Technical University of Munich
TUM School of Computation, Information and Technology

Crowd management based on direct communication technology
Christina Maria Mayr

Complete reprint of the dissertation approved by the TUM School of Computation, Information and Technology of the Technical University of Munich for the award of the

Doktorin der Naturwissenschaften (Dr. rer. nat.)

Chair: Prof. Dr. Thomas Neumann

Examiners:

1. Prof. Dr. Hans-Joachim Bungartz
2. Prof. Dr. Gerta Köster
3. Senior Lecturer Dr. Fergus Gilmour Neville

The dissertation was submitted to the Technical University of Munich on 2 April 2024 and accepted by the TUM School of Computation, Information and Technology on 12 June 2024.

Abstract

In the last decade, pedestrian dynamics research has created a knowledge base on crowd behavior. Increasing focus is now being put on effectively managing crowds to ensure safety and well-being. Previous approaches to crowd management were often based on human intervention, but this is error-prone and only possible where personnel is available. In addition, the infrastructure for efficient information distribution often needs to be set up. One approach is to guide a crowd via their smartphones.

Direct communication technologies make it possible to exchange data between devices even without a cellular network infrastructure. At a major event, information could still be provided to the crowd even if the network breaks down. The technology is already used in vehicle communication and could open up new possibilities for crowd management as soon as it is available for smartphones.

In this thesis, I investigate how crowds can be managed using direct communication technology. I propose a novel system that I build and investigate step by step. In simulation studies, the suitability of direct communication for redirecting crowds is investigated, and the performance of crowd redirection algorithms that automatically generate guidance messages based on environmental data is compared. With an online survey, I assess how the willingness to follow redirection instructions can be fostered. I then combine the results of the investigations into an overall concept and test it for a real-life case, again by running simulations. It is demonstrated that the proposed system is capable of resolving safety-critical congestion.

The work contributes a new approach to the research field of crowd management. It shows that crowds can indeed be managed using mobile applications based on direct communication technology. Moreover, the work enhances the understanding of how such a system can be systematically studied and designed. Another important contribution of this work is the scientific and open-source simulation framework *CrowNet* created for the simulation studies.

Kurzfassung

In der Personenstromforschung wurde im letzten Jahrzehnt eine Wissensbasis zum Verhalten von Menschenmengen geschaffen. Der Fokus rückt nun zunehmend auf das effektiven Managen von Menschenansammlungen, um Sicherheit und Wohlbefinden zu gewährleisten. Bisher basieren Ansätze zur Personenstromleitung darauf, dass Personal aktiv eingreift, was jedoch fehleranfällig ist und nur dort möglich ist, wo Personal zur Verfügung steht. Zudem ist nicht immer eine Infrastruktur für eine effiziente Informationsverteilung vorhanden. Ein Lösungsansatz ist Informationen über Smarthphones bereitzustellen.

Direkte Kommunikationstechnologien ermöglichen den Austausch von Daten zwischen Geräten auch ohne Netzinfrastruktur. Bei einer Großveranstaltung könnten damit auch bei einem Ausfall des Netzes noch sicherheitsrelevante Informationen bereitgestellt werden. Direkte Kommunikation wird bereits in der Fahrzeugkommunikation eingesetzt und könnte, sobald diese auch für Smartphones verfügbar ist, neue Möglichkeiten zur Personenstromleitung eröffnen.

In dieser Arbeit untersuche ich, wie Menschenmengen mithilfe von direkter Kommunikationstechnologie geleitet werden können. Ich schlage ein neuartiges System vor, das ich schrittweise aufbaue und untersuche. In Simulationsstudien teste ich die Eignung von direkter Kommunikation zur Umleitung von Menschenmengen und vergleiche die Performance von Algorithmen, die Routenempfehlungen automatisiert, basierend auf Umgebungsinformationen, generieren. Mithilfe einer Umfrage untersuche ich, wie die Bereitschaft, Umleitempfehlungen zu folgen, erhöht werden kann. Die Ergebnisse aus den Untersuchungen führe ich in ein Gesamtkonzept zusammen und teste dies mithilfe von Simulationen für einen realen Anwendungsfall. Ich demonstriere, dass das von mir vorgeschlagene System in der Lage ist, sicherheitskritische Stausituationen zu vermeiden.

Die Arbeit ergänzt das Forschungsfeld Crowd Management um einen neuen Ansatz. Die Ergebnisse zeigen, dass es möglich ist, Personenströme mithilfe direkter Kommunikationstechnologie zu leiten. Die Arbeit trägt zum Verständnis bei, wie ein auf direkter Kommunikationstechnologie basierendes System zur Umleitung von Personenströmen systematisch aufgebaut und untersucht werden kann. Ein weiterer wichtiger Beitrag dieser Arbeit ist die, für die Untersuchung geschaffene, wissenschaftliche und quelloffene Software *CrowNet*.

Acknowledgments

Many thanks to all the people who supported me during my doctorate.

Special thanks go to my two supervisors, Prof. Dr. Gerta Köster and Prof. Dr. Hans-Joachim Bungartz:

Gerta, you taught me how to get the best out of myself and my thesis.

Hans, you taught me what a real (scientific) discussion looks like.

Thanks to Stefan Schuhbäck and Prof. Dr. Lars Wischhof, with whom I worked on the research project roVer at Hochschule München University of Applied Sciences:

Stefan, you showed me that Linux is cool and that software engineering is fun.

Lars, you made it possible for me to do research freely.

Many thanks to Dr. Anne Templeton, without whom the psychology part of my thesis would be missing:

Anne, you showed me how exciting interdisciplinary research is.

My thanks to the (former) members of the Pedestrian Dynamics research group at Hochschule München University of Applied Sciences: Dr. Benedikt Zönnchen, Dr. Daniel Lehmberg, Dr. Benedikt Kleinmeier, Dr. Marion Gödel, Dr. Simon Rahn, Sophia Wagner, Sabrina Kern, and (again) Stefan Schuhbäck. Thank you for your professional advice and personal opinions. I have learnt how valuable they are and how grateful I am for all of them.

Thanks to my friends and family. Thanks for your interest in my topic. Thanks to everyone who always had an open ear for me.

Contents

1. Introduction

This dissertation thesis contributes to the research field of crowd management from the perspective of a computer scientist: I design and test a novel crowd guidance system in which crowds are redirected by employing direct communication technology. The contributions are:

- A **proof of concept of the design** of a system in which crowds are redirected through mobile applications based on direct communication.

- A **methodology** how such a system can be developed systematically by combining models and methods from engineering, psychology, and computer science.

- A **simulation framework** with which such a system can be investigated (collaboratively developed in the *roVer* research project work).

> **Availability of data and software**
>
> This work is based on the principles of Open Science. Source code and data, including raw data, are freely available. Parts of the software that were developed in collaboration are clearly reported. See Appendix A.

1.1. Motivation

Despite numerous efforts to increase the safety of crowds, crowd crushes still occur [1]. In 2022, 159 people died in the Seoul accident on Halloween. In 2023, 12 people died at a soccer match in El Salvador. More than 2000 people died during the Mina stampede in 2015. In Europe, the Love Parade (2010) and the Oxford Circus stampede (2017) are among the best-known crowd crushes of recent years. However, crowds of people gather not only at large events but also in everyday life. In large cities, this mainly affects local public transport. With growing urbanization, transport hubs are becoming increasingly congested, which will be further exacerbated by efforts to reduce vehicle traffic in cities. Consequently, numerous cities declared their goal to make pedestrian traffic safer and more comfortable: London launched the Walking Action Plan of London in 2018, Munich released a Mobility Strategy in 2019, and Berlin declared the Mobility Act in 2021. To achieve this, new traffic strategies for pedestrians and crowds are needed.

One step towards making traffic safer is to provide safety-relevant information to crowds. To distribute the information, crowd managers typically use signs, light signals, loudspeaker announcements, or employ staff on site. Another approach is to inform the crowd through mobile applications. This, however, requires that information is reliably disseminated over the mobile network. At events with large crowds, the cellular network is often overloaded, which poses the risk that safety-relevant information is lost. One idea is to employ direct communication technologies where data is exchanged directly from device to device, see Fig. 1.1. With this, safety-relevant information could be provided to the crowd independently of the network infrastructure, which is crucial in emergencies.
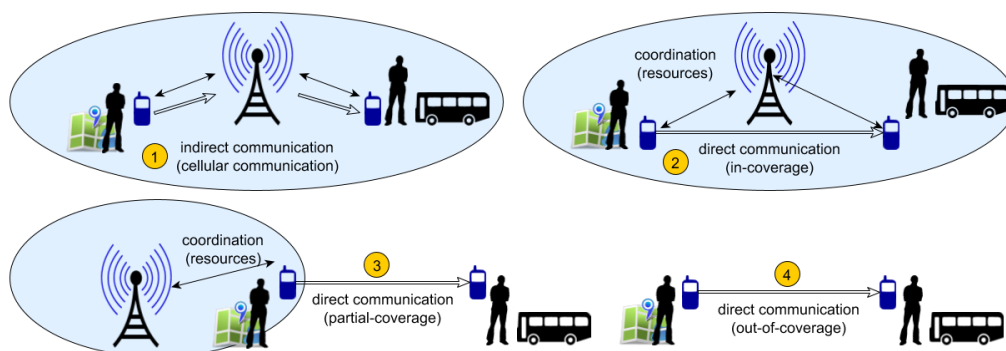
Figure 1.1.: Direct communication in cellular networks. Devices share data via the base station (1). With direct communication, devices share data directly (2). Communication is possible even in partial coverage (3) or out-of-coverage (4) scenarios. Usage of the graphic granted by the creator.

Direct communication is specified in several mobile standards, whereby two primary technologies have emerged. The first is dedicated short range communication, which is based on WLAN ad hoc networks [2]. The first standard was published in 2010 and amended in 2023. The second is 'cellular vehicle-to-everything' (C-V2X) communication, which has been developed since the 2010s, gaining more attention through the transition to 5G [3]. The initial focus of both technologies was on vehicle communication. However, vehicle and pedestrian traffic differ strongly: People do not move in lanes, their speed is lower, and they can be closer to each other than cars. For this reason, the *roVer* research project[1] investigated how direct communication approaches can be transferred to pedestrian traffic. Since the technology has not yet been available for smartphones, the simulation framework *CrowNet* was created for the investigations. One part of the project focused on mobile communication technology, see [4]. I looked at the sociocybernetics aspects: The crowd and the mobile network form a complex socio-technical system whose interactions must be understood to efficiently manage crowds.

---

[1]The roVer research project (2018-2023) was funded by the German Federal Ministry of Education and Research; grant no. 13FH669IX6.

