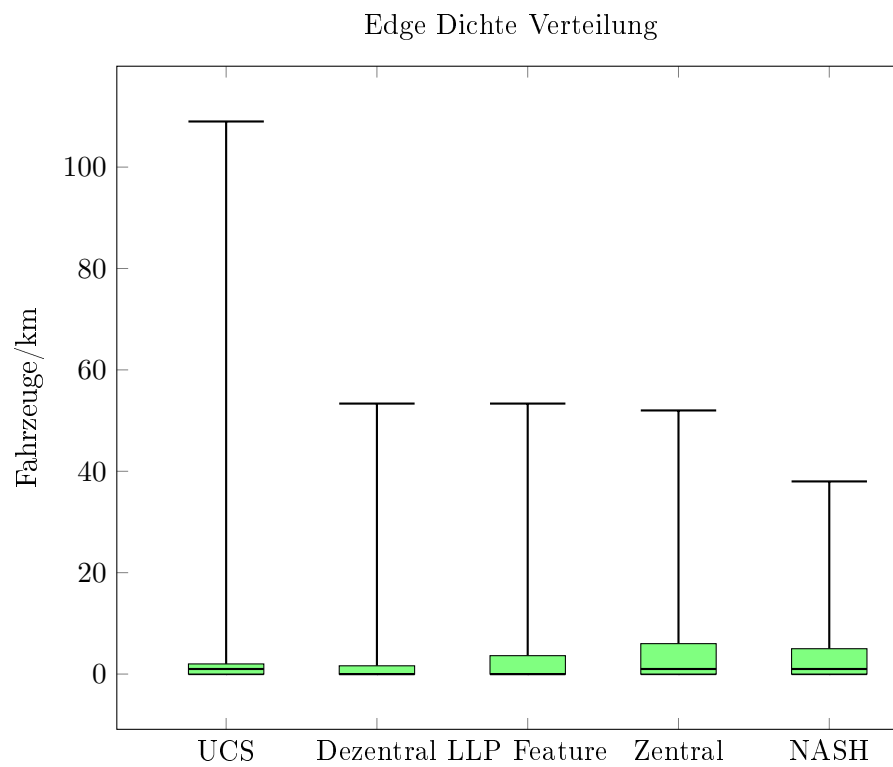


Abbildung 4.18: Ergebnis der Durchschnittsdichten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

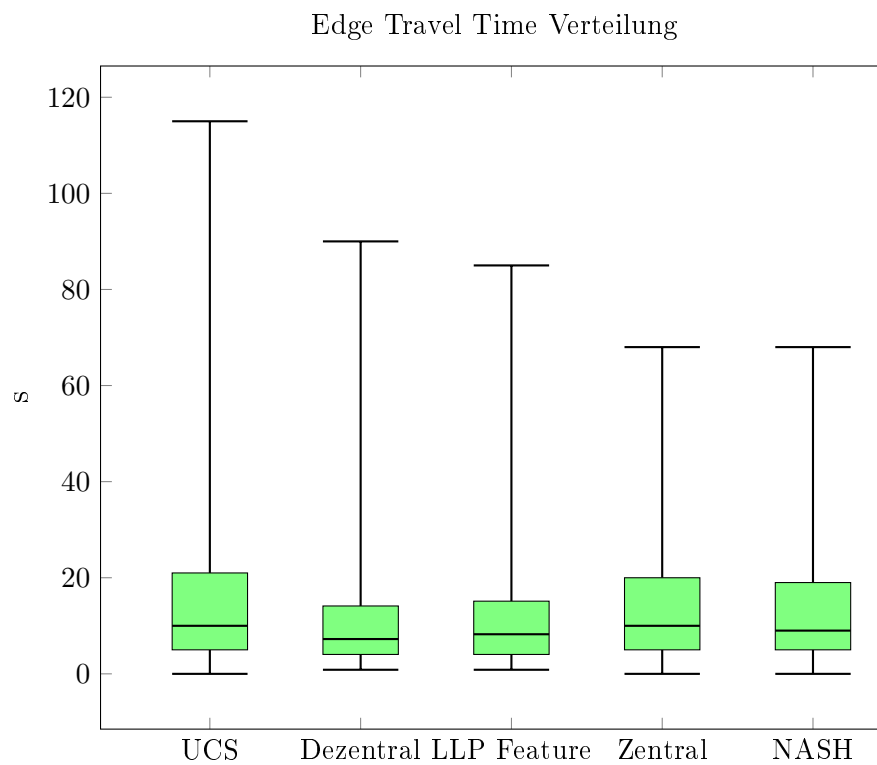


Abbildung 4.19: Ergebnis der Edge Travel Time Verteilung.