1.2. Research field and research question

Crowd management is an emerging sub-field of pedestrian dynamics research and addresses how crowds can be managed to ensure their safety and comfort. While in the early phase of pedestrian dynamics research, the focus was on creating a knowledge base on crowd locomotion – a prerequisite for managing crowds – today the focus is shifting to crowd management: see the recently published books [5–8]. Numerous strategies have been proposed, such as static measures that involve the planning of facilities [9, 10], placing obstacles [11], or the provision of evacuation plans [12]. Such static strategies are efficient when traffic conditions are precisely known in advance. In the case of an unpredictable event, they might fail as they can not adapt. The focus, therefore, shifts to strategies that dynamically intervene using dynamic signs, dynamic obstacles, loudspeaker announcements, auxiliary equipment like robots [13], background music [14] or crowd managers on site [15]. Another approach to disseminating information is to use mobile applications. There is scarcely any research on how mobile applications can be employed to redirect a crowd except for one single egress experiment [16]. Instead, the pedestrian dynamics community focused on how smartphone usage affects attention and thus movement behavior, see [17, 18]. Apart from that, various crowd sensing approaches [19–23] have been proposed, employing mobile communication.

However, there is currently no system that senses and automatically informs the crowd based on direct communication technology. I aim to close this gap by developing such a system. My overall goal is to make transportation safer. In many real-life use cases, crowds needed to be guided away from congested or safety-critical areas. Therefore, the system provides route recommendations. To distinguish the system from navigation systems for individuals, as one knows them for car drivers, I use the term 'crowd guidance system' in this thesis. My overarching research question is: How can crowds be redirected through mobile applications based on direct communication technologies? The main question leads to sub-questions on information dissemination, compliance with instructions, and suitable message design.

Research question (RQ)

How can crowds be redirected through mobile applications based on direct communication technologies?

- Research sub-question (RQ-1): How reliably are route recommendations disseminated using direct communication in a crowd?

- Research sub-question (RQ-2): Which algorithms are suitable for generating route recommendations for crowds, given not all people follow instructions?

- Research sub-question (RQ-3): How can mobile messages with route recommendations be designed so crowds follow instructions?

1.3. Research methodology

To answer the research question, I propose and investigate the crowd guidance system step by step. In the course of this, technical and psychological aspects are examined that may affect the performance of a crowd guidance system, see Fig. 1.2.
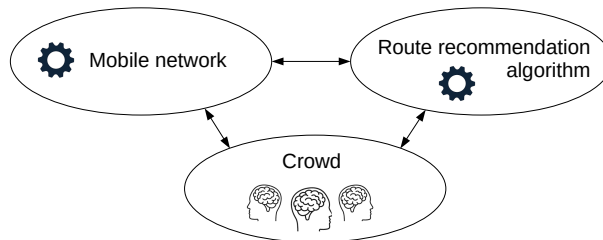
Figure 1.2.: Components of a crowd guidance system examined in this thesis. I analyze the system from a sociocybernetics perspective, looking at technological and psychological aspects and their interaction.

In step (1), I investigate the suitability of direct communication technology for redirecting mobile crowds. Several studies [24–29] showed that mobility behavior affects the information dissemination in the mobile network. Crucially, the accuracy of the locomotion model affects the results of the mobile networks simulation [30]. Many movement models have already been used in network simulations, see the overview [31]. Among them are models from the pedestrian dynamics community, such as the Legion Model and the Social Force Model, see [26, 29]. So far, only the influence of mobility behavior on the network simulation has been investigated. In a crowd guidance system, however, the mobile network and mobility behavior influence each other bi-directionally: The crowd changes its behavior due to the information disseminated via the mobile network, which, in turn, influences the mobile network. Such a dynamic system has yet to be investigated. It needs to be clarified how redirection measures affect shadowing and interference, thus, information dissemination in the crowd. My first sub-question therefore asks how reliably redirection information is disseminated in a redirected crowd using direct communication technology, see (RQ-1).

In step (2), I investigate route recommendation algorithms. Meanwhile, numerous guidance approaches have been proposed for crowd management. The algorithms, implemented in the so-called controller, mainly come from engineering. Among them are feedback controllers that aim to affect the locomotion behavior [32–34] and controllers trying to influence the decision-making of pedestrians [35]. The complexity of these controllers varies greatly from simple On-Off controllers [36] to more sophisticated methods, such as Model Predictive Control [37]. Unfortunately, most algorithms were developed under unrealistic conditions: It was assumed that the crowd always followed instructions. This is wrong because people may but do not have to follow instructions. Even if they are willing, they may be physically unable or simply not understand what they are

supposed to do. I, therefore, find it essential to investigate which algorithms are suitable when not all people follow instructions, see (RQ-2).

In step (3), I investigate how one can improve the compliance of the crowd to follow route recommendations. Studies from communication sciences showed that human behavior can be influenced using message framing, i.e., 'presenting logically equivalent options in semantically different ways' [38]. Message framing has been applied in numerous fields like health sciences, marketing, or political communication [39]. Techniques from message framing have also been applied to influence crowd behavior: Carter et al. [40, 41] varied the detail of information in a decontamination experiment and found that the crowd complied most when instructions included detailed explanations and instructions. Similar findings have been obtained from studies with individuals [42, 43]. The communication of information is one factor that can have an impact on behavior. Research from the field of social identity showed that groups and individuals behave differently when group members share a social identity [41, 44, 45]. This can indeed affect the perception of group members [46], and, thus, the decision behavior. Therefore, I want to investigate how message framing can be combined with theory on social identities to find a suitable message design that fosters crowd members to follow route recommendations, see (RQ-3).

I combine methods from computer science, engineering, and the social sciences to answer the main research question and the sub-questions. To investigate the suitability of direct communication in step (1), I model a redirection scenario and conduct a simulation study. I proceed similarly in step (2) to find a suitable route recommendation algorithm. In step (3), an online survey is carried out to assess the effect of message design on route choice. In step (4), I present the complete crowd guidance system, apply it to a real-life use case at the metro station Münchner Freiheit in Munich (Germany), and test it in simulations for which I estimate parameters from the survey data.

Proposed methodology for designing a guidance system

(1) Test the suitability of direct communication for disseminating route recommendations in a moving crowd ⇒ RQ-1. *Method: Modeling & Simulation*

(2) Find a suitable route recommendation algorithm ⇒ RQ-2. *Method: Modeling & Simulation*

(3) Find a message design that convinces crowd members to follow route recommendations ⇒ RQ-3. *Method: Survey*

(4) Use the findings from (1)-(3) to design and test a proof of concept. ⇒ RQ. *Method: Modeling & Simulation*

1.4. Structure of this work

The thesis is divided into three main chapters, see Fig. 1.3. Chapter 2 presents the state of the art on crowd management and approaches for modeling, simulating, and investigating a crowd guidance system. In Chapter 3, I introduce the novel simulation framework *CrowNet* that was collaboratively developed in the *roVer* research project. In Chapter 4, I use this software to investigate my research questions following the proposed methodology. Sections 4.1-4.3 correspond to steps (1)-(3), where I investigate isolated components of a crowd guidance system to answer the research sub-questions. In Section 4.4, I answer the main research question: I propose and test a complete crowd guidance system for a real-life use case at a metro station in Munich (Germany) where football fans need to be guided away from a congested route.
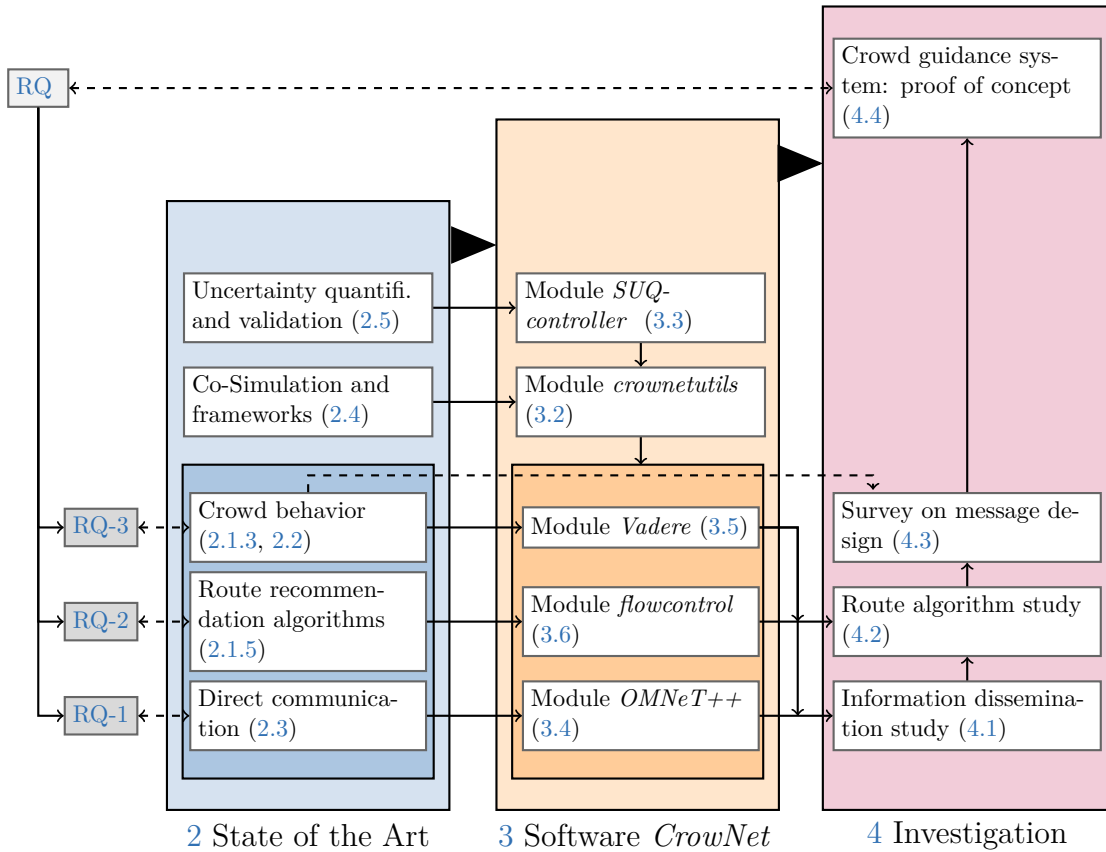
Figure 1.3.: Reading instructions. The thesis is divided into three main chapters. The arrows indicate how the research question (RQ) and the sub-questions (RQ-1, RQ-2, RQ-3) are related to the individual chapters and sections.

2. State of the Art

This section consists of an application-related part, where I present state-of-the-art approaches and theory on crowd management, and a methods-related part that comprises approaches from computer science for the simulative assessment of a crowd guidance system.

In the first part (Section 2.1), I look at crowd management from a traffic engineering perspective to give an overview of the research field and its application in practice. I identify research gaps that I aim to close in my later investigations. Therefore, the first part is strongly connected to the research questions presented in the introduction.