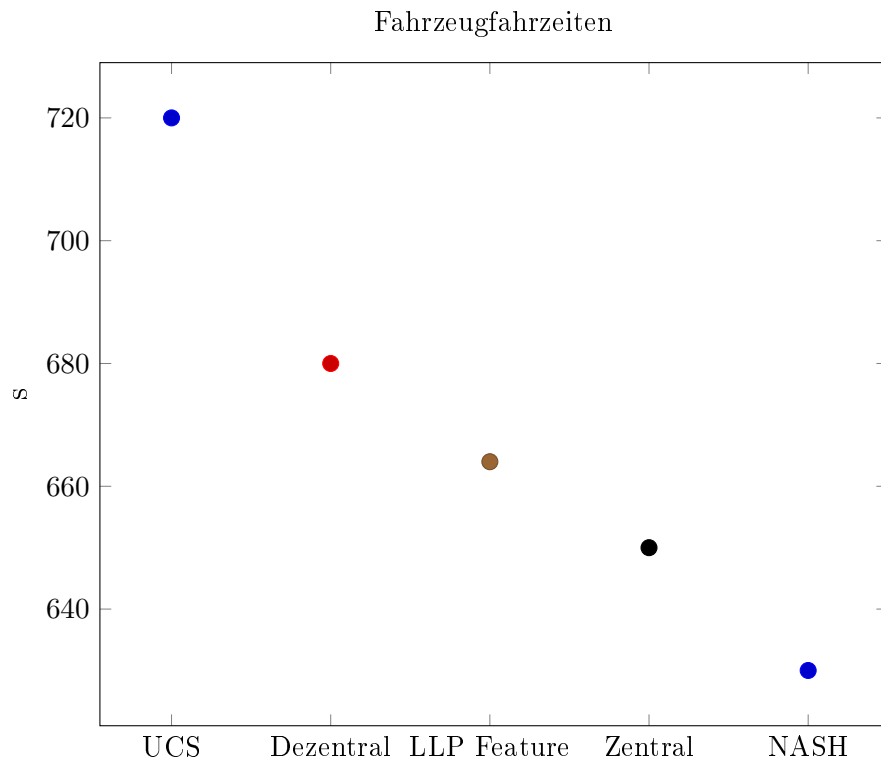


Abbildung 4.20: Ergebnis der Fahrzeugfahrzeiten, die Fahrzeuge durchschnittlich für ihre Routen benötigen.

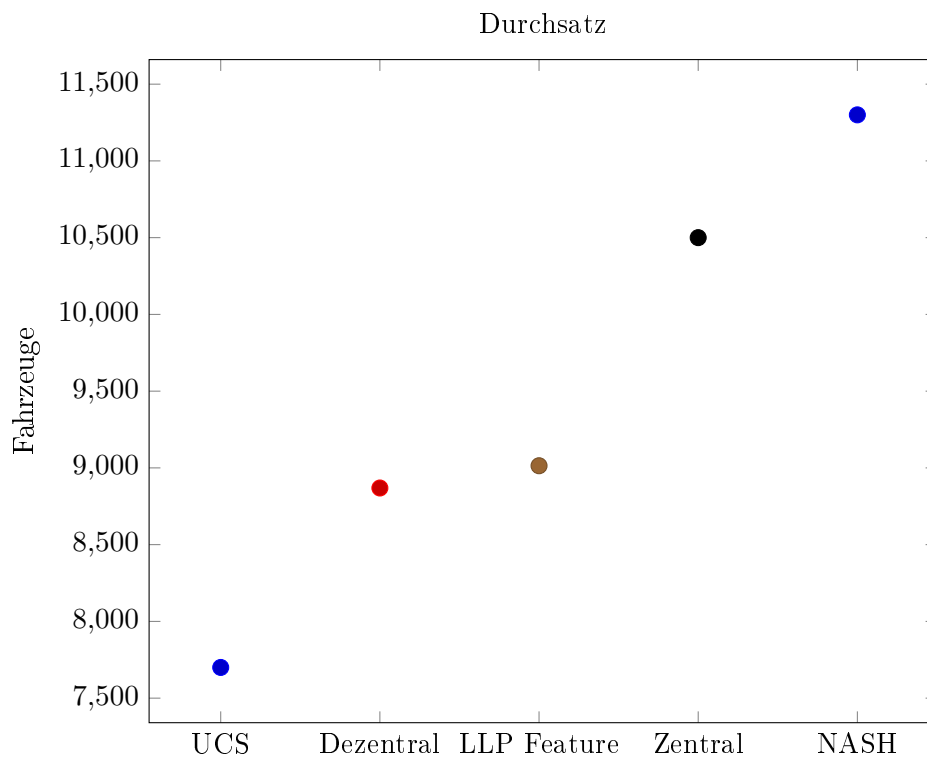


Abbildung 4.21: Ergebnis des Durchsatzes des Verkehrsnetzes.

4.5 Verkehrslenkungsexperiment mit LLP

Diese Form der Verkehrslenkung nutzt ausschließlich die Labelvorhersage mit LLP. Das genaue Vorgehen wird in 3.5 gezeigt. Hierbei wird nur anhand der Vorhersage der Sensoren darüber entschieden, welcher Weg eingeschlagen wird. Dabei wird das durch LLP bestimmte Verkehrsflusslabel dazu genutzt, um die überfüllten Straßen an Kreuzungen zu erkennen. Erkannte Überfüllungen können nun durch die Verkehrslenkung umfahren werden. Dies stellt ein Experiment dar, um LLP auf eine alternative Weise einzusetzen.

In den Abbildungen ist ein ähnliches, aber etwas schlechteres Ergebnis zur dezentralen Verkehrslenkung mit POEM zu erkennen. Also ebenfalls ein schlechteres Ergebnis, als die Verkehrslenkung mit genutztem LLP Feature. Folglich ist die Verwendung von LLP besser geeignet, um es als Feature in POEM zu nutzen.