In the second part (Sections 2.2-2.5), I look at methodological aspects from a computer science perspective. As I base my investigations on simulations, I look at how the key components of a crowd guidance system, which are the crowd and the mobile network, can be modeled and simulated. In Section 2.2, approaches for modeling and simulating crowd behavior are presented. I introduce a hierarchical modeling approach where locomotion is modeled at the operational layer and route choice at the tactical layer. In Section 2.3, I provide an overview of mobile communication standards that specify direct communication and discuss how the respective protocol stacks and the radio channel are modeled and simulated. Since crowd and network models are separate models, Section 2.4 presents techniques for connecting them. This includes update schemes and techniques for data exchange between simulators. It is crucial to quantify errors or uncertainties in the simulation output. Therefore, I present a surrogate-based approach with polynomial chaos expansions in Section 2.5.

2.1. Crowd Management

This section belongs to the application-related part of the state of the art. First, crowd management is differentiated from crowd control. Next theory and metrics on crowd safety and comfort are introduced. To foster people's compliance to follow route instructions, I look at theories that describe the psychological behavior of crowds and the aspects of communication. Then I describe approaches for measuring crowd properties. At the end of this section, I give an overview of suggested approaches where guidance signals are automatically generated using algorithms.

2.1.1. What is Crowd Management?

"Crowd management aims for behavioral change by applying information measures, especially to optimize the use of available infrastructural capacity" [47]. The term 'crowd management' denotes "proactive security activities that bring safety and comfort to individuals by facilitating efficient movement of crowds" [7, p.3]. Crowd management is employed at busy places such as airports, shopping malls, exhibition halls, or train stations and at major events such as concerts or sports competitions [7]. There are numerous strategies to influence the movement of crowds, which are divided into five levels according to their intensity: inform, advise, guide, steer, and enforce [47, 48], see Fig. 2.1.
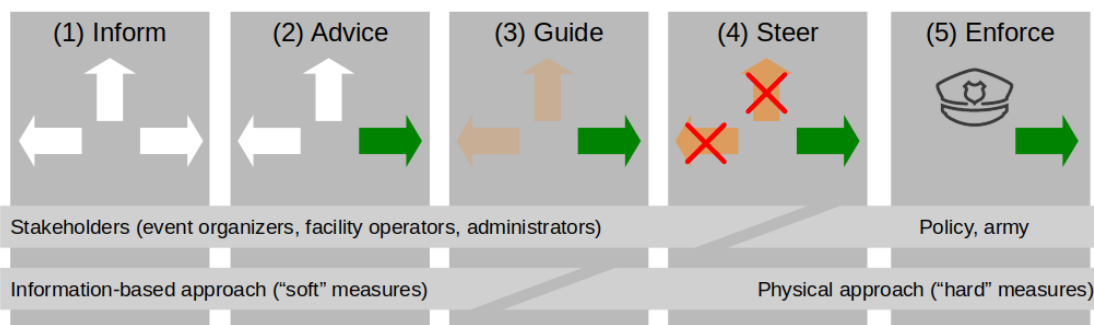
Figure 2.1.: Strategies for influencing crowd behavior. Crowd management uses information-based approaches such as informing and providing advice (1,2). Crowd control employs physical measures to enforce a certain behavior (4,5). Guiding people (3) is located at the transition between the soft (1,2) and the hard (4,5) measures. Own graphic inspired by [7, p.168].

In the 'inform' stage, neutral information is provided using signs, maps, and colored paths. In the 'advice' stage, information is manipulated to bias the decision, for example, the naming of walking paths: football fans might favor a walking path to the stadium named after a famous football player. Notably, the crowd is unaware of the manipulation [7, p.169]. In the 'guiding' stage, the manipulation is presented obviously to the crowd using instructions that are disseminated through loudspeakers, announcements, information monitors, or light signage. The intensity of the measures is higher than in the 'advice' stage and can involve low-intense physical measures such as staff blocking a closed gate. The crowd is fully aware of the manipulation. In the 'steer' stage physical measures, like for example placing a barrier, are employed. In this stage, it takes people great effort to defy the measures. In the 'enforce' stage, specific behavior is enforced by the police or army using water cannons, tear gas, or brute force. Crowd management corresponds to the stages that aim to inform, advise, and guide [7]. Importantly, people are free to decide whether to follow instructions. Therefore, it is essential to consider the compliance of people when developing a crowd guidance system.

2.1.2. Quantitative characteristics of crowds and safety evaluation

Crowd and pedestrian behavior can be quantified and evaluated using several measurement quantities and metrics. One of them is the fundamental diagram that describes the relationship between density and flow [49], see Fig. 2.2. The flow increases over the density until the critical density is reached, at which the flow is maximal, that is, the capacity. Beyond the critical density, the flow decreases until the jamming density is reached, and movement is no longer possible. The density range between $0\,\mathrm{ped/m^2}$ and the critical density is called the free flow regime because there is enough space to allow most pedestrians to walk at their desired speeds [7]. According to the Handbook of Fire Protection Engineering (SFPE) [50], the free flow regime corresponds to a density range $[0\,\mathrm{ped/m^2}, 1.88\,\mathrm{ped/m^2}]$.
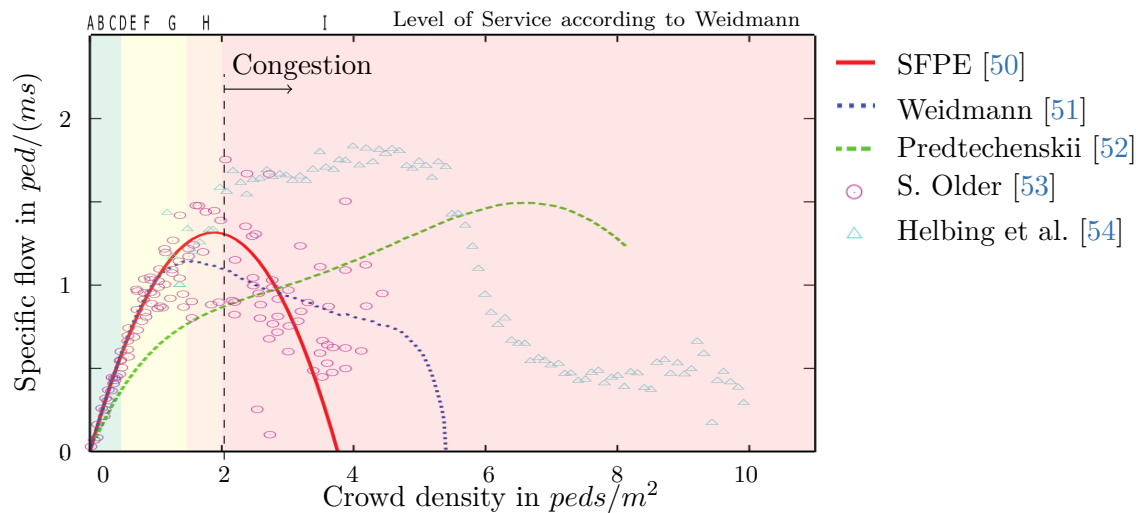
Figure 2.2.: Fundamental diagram. The flow increases over the density until the maximum flow (capacity) is reached. For higher densities, the flow decreases. The shape of the fundamental diagram differs for the studies [50–54] because it depends on several factors, such as age or culture. Own graphic inspired by [55, 56].

As one can observe in Fig. 2.2, the fundamental diagram differs from study because the density-flow-relationship depends on several factors: topological features, such as crossings, stairs [57], the directionality of flows [58], age [59, 60], religion [61], and culture [62], visibility conditions [63], the motivational level [64], environmental conditions such as the temperature [65], and the measurement method [55]. For an overview of factors, I refer to [66].

Important for crowd management applications is that the location and size of the measurement area have an influence: The maximum density cannot be measured in the

center of a bottleneck but in front of it [55]. Also, the larger the area, the smoother the density measurements.

To describe the relationship between flow and density analytically, numerous models have been proposed, see e.g. [50, 67–71]. In the pedestrian dynamics community, the Kladec equation [72] is widely used that relates the specific flow J_s to the density ρ:

$$J_s = \rho v_0 \left(1 - e^{-1.913\left(\frac{1}{\rho} - \frac{1}{\rho_{\mathrm{jam}}}\right)} \right) \tag{2.1}$$

with the jamming density $\rho_{\mathrm{jam}} = 5.4 \, \mathrm{ped/m}^2$ and the free flow velocity $v_0 = 1.34 \, \mathrm{m/s}$ according to Weidmann [72].

Congestion definition based on the fundamental diagram

The definition of congestion is an ongoing topic within the pedestrian dynamics community. Currently, no standardized definition is available. Compare, for example, the definitions in [73] and [56]. Kitzlinger et al. [56] suggest defining congestion based on the fundamental diagram: Congestion is present when the capacity is exceeded, or, in terms of flows, jamming occurs if the inflow is larger than the capacity. This criterion is identical to the jamming criterion proposed in [74, 75]. In terms of densities, this means that the density must stay below the critical density.

There are several values suggested for the critical density: $1.75 \, \mathrm{ped/m}^2$ according to Weidmann [51], $2.16 \, \mathrm{ped/m}^2$ according to Fruin [69], $1.88 \, \mathrm{ped/m}^2$ according to the Handbook of Fire Protection Engineering [50] and $6.64 \, \mathrm{ped/m}^2$ according to Predtechenskii [52]. Except from the study in [52], the critical density is always around $2 \, \mathrm{ped/m}^2$ which aligns with the qualitative observations of Oberhagemann [76]. Therefore, this value is a suitable threshold for crowd management applications.

Level of service: Basis for safety evaluation

The Level of Service (LOS) qualitatively describes the movement behavior of a crowd. The density is divided into levels, each qualitatively describing a certain movement behavior, see Tab. 2.1. Therefore, it is used to evaluate safety and comfort.

Like the fundamental diagram, the qualitative movement behavior depends on numerous factors, which is why different level of service schemes have been proposed [77]. In crowd management applications, the categorizations according to Fruin [69] and Weidmann [51] are typically used, see Fig. 2.3.