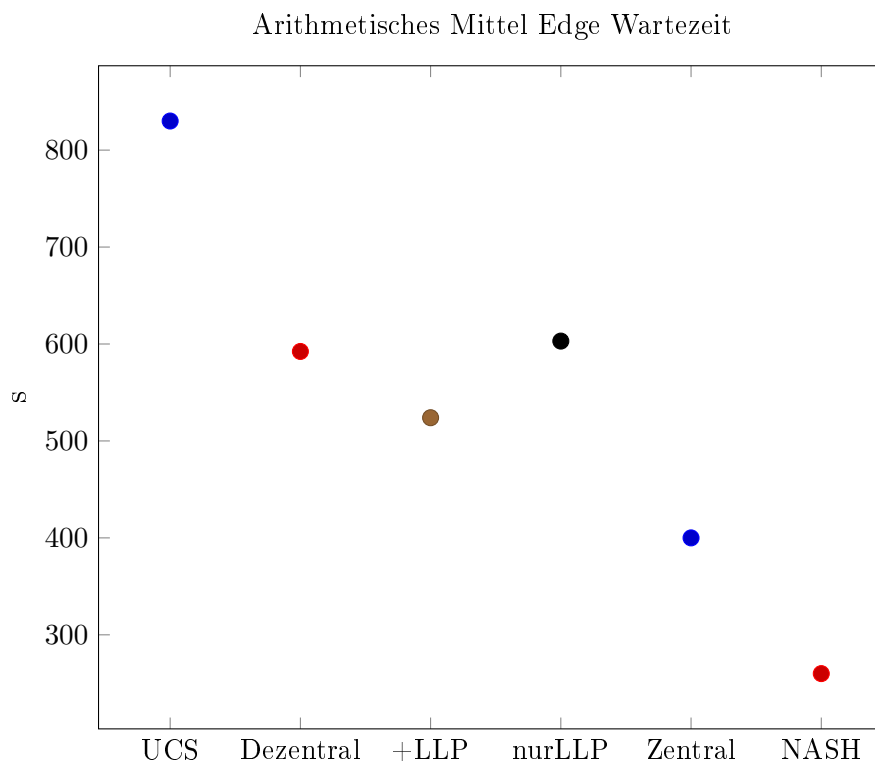


Abbildung 4.22: Ergebnis der Edge Wartezeit als arithmetisches Mittel.

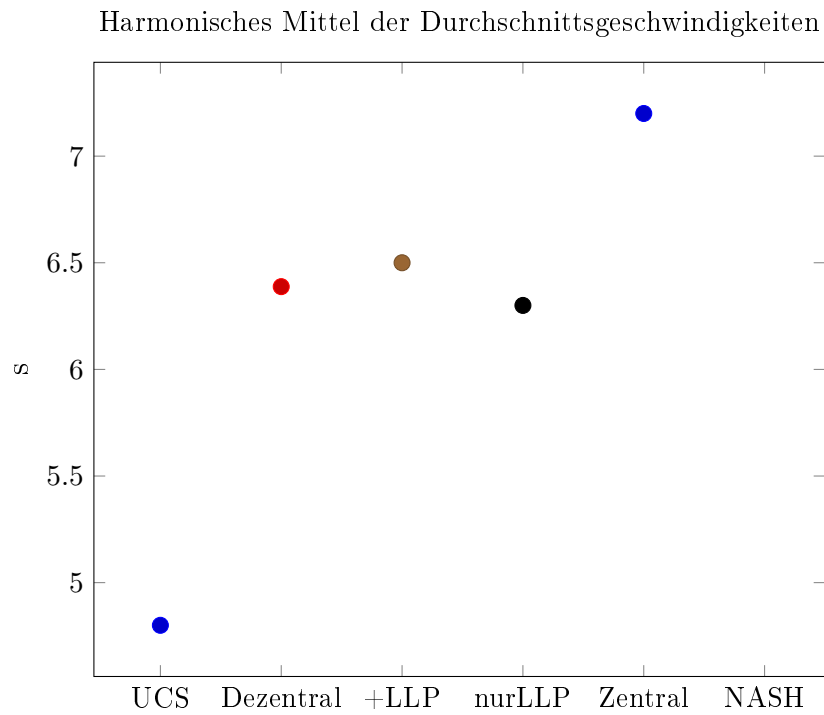


Abbildung 4.23: Ergebnis der Durchschnittsgeschwindigkeiten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