Weidmann [51] divides the density into nine levels depending on the traffic situation, assigning them levels A (best level) to I (worst level). Free movement in the plane is

Level of Service according to Fruin [69]

Level	Walking behavior	Density ped/m^2	Flow ped/m/min
A	Pedestrians walk with their desired speed	< 0.31	< 23.0
B	Occasionally interference with other pedestrians	< 0.43	< 30.5
C	Partially reduced walking speed	< 0.72	< 49.2
D	Rare passing maneuvers	< 1.08	< 65.6
E	Temporal stops, no passing maneuvers	< 2.17	< 82.0
F	Physical contact between crowd members	≥ 2.17	≥ 82.0

Table 2.1.: Level-of-Service levels for pedestrian traffic according to Fruin [69]. Density and flow values are only valid for walkways. The best category is the A level, where the density and flow are the lowest.
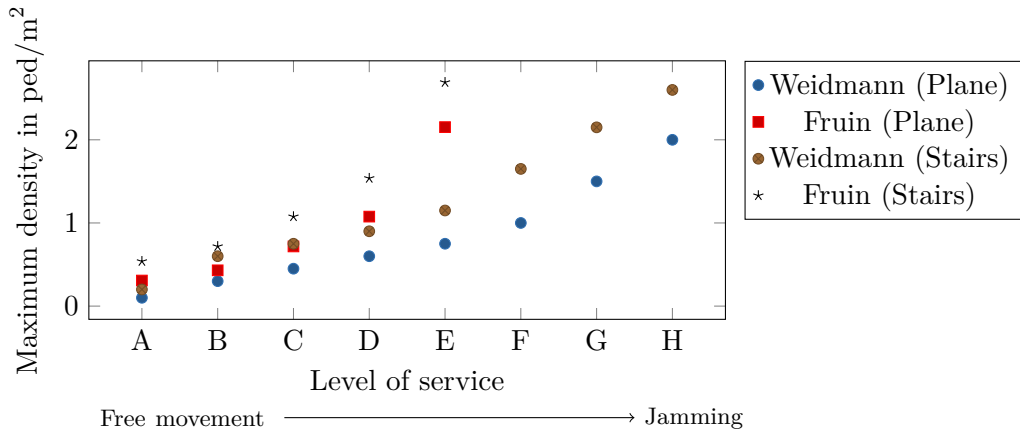
Figure 2.3.: Upper density bounds of the Level-of-Service concepts according to Weidmann [51] and Fruin [69]. For each level the maximum density is depicted for walkways ('Plane') and stairs. The higher the level, the stronger the maximum densities differ (A is the lowest level). One can observe that Weidmann's classification [51] is more rigorous than Fruin's [69]: The blue and brown data points are always below the red and black data points.

present at densities $< 0.1 \, \text{ped/m}^2$, which corresponds to level A [51, p.97]. In Level I, the density $> 2 \, \text{ped/m}^2$ which means there is 'massive crowding' [51, p.97].

Fruin distinguishes only six levels: A-F. The A level describes a free-flow situation, while the F level describes a standstill situation, see Tab. 2.1. Like Weidmann [51], Fruin [69] adjusts the density and flow intervals depending on the scenario. For walking in the plane and for stairs, the density maximal density for each level differs strongly, see again Fig. 2.3.

Evaluation of safety and comfort

Safety and comfort are evaluated using flow or density-based criteria. Several recommendations are available that are based on the level of service concept: Weidmann [51] recommends a B level for walkways in general. A level D is only acceptable during the rush hour and a level F is acceptable in bottlenecks where people do not stay a long time [51]. The Highway Capacity Manual [78] provide recommendations based on the level of service concept according to Fruin [69]: at least a level E should be achieved which is similar to the jamming criterion.

The Pedestrian Comfort Level [79] is a flow-based metric for evaluating comfort. The recommended level depends on the scenario because it is assumed that pedestrians' perception is scenario-specific. In tourist areas where people are likely to spend time voluntarily, people might be particularly sensitive to crowding. In this case, a maximum flow of 11 ped/min/m is recommended. For a transfer scenario at a train station where commuters stay just a short time, a flow of 17 ped/min/m is acceptable, see Fig. 2.4.

Level	Flow *in* peds/m/min	Tourist attractions Streets dominated by retails, food premises		Areas and streets close to privately owned properties		Transport interchanges Areas and streets close to government or office buildings	
		Peak	Average	Peak	Average	Peak	Average
A+	< 3	Comfortable		Comfortable		Comfortable	
A	≤ 5						
A-	≤ 8						
B+	≤ 11						
B	≤ 14	Acceptable		Acceptable			
B-	≤ 17	At risk				Acceptable	
C+	≤ 20	Unacceptable		At risk	At risk		
C	≤ 23						
C-	≤ 26			Unacceptable		At risk	At risk
D	≤ 35						
E	> 35					Unacceptable	

Figure 2.4.: Pedestrian Comfort Levels for different traffic scenarios. It is assumed that people experience comfort scenario-specifically: They are more sensitive to crowding in tourist areas, where they spend their spare time, than at train stations where they only pass a short time for traveling. Own graphic based on the recommendations from [79].

Finally, I would like to address so-called safety indices [80–83], as I have repeatedly come across these metrics in my literature research. Safety indices are used in traffic engineering to compare traffic scenarios within a city. With this, a traffic engineer can identify the most safety-critical spots for improving traffic. However, these indices are unsuitable for crowd management because they only compare but do not evaluate safety.

2.1.3. Key theories on crowd behaviour

A prerequisite for the development of crowd management applications is understanding crowds' behavior. Several theories have been proposed to explain the psychological behavior of crowds [84–91]. One can distinguish between classical and modern theories. Classical theories only explain violent behavior but cannot explain peaceful gatherings, so these theories are regarded as outdated [92]. Since modern theories have emerged from these, I find it essential to be familiar with their basic principles, so I briefly list the key ideas in Tab 2.2.

Classical theory	Basic idea
Le Bon's Group Mind Theory (1895) [84]	Individuals lose their sense of self in a crowd and become anonymous. The actions of the crowd are driven by a group mind controlled by primitive instincts. Therefore, crowds are disruptive by nature.
Theory according to Freud (1921) [85]	Individuals' super-ego is surpassed when they are in a crowd which unlocks their unconscious mind. Therefore, individuals are driven by their instincts.
Individualism Theory (1924) [86]	Crowd violence is caused by violent predispositions of individuals. Predispositioned individuals reinforce each others' behavior, which aggregates violent behavior.
Deindividuation Theory (1952-1989) [87]	The anonymity of the crowd members controls the behavior. Individuals lose their self-awareness. The resulting anonymity causes antisocial behaviors.

Table 2.2.: Classical theories on crowd behavior. These theories explain violent behavior of crowds but cannot explain peaceful gatherings.

Modern crowd theories

Modern crowd theories aim to explain crowd behavior more generally than classical theories. The *Theory of Social Norms* (1936) [88] introduces the concept of social norms. "When a group of individuals faces a new unstable situation and has no previously established interests or opinions regarding the situation, the result is not chaos; a common norm arises, and the situation is structured in relation to the common norm" [88, p.111]. The *Emergent Norm Theory* (1957-1964) [89] extends the concept of social norms. Through interactions, individuals develop new behavioral norms. The *Game Theory* (1972-1974) [90, 91] assumes that crowds act rationally. Individuals perform actions they believe bring the highest payoff. Crowd members seek information from which they predict possible events. Depending on the reward, the actions are derived, starting with the action that maximizes the reward. Since the reward is difficult to determine in advance, the theory lacks of applicability. The *Place Scripts Theory* (1992) [93] suggests that individuals follow behavioral patterns. The sequence of actions is called the script.

Each action is associated with a certain place. "They define, understand, or formulate a script in relation to where they are and interpret the behavior of others." [93]. According to the theory, familiarity with the environment is disadvantageous in emergency situations, which might not be always the case in reality. The *Self-categorization Theory* explains how individuals view themselves in relation to other individuals as a group and how the view of an individual changes when they become a group member. It is based on the meta-contrast principle: Group members perceive strong similarities between themselves and other group members and large differences between themselves and non-group members [94]. The *Social Identity Theory* [95–101] explains how an individual's behavior is influenced by a social identity to a group or crowd, which in turn influences collective behavior. The extent of the sense of belonging is also referred to as the degree of social identification [102]. An example is a group of football fans who identify themselves as football fans. The assumption is that exactly one of a person's social identities is salient at a time: For the football fan only the fan identity is salient. Other identities such as being an academic or a painter do not have an effect.

The *Social Identity approach* combines the self-categorization theory with the social identity theory [102]. It is supported by empirical evidence from numerous experiments that demonstrate the effect of social identities on decision-making and movement behavior. In-group members who share a social identity are more strongly influenced by members of their social group (in-group members) than by non-group members [41, 44, 45, 103]. Non-group members are members of an opposing social group, e.g., football fans of the opposing team or people whose group membership is unknown. Sharing a social identity affects the behavior: It was observed that people are more likely to help each other when sharing a social identity [103]. Consequently, social identities can reinforce positive characteristics and behavior. Negative emotional responses become weaker: One example is that less disgust is perceived within a group compared to non-group members [44]. Social identities also influence mobility behavior. Group members tolerate dense situations more likely and might even enjoy staying in denser and more central parts of the crowd [46]. They also move slower than individuals and stay closer together [104]. Also, different formations have been observed for groups: walking next to each other, walking in line, or forming a V-shape [105–109].

Communication with crowds

For the categorization of communication strategies, a basic understanding of the elements of the communication process is necessary. Therefore, I briefly introduce Berlo's communication model [110] which focuses on the components of the communication process. I used them as categories in my literature research.

Berlo's model [110] is composed of a sender, a message, a channel, and a receiver, see Fig. 2.5. Senders encode their thoughts into a message that is transmitted over a communication channel. The receiver perceives the message and decodes the information. Several factors, such as the socio-cultural background, the structure and content of a

message, or the channel, influence the communication process. Studies in the field of crowd management mainly focus on the channel, testing the suitability of instructions from personnel [15], background music [14], or auxiliary equipment, such as robots [13], for the provision of information. There is scarcely any research on employing mobile applications and smartphones for guiding crowds except for one experiment in which a crowd received instructions via their smartphones to accelerate evacuation from a room with four exits [16].

encodes   decodes

| Sender | Message | Channel | Receiver |
| Communication skills, attitudes, knowledge, socio-cultural system | Code, content, treatment, elements, structure | Seeing, hearing, touching, ... | Communication skills, attitudes, knowledge, socio-cultural system |
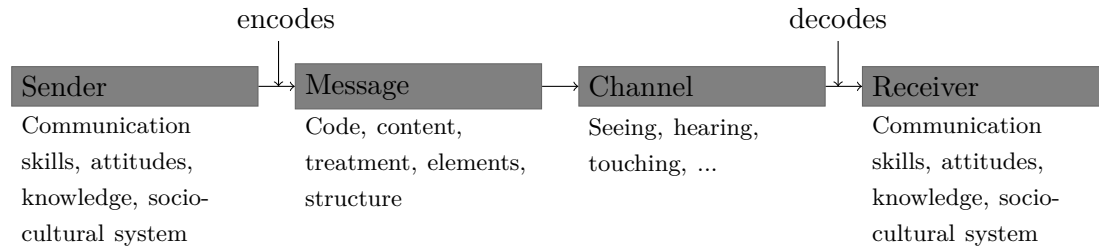
Figure 2.5.: Components and factors of the communication process. Own graphic of Berlo's communication model proposed in [110].