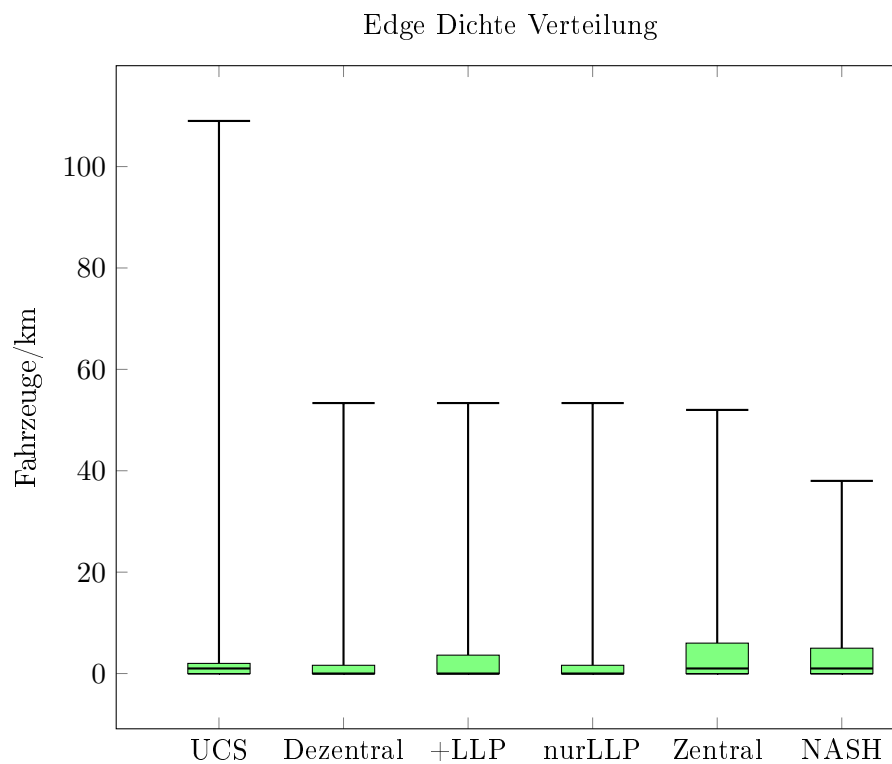


Abbildung 4.24: Ergebnis der Durchschnittsdichten, die an Edges gemessen wurden, auf denen Fahrzeuge fahren.

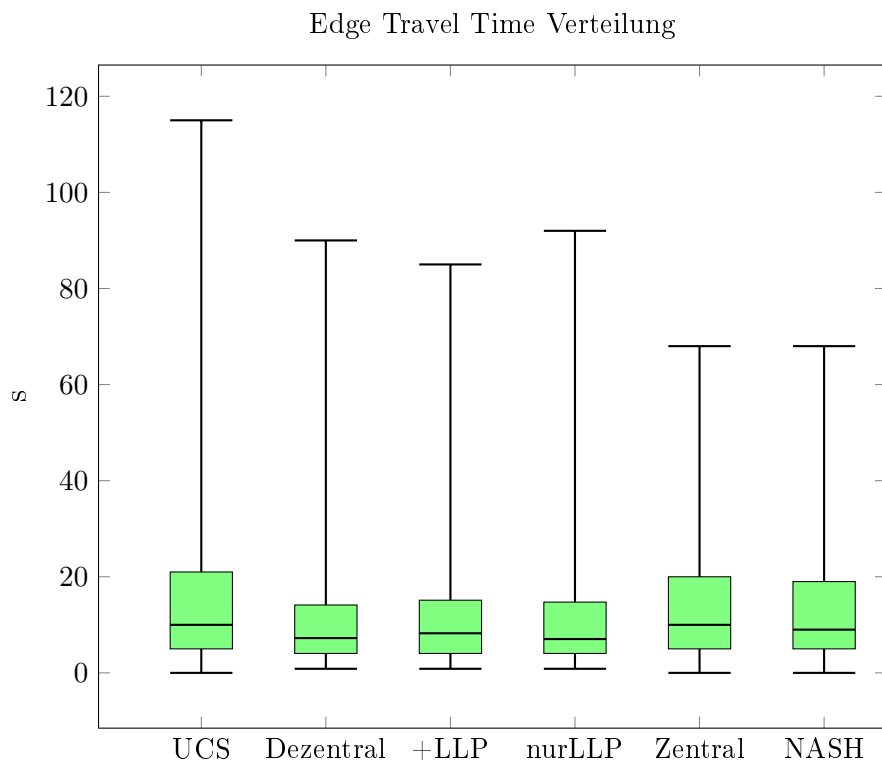


Abbildung 4.25: Ergebnis der Edge Travel Time Verteilung.