Numerous theories and experimental studies aim to understand how information can be efficiently communicated to crowds in emergencies. Reynolds et al. [111] provide a theoretical basis by introducing the Crisis and Emergency Risk Communication Model that defines communication activities for different phases of a disaster. Drury et al. [112] conducted a virtual reality experiment where participants imagined being at a metro station when a fire breaks out. They found that people helped each other more often when they shared a social identity.

Reicher et al. [113, p.16] stress the necessity of communication in conflicts: "[...] it becomes increasingly important to communicate with the crowd where one seeks to avoid a potentially conflictual relationship, but in situations where relationships are potentially conflictual, crowd members are least likely to trust what the police have to say". Also, it is suggested to use mediators that are trusted and respected within the crowd [113]. Other studies focus on the design of signage that assesses where and how signs should be placed [114].

Palttala et al. [115] suggest employing several communication strategies at once to address heterogeneous crowds. Van der Wal et al. [116] found that "dynamic emergency exit floor lighting and staff guiding people to exits were only beneficial for high-density crowds and those unfamiliar with the environment".

Numerous studies from the communication sciences investigate how the message content affects decision behavior. In particular, the quantity and quality of information are analyzed: how much information is necessary? In which way should a particular piece of information be presented? The latter includes the so-called message framing, i.e., 'presenting logically equivalent options in semantically different ways' [38]. Message

framing plays a role in numerous fields of application. In some studies, message framing did not have any effect: Carfora et al. [117] found that the message components '..., you will feel more energetic' and '..., you will feel less tired' were equally effective in persuading people to switch to a particular diet. Capps at al. [118] tested the effect of message framing on sanitizer usage during the Covid-19 pandemic. They put different signage next to sanitizers with three different messages: (1) 'Using hand sanitizer means you're doing your part to fight the spread of COVID-19 ... Stay healthy!', (2) 'Many people use hand sanitizer to fight the spread of COVID-19 ... Join them!', and (3) 'More and more people are using hand sanitizer to fight the spread of COVID-19 ... Join them!'. They could not find a statistical difference in sanitizer usage.

Other studies provide strong evidence that message framing indeed affects the behavior: Nobel [119] investigated how message framing can affect peoples' decision to quit smoking. He found that messages that emphasize the financial gain from cigarette cost savings are more effective than loss-oriented messages, such as 'if you continue to smoke, you will lose money'. Moreover, he found that addressing the financial advantage is more effective than addressing health aspects such as longevity 'if you quit smoking, you will live longer'. In some of the studies the amount of information and number of instructions was varied: Carter et al. [40, 41] conducted studies on mass decontamination and found that the highest level of compliance was achieved when the public was provided with both health-focused information and practical information. Osman et al. [43] found that health-related information about smoking increases the tendency to advocate regulations on smoking. Several other studies exist in which message framing was tested for a particular purpose, see the overview in [39].

The examples from my literature research show that message framing can indeed influence behavior of individuals and crowds. However, it depends on the use case and the wording whether message framing is particularly effective. It is unclear how message framing can be employed via mobile applications in a crowd guidance system. It is, therefore, necessary to investigate how redirection recommendations should be phrased to foster compliance.

2.1.4. Pedestrian and crowd sensing technologies

Crowd-sensing is the task of measuring at least one crowd characteristic [120]:

- Presence: Is someone present?
- Count: How many people are present?
- Location: How are pedestrians distributed in a space?
- Track: How do pedestrians move?
- Identity: Who are the pedestrians?

Visualization-based approaches

Several technologies are available to detect, count, track, and monitor pedestrians and crowds. "One of the most commonly employed solutions to count, track, and analyze pedestrian activity in public spaces is the use of images from (surveillance) cameras" [7, p.79]. Optical cameras, thermal cameras, and LiDar-based systems are categorized as visualization-based technologies. All visualization-based approaches have in common that crowd characteristics are derived from image data for which computer vision or machine learning techniques are used, see Tab. 2.3.

Image descriptors	Classifiers
Histogram of oriented gradients	Support vector machines
Local Binary Pattern	Neuronal Networks
Haar-like feature	Gaussian (Mixture) Model
Viola-Jones feature	...
...	

Table 2.3.: Examples of image descriptors and classifiers used to gain information about a crowd from images. Application examples can be found in [121].

Non visualization-based approaches

Non-visualization-based technologies estimate crowd properties by using radio signals, temperature, or sounds, see Fig. 2.6. The approaches can be classified as device-free or device-based.

Non-visual sensing approaches			
Device-free		Device-based	
Radio Frequency	Others	Acoustic	Network
• Received Signal Strength Indication approach • Channel State Information approach • Ultra-wide band radar approach • ...	• Infrared • CO2 concentration • LiDAR • ...	• Tones broadcast • Speech detection • ...	• Radio Frequency Identification • Wireless Sensor Networks • Bluetooth • Wi-Fi • Cellular • ...

Figure 2.6.: Categorization of non-visualization-based sensing technologies. Own graphics inspired by [122].

Device-free approaches do not require crowd members to carry special sensors or devices [122]. Some of these device-free approaches employ the radio frequency band. In the *Received Signal Strength Indication* approach, the sensor measures the power of a radio signal of a known radio infrastructure node such as WLAN access point. Since pedestrians hinder wave propagation, it is assumed that the number of pedestrians is

high when the power is low and vice versa. The same principle is used in the *Channel State Information* approach. Here, however, signals are transmitted on multiple frequencies (OFDM), which allows the power level and, thus, the number of people to be determined more precisely [123]. So-called *ultra-wide band* radar sensors also work with radio frequencies. They estimate the number of people by emitting an impulse signal and analyzing its reflection [124]. Some other approaches do not use the radio band. These include *infrared* sensors, which correspond to the industry standard among device-free methods [122]. They either detect the body temperature (pyroelectric infrared sensor) or act as a light barrier: when the light beam is interrupted, the count is increased. Although simultaneous crossings are not recognized, the number of people is only underestimated by 10% on average [125]. Some procedures estimate the number of people based on the CO_2 concentration in a room [126]. Others are based on LiDAR or use a combination of technologies (see e.g. [127]). An overview of device-free approaches can be found in [122].

Device-based approaches require people to carry a sensor or device that is recognized [122]. Therefore, the accuracy depends on the participation. To tackle this, a procedure to estimate the participation has been proposed, see [128]. Unfortunately it is based on collected data and cannot measure the current participation. Most device-based approaches use network technologies to count people. Technologies employing Radio Frequency Identification (RFID) require that the crowd members carry RFID tags. Hidayat et al. [129] propose an RFID-based system for tracking pilgrims during the Hajj festival. Another technology is Near Field Communication (NFC). A device with NFC functionalities sends a signal to the reader if they get in proximity. Bluetooth-based technologies employ the same principle. Weppner et al. [130] installed Bluetooth scanners at a bridge to detect spectators in Zurich, Switzerland. They found that the motion pattern of the crowd can be determined from a small number of mobile nodes. Versichele et al. [131] used Bluetooth to estimate the spatio-temporal distribution of a crowd at a festival. Den Heuvel et al. [132] use Bluetooth scanners to investigate route choice behavior and waiting time at stairs and escalators at Utrecht Central Station (Netherlands).

Another device-based approach is based on WiFi: At a WiFi Access Point the number of Medium Access Control (MAC) addresses is counted. It is assumed that each count is one pedestrian. Such systems are in operation at Amsterdam Airport Schiphol[1] and in public transport stations in London[2]. However, the estimates have become inaccurate after procedures were introduced that randomize the medium access control addresses regularly. Active research tries to improve the accuracy [133, 134]. Other approaches rely on the cellular network. Shibata et al. [135] propose to estimate the crowd density from the signal strength of the uplinks from smartphones in an LTE network. Di Domenico et al. [136] propose a procedure similar to the concept that is already applied in device-

---

[1] Amsterdam Airport Schiphol. Royal Schiphol Group Privacy Statement. Available online: https://www.schiphol.nl/en/privacy-policy/ (accessed on 13 Jan 2024)

[2] Transport for London. Review of the TfL WiFi Pilot. Available online: http://content.tfl.gov.uk/review-tfl-wifi-pilot.pdf. (accessed on 12 February 2019).

free approaches: They estimate the number of pedestrians based on fluctuations of signals. There is also an approach suggested that is based on direct communication: UrbanCount [137] is a high layer protocol that has been proposed to estimate the total number of pedestrians in an area. It does not require a certain technology and has not been implemented in practice so far. It is unclear whether the protocol can identify crowd congestion.

Range and accuracy

Due to the different technologies, the range and accuracy of sensing approaches differ strongly, see Tab. 2.4. Infrared sensors have the shortest range, except for near-field communication technologies and pressure sensors. Network-based technologies have a range of several hundred meters or even kilometers. However, these methods have the disadvantage that personal data is processed, which is why they are critical in terms of privacy in [138]. Direct communication technologies may not face this problem because there is no intermediate processing data.

Technology	Frequency	Max Range	User Cooper.	Accuracy	Privacy Issues
Optical camera	Visible	100 m	No	High	Critical
Thermal camera	Infrared	500 m	No	High	Moderate
LiDAR	Ultraviolet, Visible, Near Infrared	1 km	No	High	Moderate
Synthetic Aperture Radar	Ku/Ka band	10 m	No	High	Moderate
(Passive) Infrared Sensor	Infrared	10 m	No	High	Low
Acoustic	0.01-100 kHz	20 m	No	Low	Moderate
Impulse-Radio Ultra-Wideband radar	3.1-10.6 GHz	Low	No	High	Low
Radio Frequency Ident.	13.56 MHz	10 cm	Yes	High	Critical
Near Field Communication	13.56 MHz	10 cm	Yes	High	Critical
Bluetooth	2.4 GHz	50 m	Yes	Medium	Critical
WiFi	2.4/5 GHz	100 m	Yes	Low	Critical
LTE	0.8/1.8/2.6 GHz	10 km	Yes	High	Moderate
5G	Sub-6 GHz	10 km	Yes	Low	Moderate
5G	MMW band	1 km	Yes	Medium	Moderate
Pressure	None	Contact	Low	No	Low
Received Signal Strength Indicator/Channel State Information	Various	Variable	No	Medium	Low

Table 2.4.: Crowd sensing technologies. The upper part comprises visualization-based technologies like cameras. The lower part contains non-visualization-based technologies. The table data is based on the data presented in [138].

My review of crowd-sensing approaches shows that there are several technologies for measuring crowd characteristics. Importantly, there is currently no sensing approach

available that measures safety-critical crowd densities based on direct communication. I conclude that a novel approach is needed.