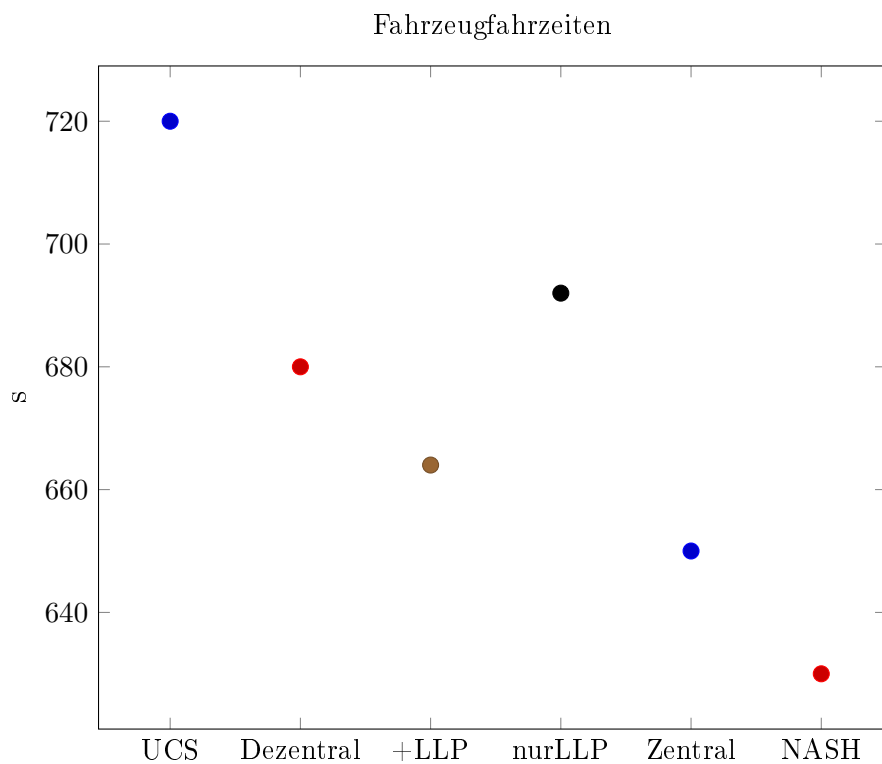


Abbildung 4.26: Ergebnis der Fahrzeugfahrzeiten, die Fahrzeuge durchschnittlich für ihre Routen benötigen.

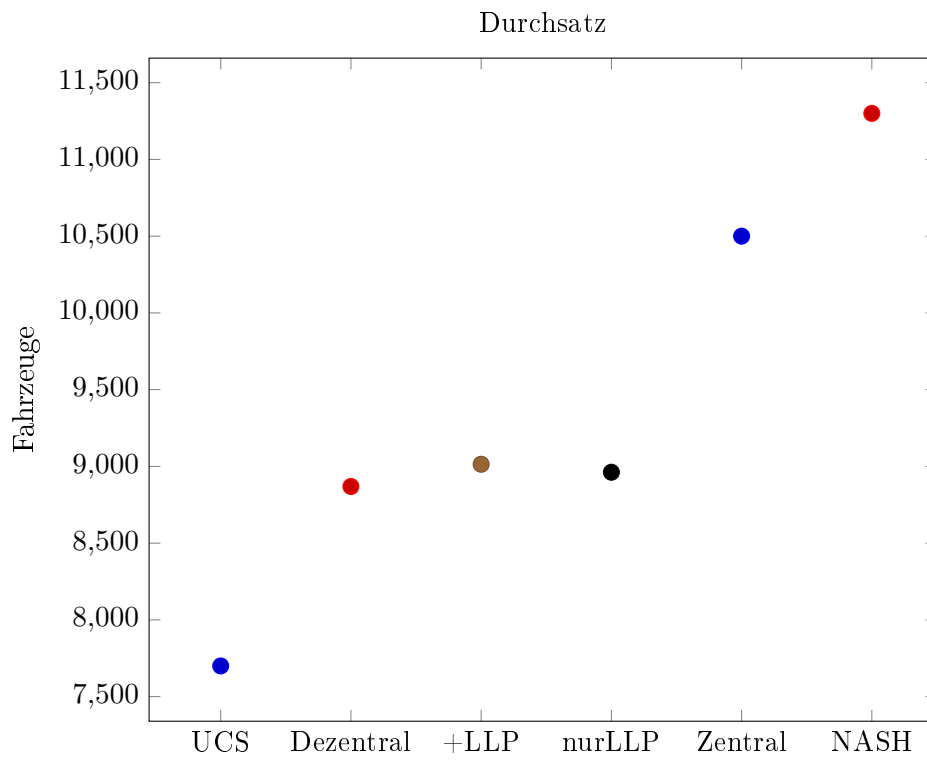


Abbildung 4.27: Ergebnis des Durchsatzes des Verkehrsnetzes.

Kapitel 5

Zusammenfassung

Es ist möglich, durch LLP und POEM die Verkehrslenkung positiv zu beeinflussen. Insbesondere die Erkennung von Verkehrsflusslabels ermöglicht eine bessere lokale Entscheidung, welche Route dynamisch gewählt wird. Dynamische Verkehrslenkung hat einen sehr positiven Einfluss auf die Verteilung von großen Verkehrsaufkommen in einem Verkehrsnetz, sodass dieses besser ausgelastet ist. Somit wird ein besserer Verkehrsfluss in der Verkehrsumgebung ermöglicht. Die Überfüllungen im Verkehrsnetz werden verringert, sodass jedes Fahrzeug weniger Wartezeiten auf dem Weg zu Ziel hat. Auch eine Verbesserung der Fahrzeiten für jeden Verkehrsteilnehmer wird erreicht, sodass ebenfalls mehr Fahrzeuge ihr Ziel schneller erreichen.

Der Einsatz von vorhergesagten Verkehrsflusslabels hat einen positiven Einfluss auf die Erkennung von überfüllten Straßen mit POEM. Durch den Zusatz des LLP Features hat sich die Verkehrslenkung leicht verbessert, kommt jedoch nicht an Ergebnisse von Verkehrslenkungen heran, die auf die gesamten Verkehrsdaten Zugriff haben. Die vorhergesagten Verkehrsflusslabels als Feature für eine Verkehrslenkung durch POEM einzusetzen beeinflusst das Verkehrsverhalten besser als eine ausschließliche Nutzung von LLP Verkehrsflusslabel.

In der Praxis muss eine Stadt für die Anwendung von dezentraler Verkehrslenkung zunächst mit geeigneten Verkehrssensoren ausgestattet werden. Dies könnte in jeder Großstadt relevant sein, in der wenig Möglichkeiten bestehen, weitere Straßen auszubauen, sodass der Weg zu einer durch intelligente Verkehrslenkung gesteuerten Stadt gewählt werden kann. Über Navigationsgeräte oder Navigationssoftware auf Mobiltelefonen könnte der dezentrale Zugriff auf die Verkehrsumwelt eingebracht werden.