Detection of 'anomalies' and emotional states

Novel sensing approaches aim to detect anomalous behavior and emotions in a crowd. In [139], the authors suggest using a support vector machine to identify violent behavior in a crowd from video footage from surveillance cameras. To measure the stress of spectators during a festival, Bergner [140] uses wristbands with sensors that measure accelerations, changes in skin conductivity and temperature, and GPS-trackers. Baig et al. [141] suggest to infer emotions from motion behavior. Most approaches use machine learning to detect 'anomalies' in the behavior of crowds: See the overviews in [142, 143]. This is debatable since no standardized definition of anomaly exists: compare, e.g., [144] and [145]. In my investigates, I will, therefore, rely on established metrics based on density and flows to assess the current state of the crowd.

2.1.5. Suggested automatic crowd guidance systems

I understand automatic guidance systems to automatically provide recommendations for pedestrian crowds. Advanced Traveller Information Systems, as proposed in [146, 147], cannot be considered as crowd guidance systems because they are designed for individuals. The core of a crowd guidance system is the algorithm that generates a recommendation based on density and flow measurements. The algorithm is implemented and executed in the so-called controller. Please note that the component and the algorithm itself is referred to as 'controller' in engineering. I use the term 'algorithm' in my investigations to avoid associations with crowd control.

Ren et al. [148] simulatively investigate the performance of so-called *Proportional Integral controllers*. Their goal is that the density in front of a particular exit equals a desired density. The idea is to attract pedestrians to exits using sound or light emitted by so-called evacuation assistance. Each exit has an evacuation assistant assigned. Each evacuation assistant has an independent *Proportional Integral controller* assigned that computes the magnitude of the signal based on the difference between a desired and a measured density. Ren et al. [148] do not investigate whether people understand or comply with the light signals.

Gao et al. [36] switch signals on and off to reduce densities and travel times in a simulated evacuation scenario. Like Ren et al. [148], they use evacuation assistants that emit signals to attract pedestrians. *On-Off-controllers* dynamically adjust the signal strength. An attracting signal is emitted only when the density nearby is less than the target density. The signals of the evacuation assistants are independent. Hence, multiple exits could be recommended at the same time, or, in case of overall congestion, none of the exits

would be recommended at all. It is unclear whether people can understand the signal and follow it.

Lopez-Carmona et al. [149] intend to make the egress of a football stadium safer. They divide the stadium into cells and dynamically recommend exits based on the current density. Pedestrians are equipped with radio frequency receivers that indicate their target exist using a color scheme. The authors use a *Tabu Search* algorithm to find an optimal sequence of cells to reach the exit. The search-space is high dimensional: for every time step they can choose from one 2^{126} possible solutions that arise from the fact that they have several cells and exits. It is impossible to apply the optimized sequence to an arbitrate scenario. It is unclear whether their methodology can be transferred since they utilize a nonstandard objective function for the optimization. To model the compliance of the crowd, Lopez-Carmona et al. [149] introduce a parameter which they vary to assess the effect of the compliance.

Menner at al. [37] aim to guide a crowd at a cross-junction. To model the crowd locomotion they use simple flow equations. They assume that pedestrian streams can be controlled like valves. In their scenario, there are three routes, each assigned an arrow of varying size. They apply *Model Predictive Control* to find the optimal arrow sizes for each time step, employing quadratic programming. The authors assume that the flow size of a pedestrian stream linearly depends on the arrow size. This is a quite technical perspective, which does not reflect realistic human behavior.

In several studies, authors propose guiding strategies that aim to adjust the walking speed of pedestrians. Zhang et al. [35] propose a hierarchical control model to maximize the throughput. The basic idea is to employ a *State-Space Controller* that adjusts the flows due to a desired Level of Service [69]. They compute a desired flow from which they derive a walking speed recommendation. To inform pedestrians, they propose to use speakers and video displays. They assume in their simulation study that pedestrian are willing and able to adjust their walking speed. It remains unclear how walking speeds can be adjusted in real-life systems.

A similar approach that also aims to change the speed is adjusting the free flow speed [32–34]. The free flow speed is the desired speed of a pedestrian at which pedestrians would walk when the path is free. The free flow speed of a population is usually normally distributed with a mean value of $1.34\,\mathrm{m/s}$ and a standard deviation of $0.26\,\mathrm{m/s}$ [51]. If it is dense, and people cannot freely move, pedestrians' actual walking speed will be below the free flow speed. Hence, it depends on the environmental conditions whether the free flow speed has any effect. There are several suggestions [32–34] to employ *State-Space controllers* to adjust the free flow speed. I believe it is impossible to adjust a person's free-flow velocity in a real system.

Molyneaux et al. [150] propose to adjust pedestrian flows by dynamically adjusting the corridor width through a dynamic flow separator. As an algorithm, they use a simple *heuristic* to compute the desired position of the flow separator: the width of the corridor

is divided in proportion to the inflows. This concept significantly reduces travel time, which motivates me to use simple heuristics to improve the traffic situation.

Although there are several suggestions, none of them has been tested or used in practice. One reason might be that the approaches are tailored to specific scenarios and, therefore, cannot be transferred to arbitrary use cases. Another reason might be that the approaches have been simulated under the assumption that all people follow instructions which is wrong. It is unclear how they perform in a real system. I conclude that algorithms are needed that can be applied to arbitrary scenarios. They must be tested under realistic conditions, that is, not all people follow instructions.

2.2. Modeling crowds and pedestrian dynamics

This section belongs to the methods-related part of the state of the art that comprises approaches for the simulative assessment of crowd guidance systems. First, the state of the art of crowd modeling and simulation is presented. A hierarchical modeling approach is introduced that models crowd behavior at a strategic, tactical, and operational layer. Then, locomotion models are introduced that capture the mobility behavior at the operational layer. Next, route choice models from the tactical layer are discussed: These are essential for a crowd guidance system where pedestrians should be redirected. Also, modeling approaches that consider communication between agents and psychological effects are discussed. Finally, I provide an overview of simulators and model libraries that can be used for crowd simulations.

2.2.1. The hierarchical layer model

Hoogendoorn's hierarchical model [151] describes the interaction of pedestrian models on a strategic, tactical, and operational level, see Fig. 2.7 (left). The strategic level models the overall goal like, for example, a person wants to leave a building as quickly as possible because there is a fire. The tactical level describes the scheduling of activities: a person leaves the office, takes the stairs, and leaves the building through the main exit. The movement behavior is modeled at the operational level. A weakness of a hierarchical modeling approach is that layers cannot always be clearly distinguished [152].

There are several suggestions how Hoogendoorn's hierarchical model [151] can be refined. Haghani [153] divides the operational layer into a 'step taking' and a 'local path finding' layer. The tactical layer is divided into 'exit choice' and 'exit choice change' and the strategic layer into 'room choice' and 'reaction time'. Seitz [154] replaces the tactical and operational layers with a social, psychological, and physical layer, see Fig. 2.7 (middle). His approach was driven by the need to incorporate collective behavior.

Kleinmeier [155] builds on Seitz' [154] approach and divides the psychological layer into three sub-layers: a perception layer, a cognition layer, and a behavior layer. The basic

idea is that pedestrians perceive stimuli that are sequentially processed in the three layers, see Fig. 2.8. The perception of an individual is modeled at the perception layer. At the cognition layer, the stimulus is processed which results in the assignment of a self-category. Based on this a behavior is selected. A similar approach can be found in Pan's dissertation thesis [156], who also proposed three layers: sensing, decision-making, and behavior selection.

Importantly, none of the proposed refinements fits any needs. Which hierarchical modeling approach should be chosen, depends on the application scenario and the research question.
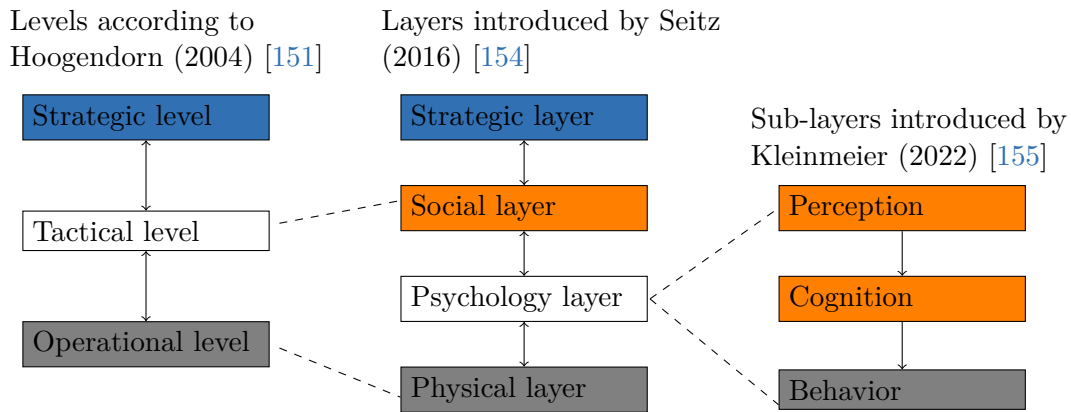
Figure 2.7.: Hierarchical modeling approaches for crowd behavior. The hierarchic model approach introduced by Hoogendoorn [151] has been refined by Seitz [154] and Kleinmeier [155]. The refinements allow to model collective, social and psychological behavior.
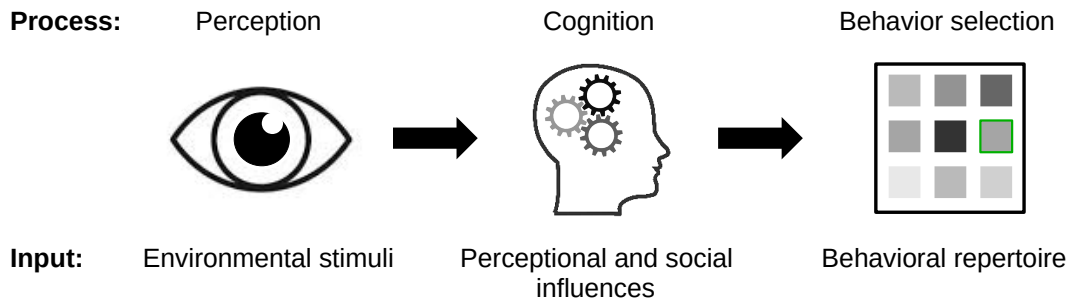
Figure 2.8.: Modeling perception, cognition, and behavior selection. At the perception layer, the most important stimulus is selected. At the cognition layer, a self-category is selected based on the most important stimulus and environmental conditions. From the self-category, a behavior is selected at the locomotion layer. Own graphics inspired by [155, p.89].

2.2.2. Modeling locomotion at the operational layer

The movement behavior of a crowd is modeled at the operational level. The main inputs of a locomotion model are a discretized topography with specified start and destination areas, and the crowd size.

The granularity of the simulation output depends on the type of simulation model. Models can be divided into microscopic, mesoscopic, and macroscopic models [49]. Macroscopic models describe the flows of people as a continuum using macroscopic equations. This model group provides macroscopic output variables such as flows and densities. In a microscopic model, each pedestrian is modeled individually as so-called agent. Models from this category provide agent-specific output data, such as trajectories or individual waiting times. Mesoscopic models combine microscopic and macroscopic properties. Many of them are multiscale models [157]: Microscopic models are used in parts of the topography where detailed pedestrians locomotion is needed. Macroscopic models provide pedestrian flows in between these areas. Neither macroscopic nor mesoscopic models can capture psychological behavior in detail [155]. In a crowd guidance system it is essential to model peoples' responses to route recommendations. Therefore, microscopic models are required.

Several microscopic locomotion models have been developed in the last decades. So-called *cellular automatons* [158] are zero-order models that consider only the current state ('zero order of the derivative') in the propagation of the movement. The topography is discretized as a grid where each cell is either empty or occupied by a single agent. If a target cell is free and not a target of another agent, a move is executed. If multiple agents share the same target, the agent that has the higher probability is assigned to the cell. The so-called floor field modifies the probabilities to ensure that moves towards the target are preferred [159, 160].

The *Optimal Steps Model* [161] is a zero order model that is continuous in space and time. It was validated in several works [161–164]. The basic idea is that agents try to improve a utility with every step [161]. The utility of each position in space is coded in a scalar function, also called floorfield. The utility, or potential, depends on the geodesic distance to the target and the proximity to other agents, see Fig. 2.9. Utility increases when approaching targets and decreases when getting too close to obstacles and other agents. Physically spoken, agents, when moving, are attracted by targets and repulsed by obstacles and other agents. A dynamic floorfield can be used to model long- and medium-range interactions [165]. Sivers et al. [166] modified the potential functions based on Hall's theory of interpersonal distances.

So-called *Velocity-based Models* are first-order models that consider agents' position and speed (first derivative). In contrast to the cellular automaton, this model type is continuous in space. Velocity-based models are either based on the concept of 'optimal velocities' or the concept of 'velocity obstacles'. In the 'optimal velocities' approach,

Static floorfield potential   Obstacle potential

Agent potential

Agent (1) steps to the position where the superposed potential is minimal within the step circle (dashed).
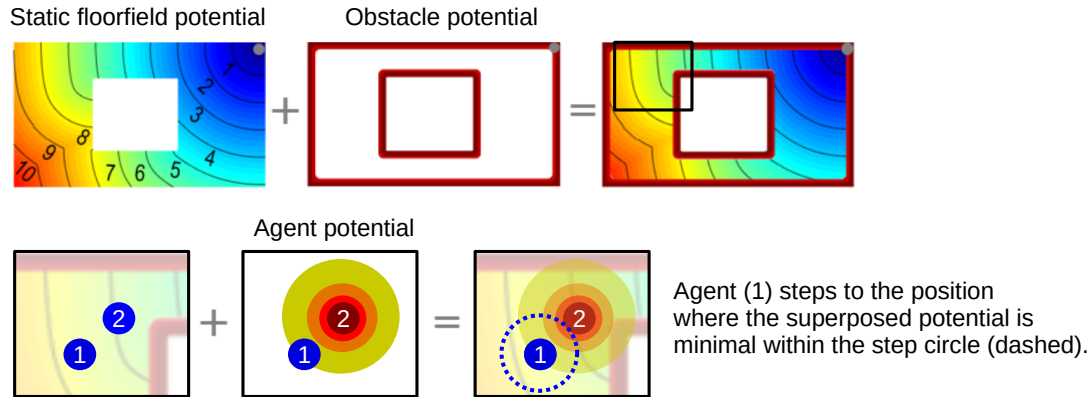
Figure 2.9.: Basic idea of the Optimal Steps Model [161]. Agents use the shortest path to a target (floorfield potential) while skirting obstacles (obstacle potential) and other agents (agent potential). The potentials are superposed. For each step, the optimal position needs to be found, that is, where the potential has a minumum within a step circle.

agents' velocity is adjusted using an optimal velocity function that depends on the minimum spacing in front [167]. In the 'velocity obstacles' approach, the orientation and the speeds are varied to find a configuration of agents where they do not collide [168].

So-called *Force Models* are second-order models that employ accelerations (second spatial derivative) and force terms to model repulsion between agents and attraction to target. The most popular force model is the *Social Force Model* [169, 170] for which numerous extensions exist, see e.g. [171]. Social Force Models are widely used in the pedestrian community despite numerical instabilities that can lead to oscillations [172].

Hybrid models combine the advantages of velocity-based and force-based models. This modeling approach includes the *Gradient Navigation Model* [173] where velocity and acceleration are considered in a decoupled manner: The position of the next step only depends on velocity-based quantities, but the magnitude of the velocity vector depends on an acceleration term.

So-called *Decision-based or Decision Models* form another model category. These use simple rules for locomotion, such as 'slow down', 'keep a certain speed', or 'accelerate' [174]. One of them is the *Behavioral Heuristics Model* [162] that has four rules: (1) perform a step towards the target if the position is free. (2) If occupied try to evade tangentially. (3) If the tangential position is occupied perform a side step. (4) If the side is occupied wait.

Which locomotion model is most suitable for an applications depends how much computational effort one is willing to invest and on the required accuracy. Space-continuous models, such as force models or optimal steps models, produce more realistic trajectories than models with spatial discretization, such as the cellular automaton. However, such

sophisticated models are computationally expensive: In the Optimal Steps Model an optimization problem needs to be solved for each step. The social force model requires solving a system of differential equations. If the scenario is large and low accuracy is acceptable, space discrete models, such as the cellular automaton, are still in use. Again, there is no model that fits all needs.

2.2.3. Modeling route choice at the tactical layer

Different types of models can be found at the tactical level: models for queuing behavior (see, e.g., [175]), models for avoiding crowds (see, e.g., [176]) , and small group models (see, e.g., [177]). Also route choice behavior is modeled at the tactical layer for which one can differentiate two model types, see Fig. 2.10.

Link choice models, or traffic assignment models, are used in traffic engineering to model the flow distribution at junctions [178]. They model the decision of the entire population depending on the current location in the traffic network. The input of such a model is the network topology and an origin-destination matrix. The output of the model is various route assignments for each origin-destination pair. The assignments are determined by optimizing the route assignments so that the average travel time [179] or an alternative objective becomes minimum.
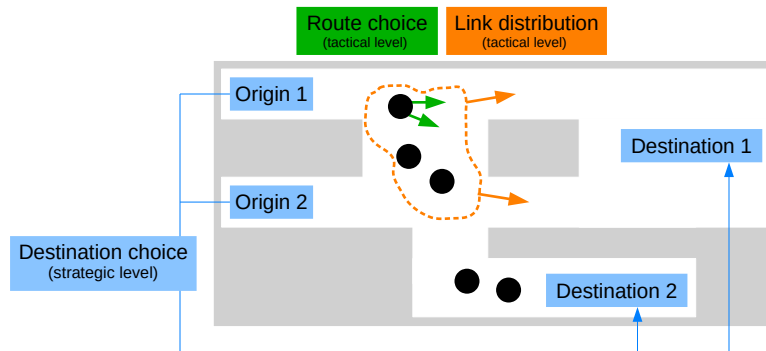
Figure 2.10.: Relationship between strategic and tactical layer models. Route choice models capture individual route choice, while link choice models simulate the route distribution of a population. So-called destination [180] and location choice models [181] belong to the strategic layer.

Route choice models reflect the decision-making process of an individual. Therefore, they belong to the cognition layer of Kleinmeier's psychology layer [155]. Numerous experiments have been conducted to identify factors or stimuli that affect the route choice, see e.g. [182–184]. Many route choice models are multi-logit models that make a route decision based on environmental and exogenous factors [149]. Shi et al. [185] propose a conditional multinomial logit model to model the route choice behavior of metro passengers. Daamen [186] proposes a logit model that models route choice with

the factors of walking and waiting times, route lengths, and walking comfort. Xu et al. [187] suggest a model that considers the distance to the exits and their attractiveness. Ramos et al. [188] use the travel time as a factor in their logit model. Tong et al. [189] extend a logit model proposed by Arganda et al. [190] by adding a factor that has the effect that people become "less sensitive to environmental information the more decisions they have to make". In a second study, Tong et al. [191] propose to add another factor that models the so-called 'route commitment effect', that is, "the more pedestrians invest into a planned route by walking along it, the more likely they are to adhere to this route even when encountering congestion". Other logit models consider stairs [192], visibility conditions [193, 194], the tendency to stick to a route [195], or the effect of instructions from human guides [196]. Apart from route choice models, machine learning has been proposed to emulate route choice behavior, see e.g. [197].

Few approaches try to model people's compliance to instructions: Lopez [149] split agents into two groups. For non-compliant agents, they use a route choice model. Compliant agents follow the instruction, ignoring other factors. Haghani et al. [195] use another approach: They implicitly model compliance by introducing a factor called 'inertia' to a logit model that biases the decision.

As one can see there are several suggestions to model route choice behavior. All of the proposed models were derived for different factors and scenarios. Consequently, the models cannot be simply transferred to arbitrary scenarios. Moreover, there is no approach that models the compliance with route recommendations provided by a mobile application. Therefore, a new modeling approach is needed.

2.2.4. Modeling of communication and psychological factors

How people respond to information depends on how information is communicated, see e.g. [41]. There are different approaches to model communication aspects. Templeton et al. [198] distinguishes four principle modeling strategies:

- Field of influence: Agents have a bounded field of influence around them, which transmits information to other agents. See e.g. [199].

- Network-based: Agents share information with others within their social network. See e.g. [200].

- Information trails: Agents disseminate information via trails in the environment as they move. See e.g. [201].

- External information: Agents in particular areas become informed by an external source. See e.g. [202].

Some other approaches aim to model the effect of the emotional state of an individual. Pelechano et al. [203] propose a model in which the decision-making process depends on the emotional state affected by stress and physiological factors. Bosse et al. [204]

model the effect of beliefs, desires, and intentions. Kielar [205] takes memory and individual preferences into account. Wijermans [206] considers the effect of the physiological constitution.

Some approaches focus on modeling social identities (Section 2.1.3). In her dissertation, Sivers [207] modeled the effect of social identities on helping behavior. Kleinmeier [155] investigated how cooperative behavior between group members can be modeled using self-categories.

The brief overview shows that there are indeed approaches for modeling psychological changes and the communication in a crowd. However, it is unclear how one can model the effect of route recommendations that are communicated via mobile applications in combination with social identities in a crowd guidance system.

2.2.5. Simulation frameworks

Several open-source simulation frameworks are available to simulate crowd behavior, see Tab. 2.8. Some of them are no longer maintained. Most of them operate at the operational layer, that is, providing locomotion simulation. The only simulator providing a psychology layer is the *Vadere* simulation framework [155].