Im Vergleich zur dynamischen Verkehrslenkung auf Basis von zentralen Daten kann die dezentrale Verkehrslenkung durchaus mithalten. Die Vorteile bestehen darin, schnell auf überfüllte Straßen zu reagieren und diese zu umfahren. Durch die Vermeidung von erkannten überfüllten Straßen wird der Verkehrsfluss optimiert und überfüllte Straßen erst wieder genutzt, wenn diese wieder frei sind. Jedoch ist es nicht möglich, weit entfernte

überfüllte Straßen zu erkennen, sodass weitere Umwege nötig sind, die vorher abgefangen werden könnten.

Der Einsatz von LLP als weiteres Feature für eine Erkennung von Verkehrsflusslabels verringert vor allem die Wartezeit im gesamten Verkehrsnetz. Ein Vorteil für den einzelnen Verkehrsteilnehmer ist dabei minimal. Jedoch kann ein Umfahren von überfüllten Straßen, die durch LLP erkannt wurden, sehr positiv für den einzelnen Verkehrsteilnehmer sein, da eventuell ein aufkommender Stau vermieden wurde. Der Vorteil besteht darin, dass durch das vorhergesagte Label auf zukünftige Überfüllungen reagiert wird. Im Gegensatz zur Betrachtung von ausschließlich aktuellen Verkehrsinformationen wird eine Überfüllung einer Straße also auch auf die zukünftige Entwicklung überprüft. Es kann also durchaus zu einer Verbesserung des Verkehrsverhalten führen, überfüllte Straßen zu benutzen, da das Wissen über eine Verringerung einer Überfüllung existiert.

Um eine dezentrale Verkehrslenkung zu verbessern, kann direkt durch eine Verbesserung der Labelerkennung positiver Einfluss genommen werden. Hierbei kann beispielsweise eine Anpassung der in POEM genutzten Exponentialfunktion bereits Auswirkungen auf das Ergebnis haben. Auch die Auswahl an weiteren Features für POEM kann eine Verbesserung herbeiführen, da die Nutzung eines LLP Features eine leichte Verbesserung herbeigeführt hat.

Der verwendete LuST Datensatz besteht aus einem Verkehrsnetz mit vielen Hauptstraßen, die durch die Fahrzeuge auch hauptsächlich verwendet werden. Die dezentrale dynamische Verkehrslenkung profitiert von Straßen, die eine Ausweichmöglichkeit zur Hauptstraße bildet. Daher kann die Wahl eines anderen Datensatzes mit der Anwendung der dezentralen dynamischen Verkehrslenkung zu besseren Ergebnissen führen und könnte in weiteren Experimenten erforscht werden. Generell lohnt es sich, diese Form der Verkehrslenkung in Großstädten einzusetzen, da die Entscheidung Überfüllungen zu umfahren dazu führt, dass Fahrzeuge schneller an ihr Ziel gelangen. Gerade kurzfristig entstandene Überfüllungen werden in Großstädten sehr gut abgefangen. Das LuST Szenario ist aktuell die größte offen zugängliche Simulation einer Großstadt. Es ist wünschenswert, weitere Experimente in unterschiedlichen Großstädten durchzuführen, um das Ergebnis zu überprüfen.

In zukünftigen Arbeiten können mehrere Einflüsse auf LLP näher betrachtet werden. Die Parameter wurden aktuell noch nicht umfangreich angepasst. Beispielsweise die Batchgröße, die zur Generierung der Modelle beim Lernen mit LLP genutzt werden, könnten noch weiter untersucht werden, um bessere Ergebnisse zu erhalten. Auch ein zeitlicher Horizont kann für die Nutzung von LLP zu einer Verbesserung führen.

Durch die Existenz der neuen Map für den kritischen Fluss im LuST Szenario besteht die Möglichkeit, daraus weitere Schlüsse über das Verkehrsverhalten zu ziehen. Gerade die Betrachtung von Bottlenecks hat einen großen Einfluss in der Verkehrslenkung.

In dieser Arbeit wurde gezeigt, dass der Einsatz von vorhergesagten Verkehrsflusslabels durch LLP in einem bestärkenden Lernen einsetzbar ist. Die Anwendung auf eine Verkehrsumgebung führt zu einer Verbesserung des Verkehrsverhaltens.

Abbildungsverzeichnis

1.1	Motivation Verkehrslenkung.	2
1.2	Beispielhafte Skizze einer Verkehrsumgebung, die mit Verkehrssensoren ausgestattet ist.	3
2.1	V2X Szenario Übersicht.	6
2.2	Beispiel einer Verkehrsflusssimulation.	8
2.3	Fundamentaldiagramm des Verkehrsflusses.	9
2.4	Freiflussast und gestauter Ast.	10
2.5	Bottleneck Problematik im Graphen.	12
2.6	Beispiel eines unübersichtlichen Bottleneck Problems.	13
2.7	Reinforcement Learning Übersicht.	15
2.8	LLP Bags.	17
3.1	Ausschnitt aus der Kritischer Fluss Map.	24
3.2	Verkehrslenkungsbeispiel.	26
3.3	LLP Datenvorverarbeitung.	32
3.4	Aggregation von Labels.	32
3.5	Cluster Beobachtungen.	33
3.6	Label werden zu Clustern zugeordnet.	34
3.7	LLP als weiteres Feature zur Verkehrslenkung.	34
4.1	Simulation der Stadt Luxemburg als SUMO Szenario.	40
4.2	Luxemburg in SUMO und kritische Dichte Map.	43
4.3	Map des kritischen Flusses der Straßen in der Stadt Luxemburg.	44
4.4	Ergebnis Bestärkendes Lernen: Edge Wartezeit	47
4.5	Ergebnis Bestärkendes Lernen: MeanSpeed	47
4.6	Ergebnis Bestärkendes Lernen: Edge Durchschnittsdichte	48
4.7	Ergebnis Bestärkendes Lernen: Edge Travel Time	48
4.8	Ergebnis Bestärkendes Lernen: Fahrzeugfahrzeiten	49
4.9	Ergebnis Bestärkendes Lernen: Durchsatz des Verkehrsnetzes	49
4.10	Ergebnis Bestärkendes Lernen: Edge Wartezeit im Vergleich	51

4.11	Ergebnis Bestärkendes Lernen: MeanSpeed im Vergleich	51
4.12	Ergebnis Bestärkendes Lernen: Edge Durchschnittsdichte im Vergleich	52
4.13	Ergebnis Bestärkendes Lernen: Edge Travel Time im Vergleich	52
4.14	Ergebnis Bestärkendes Lernen: Fahrzeugfahrzeiten	53
4.15	Ergebnis Bestärkendes Lernen: Durchsatz des Verkehrsnetzes im Vergleich	53
4.16	Ergebnis LLP als Feature: Edge Wartezeit	56
4.17	Ergebnis LLP als Feature: MeanSpeed	56
4.18	Ergebnis LLP als Feature: Edge Durchschnittsdichte	57
4.19	Ergebnis LLP als Feature: Edge Travel Time	57
4.20	Ergebnis LLP als Feature: Fahrzeugfahrzeiten	58
4.21	Ergebnis LLP als Feature: Durchsatz des Verkehrsnetzes	58
4.22	Ergebnis Alternative LLP Anwendung: Edge Wartezeit	59
4.23	Ergebnis Alternative LLP Anwendung: MeanSpeed	60
4.24	Ergebnis Alternative LLP Anwendung: Edge Durchschnittsdichte	60
4.25	Ergebnis Alternative LLP Anwendung: Edge Travel Time	61
4.26	Ergebnis Alternative LLP Anwendung: Fahrzeugfahrzeiten	61
4.27	Ergebnis Alternative LLP Anwendung: Durchsatz des Verkehrsnetzes	62

Literaturverzeichnis

- [AC16] Ehsan Mohammady Ardehaly and Aron Culotta. Domain adaptation for learning from label proportions using self-training. In *IJCAI*, pages 3670–3676, 2016.
- [ALS13] Julian Arz, Dennis Luxen, and Peter Sanders. Transit node routing reconsidered. In *International Symposium on Experimental Algorithms*, pages 55–66. Springer, 2013.
- [ARI⁺15] Afshin Abadi, Tooraj Rajabioun, Petros A Ioannou, et al. Traffic flow prediction for road transportation networks with limited traffic data. *IEEE Trans. Intelligent Transportation Systems*, 16(2):653–662, 2015.
- [AT10] Technical Committee 2.3 Urban Areas and Integrated Urban Transport. Congestion in urban areas - examples of counter-measures. technical report. 2010.
- [BBEK11] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility. In *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, volume 42, 2011.
- [Bot03] Glen D Bottoms. Continuing developments in light rail transit in western europe. In *Proc., 9th National Light Rail Transit Conf*, pages 713–728, 2003.
- [Bun18a] Statistisches Bundesamt. Güterverkehr. www.destatis.de/DE/ZahlenFakten/Wirtschaftsbereiche/TransportVerkehr/Gueterverkehr/Gueterverkehr, 2018. [Online; Eingesehen 29.11.2018].
- [Bun18b] Statistisches Bundesamt. Personenverkehr. www.destatis.de/DE/ZahlenFakten/Wirtschaftsbereiche/TransportVerkehr/Personenverkehr/Personenverkehr, 2018. [Online; Eingesehen 29.11.2018].
- [CFFE17] Lara Codecá, Raphaël Frank, Sébastien Faye, and Thomas Engel. Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63, 2017.

- [Cha14] Harideo Chaudhary. Application of the theory of a single first order equation to traffic flow. *Journal of the Institute of Engineering*, 9(1):175–181, 2014.
- [Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [DSSW09] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Engineering route planning algorithms. In *Algorithmics of large and complex networks*, pages 117–139. Springer, 2009.
- [dSYM⁺16] Allan M de Souza, Roberto S Yokoyama, Guilherme Maia, Antonio Loureiro, and Leandro Villas. Real-time path planning to prevent traffic jam through an intelligent transportation system. In *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pages 726–731. IEEE, 2016.
- [FJM⁺01] Lino Figueiredo, Isabel Jesus, JA Tenreiro Machado, Jose Rui Ferreira, and JL Martins De Carvalho. Towards the development of intelligent transportation systems. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 1206–1211. IEEE, 2001.
- [FR17] Benjamin Fish and Lev Reyzin. On the complexity of learning from label proportions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pages 19–25, 2017.
- [Hä74] T. Hägerstrand. Tiidsgeografisk beskrivning. page 86–94, 1974.
- [HK13] Serge Hoogendoorn and Victor Knoop. Traffic flow theory and modelling. *The Transport System and Transport Policy: An Introduction*, pages 125–159, 2013.
- [HNZ15] Elyes Ben Hamida, Hassan Noura, and Wassim Znaidi. Security of cooperative intelligent transport systems: Standards, threats analysis and cryptographic countermeasures. *Electronics*, 4(3):380–423, 2015.
- [Hua09] LU Huapu. Urban transportation in china: Current state of reform and future trends. *JOURNEYS*, page 7, 2009.
- [Kat03] Richard Katzev. Car sharing: A new approach to urban transportation problems. *Analyses of Social Issues and Public Policy*, 3(1):65–86, 2003.
- [Kim17] G Kim. Travel time estimation in vehicle routing problem. In *Industrial Engineering and Engineering Management (IEEM), 2017 IEEE International Conference on*, pages 1004–1008. IEEE, 2017.

- [KKS14] Boris S Kerner, Sergey L Klenov, and Michael Schreckenberg. Probabilistic physical characteristics of phase transitions at highway bottlenecks: incommensurability of three-phase and two-phase traffic-flow theories. *Physical Review E*, 89(5):052807, 2014.
- [KPF⁺09] John P Kerekes, Michael D Presnar, Kenneth D Fourspring, Zoran Ninkov, David R Pogorzala, Alan D Raisanen, Andrew C Rice, Juan R Vasquez, Jeffrey P Patel, Robert T MacIntyre, et al. Sensor modeling and demonstration of a multi-object spectrometer for performance-driven sensing. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*, volume 7334, page 73340J. International Society for Optics and Photonics, 2009.
- [LAA14] Thomas Liebig, Gennady Andrienko, and Natalia Andrienko. Methods for analysis of spatio-temporal bluetooth tracking data. *Journal of Urban Technology*, 21(2):27–37, 2014.
- [LPBM14] Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. Predictive trip planning-smart routing in smart cities. In *EDBT/ICDT Workshops*, pages 331–338, 2014.
- [LS17] Thomas Liebig and Maurice Sotzny. On avoiding traffic jams with dynamic self-organizing trip planning. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 86. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- [LSM15] Thomas Liebig, Marco Stolpe, and Katharina Morik. Distributed traffic flow prediction with label proportions: from in-network towards high performance computation with mpi. In *Proceedings of the 2nd International Conference on Mining Urban Data-Volume 1392*, pages 36–43. CEUR-WS. org, 2015.
- [MGLW15] Markus Maurer, J Christian Gerdes, Barbara Lenz, and Hermann Winner. *Autonomes Fahren: technische, rechtliche und gesellschaftliche Aspekte*. Springer, 2015.
- [MTL78] Robert McGill, John W Tukey, and Wayne A Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.
- [Nas51] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.
- [ORO02] Seri Oh, Stephen Ritchie, and Cheol Oh. Real-time traffic measurement from single loop inductive signatures. *Transportation Research Record: Journal of the Transportation Research Board*, (1804):98–106, 2002.

- [QMT⁺18] Zhiquan Qi, Fan Meng, Yingjie Tian, Lingfeng Niu, Yong Shi, and Peng Zhang. Adaboost-llp: a boosting method for learning with label proportions. *IEEE transactions on neural networks and learning systems*, 29(8):3548–3559, 2018.
- [QSCL09] Novi Quadrianto, Alex J Smola, Tiberio S Caetano, and Quoc V Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10(Oct):2349–2374, 2009.
- [RBD⁺10] Siuli Roy, Somprakash Bandyopadhyay, Munmun Das, Suvadip Batabyal, and Sankhadeep Pal. Real time traffic congestion detection and management using active rfid and gsm technology. *ITSC, Kyoto, Japan*, 48, 2010.
- [SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [Sch04] Andreas Schadschneider. Physik des straßenverkehrs. *Institut für Theoretische Physik, Universität Köln*, 2004.
- [Sch12] Matthias Schubert. Mathematik für informatiker. *Ausführlich erklärt mit vielen Programmbeispielen und Aufgaben*, 2, 2012.
- [SJ15] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823, 2015.
- [SM11] Marco Stolpe and Katharina Morik. Learning from label proportions by optimizing cluster model selection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 349–364. Springer, 2011.
- [Sot16] Maurice Sotzny. Dynamic traffic routing with bandit feedback learning. Bachelorarbeit, Technische Universität Dortmund, 2016.
- [SS12] Dominik Schultes and Peter Sanders. Dynamic highway-node routing. 2012.
- [Tom18] TomTom. Traffic Index. https://www.tomtom.com/en_gb/trafficindex/list?citySize=LARGE&continent=ALL&country=ALL, 2018. [Online; Eingesehen 29.11.2018].
- [Vra17] Tim Vranken. Untersuchung von strategien zur beeinflussung von fahrzeugen zur verkehrslenkung. Masterarbeit, Universität Duisburg-Essen, 2017.
- [WPR⁺08] Axel Wegener, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. Traci: an interface for coupling road traffic

and network simulators. In *Proceedings of the 11th communications and networking simulation symposium*, pages 155–163. ACM, 2008.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 13. Dezember 2018

Tim Bierwirth