License	Simulator	Language	Reference	Maintained
LGPL	JuPedSim	C++,Python	[208]	
LGPL	Vadere	Java	[209]	
GPLv3	GAMA	Java	[210]	
MIT	Agents.jl	Julia	[211]	
LGPL v3	jCrowdSimulator	Java	[212]	
GNU GPL	Cromosim	Python	[213]	
Apache 2.0	Mesa	Python	[214]	
GPLv3	PedSim	JavaScript	[215]	no
Apache 2.0	Menge	C++	[216]	no
-	FDS+Evac	Fortran	[217]	no
-	MomenTUMv2	Java	[218]	no

Table 2.5.: Overview on open-source software for crowd simulations. The list is based on Richard's software review [219]. I consider simulators as no longer maintained when the last commit was before Dec 2022 (checked: 18 Jan 2024).

2.3. Modeling mobile networks and direct communication

This section belongs, like the previous section, to the second part of the state of the art. I present the state-of-the-art of modeling and simulating direct communication in mobile networks. First, standards on mobile communication and their respective protocol stacks

are introduced. Then, an overview of channel modeling is given. Finally, I present state-of-the-art system level simulators. I assume that the reader is familiar with the ISO/OSI reference model and the basic principles of radio wave propagation such as "free-space (or line-of-sight) propagation, transmission (and absorption), specular reflection, diffraction, and diffusion (also known as diffuse scattering) [...]" [220]. If not, I recommend to [220].

2.3.1. Overview of mobile standards enabling direct communication

For enabling device-to-device (D2D) communication, two main technologies have emerged: dedicated short-range communication based on the IEEE 802.11p standard [221] and cellular vehicle-to-everything (C-V2X) communication based on several releases of the 3rd Generation Partnership Project (3GPP), see Fig. 2.11.

Figure 2.11.: Protocol stack reference model for direct communication. Standards for direct communication specify the lower layers of the protocol stack. Own graphics, inspired by [222, p.89].

Dedicated Short Range Communication

The *IEEE 802.11p* standard [221] was the first mobile standard enabling Device-to-Device (D2D) communication. It specifies Dedicated Short Range Communication (DSRC) on the physical layer and on the medium access control (MAC) layer. At the physical layer, the Orthogonal Frequency Division Multiplexing (OFDM) is specified for the modulating of data on multiple carrier frequencies. At the medium access control (MAC) layer, the standard specifies the Enhanced Distributed Channel Access (EDCA) method that employs carrier sense multiple access with collision avoidance (CSMA/CA): Before a packet is sent, the network node checks whether the transmission channel is free. If the channel is busy, the node waits for a randomly defined period of time and then checks again. Therefore, participants access the channel in a self-organized, decentralized way. However, this requires that devices recognize the channel occupation

correctly and do not start accessing the channel simultaneously. Therefore, the performance depends on the density of participants. The more participants, the more likely are collisions and interference that lead to packet loss [223]. It cannot be guaranteed that packets arrive within a certain time [223].

As the 802.11p standard only applies to the physical and medium access control layer, the standard is combined with other standards to enable Dedicated Short Range Communication. In the context of Intelligent Traffic Systems (ITS), there are the IEEE 1609 standard [224] (USA), the CEN/ETSI EN302 663 standard [225] (Europe) and the ARIB STD-T109 standard [226] (Japan). An overview of the country-specific specifications can be found in [227]. The transmission is intended to operate in the frequency range 5855 MHz to 5925 MHz.

The *IEEE 802.11bd* standard [228] is an amendment of the set of 802.11 standards. It was released in 2023, and it is interoperable with the 802.11p technology. The amendment aims to improve throughput, latency, reliability, and range through new specifications for the physical layer and the medium access layer [229]. Therefore, 802.11bd-based technology is more advanced than *IEEE 802.11p* technology. The bandwidth of the channel is $20\,MHz$ which is twice the bandwidth of the IEEE 802.11p standard. Like the IEEE 802.11p, the amendment specifies the EDCA method for accessing the channel. Due to the larger bandwidth and adjustments regarding the error correction scheme, more packets can be transmitted [229]. Although the 802.11bd amendment has improved network performance compared to the 802.11p standard, 802.11bd-based technology can often not outperform Cellular Vehicle-to-everything communication, as demonstrated in [3].

Cellular Vehicle-to-everything communication

Cellular Vehicle-to-everything communication (C-V2X) has been developed by the Third Generation Partnership Project (3GPP), see Fig. 2.12.

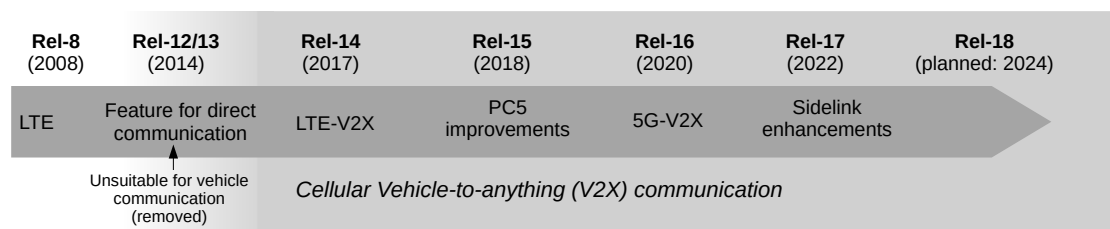| Rel-8 (2008)	Rel-12/13 (2014)	Rel-14 (2017)	Rel-15 (2018)	Rel-16 (2020)	Rel-17 (2022)	Rel-18 (planned: 2024)
LTE	Feature for direct communication	LTE-V2X	PC5 improvements	5G-V2X	Sidelink enhancements	
	Unsuitable for vehicle communication (removed)	*Cellular Vehicle-to-anything (V2X) communication*				

Figure 2.12.: Standards of the 3rd Generation Partnership Project (3GPP) for Cellular Vehicle-to-everything communication. Own graphics; inspired by [230].

LTE V2X, that is Vehicle-to-everything communication over Long Term Evolution networks, is specified in the technical reports TR 22.885 [231] and TR 36.885 [232] as well as in the technical specifications TS 22.185 [233], TS 23.285 [234], TS 36.213 [235], TS

36.321 [236] released by the 3GPP. Direct communication is specified over the PC5 interface using so-called sidelinks. The sidelink uses the same band for communication as the Dedicated Short Range Communication: 5.9 GHz. Two communication modes exist: the controlled mode (mode 3) and the autonomous mode (mode 4), see Fig. 2.13. In mode 3, a base station allocates resources. In this mode it is not specified whether dynamic or semi-persistent scheduling is employed [230]. For the autonomous mode a semi-persistent scheduling procedure is required: each network participant allocates frames and sub-channels at a fixed time interval based on channel occupancy estimates [230].
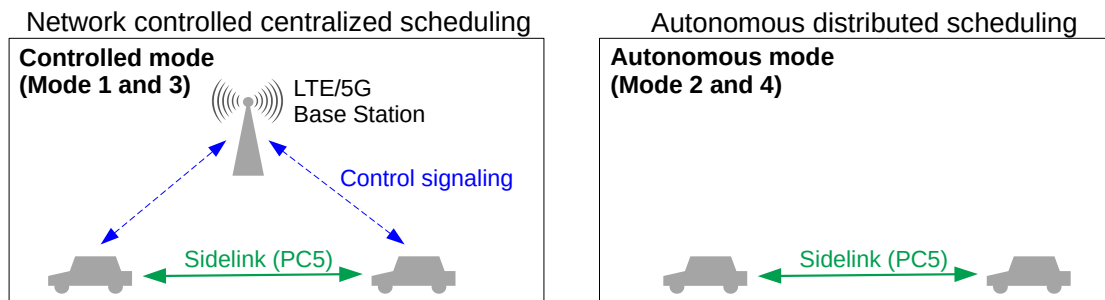

Figure 2.13.: Modes for cellular vehicle-to-everything communication. Vehicles (or devices) share data directly over the PC5 interface using so-called sidelinks. In mode 1 (5G) and 3 (LTE) the base station allocates resources to reduce the risk of collision and interference (left). In mode 2 (5G) and 4 (LTE), vehicles schedule packet transmissions autonomously [230].

5G V2X or *NR-V2X*, where NR stands for 'New Radio', means communication over sidelinks in networks of the fifth generation (5G). The 5G sidelink is defined by the 3GPP in the documents TR 22.886 [237], TR 38.885 [238], TS 23.287 [239], TS 22.186 [240], TS 38.213 [241], TS 38.321 [242] and TS 37.895 [243]. 5G sidelinks and LTE sidelinks differ in their waveform, the scheduling interval, and the channel coding procedure, see Tab. 2.6. As with LTE sidelink communication, one can distinguish a controlled (mode 1) and autonomous mode (mode 2). Importantly, the 5G sidelink operates at two frequency bands: 0.41 GHz-7.125 GHz and 24.25 GHz-52.6 GHz. For the latter, the wavelength is at the scale of a millimeter, which is why it is referred to as 5G millimeter waves (mmWaves).

Overview on technologies and interoperability

An overview of specifications and technologies can be found in Tab. 2.6. Importantly, dedicated short range communication technologies and cellular technologies are not interoperable [244]. Even in a 'hybrid system', the technologies operate simultaneously but independently [223]. The 802.11bd standard [228] is compatible with its predecessor

IEEE 802.11p. The LTE sidelink and the NR sidelink are basically not interoperable, but they can co-exist [230].

Standard	802.11p	802.11bd	LTE V2X	5G/NR V2X
Technology	WLAN	WLAN	4G/LTE	5G
First release	2012	2023	2017/18 (3GPP Rel-14/15)	2020 (3GPP Rel-16)
Wave form	OFDM	OFDM	SC-FDMA	OFDM
Ressource allocation	decentral	decentral	central (Sidelink mode 3), decentral (Sidelink mode 4)	central (Sidelink mode 1), decentral (Sidelink mode 2)
Modes	Broadcast	Broadcast, groupcast	Broadcast	Broadcast, groupcast, unicast
Range	short range	short range	extended range	extended range
Scheduling interval	None	None	One sub-frame	Slot, mini-slot or multi-slot
Channel coding	Binary Convolutional Coding (BCC)	LDPC	Data: Turbo, Control: Convolution	Data: LDPC, Control: Polar
Frequency band	5.9 GHz	5.9 GHz/ 60 GHz	5.9 GHz	0.41 GHz-7.125 GHz, 24.25 GHz-52.6 GHz (mmWave)
Sub-carrier spacing	156.25 KHz	312.5 KHz, 156.25 KHz, 78.15 KHz	15 KHz	15 KHz, 30 KHz, 60 KHz, 120 KHz
Retransmission	None	Congestion dependent	Blind	HARQ-based

Table 2.6.: Overview of standards for direct communication. The IEEE technologies are not interoperable with cellular technologies. The table is based on the data from [223, 230, 244].

2.3.2. Modeling wireless propagation channels

For a mobile network simulation, it is essential to model the propagation channel. "A wireless propagation channel is the medium linking the transmitter and the receiver. Its properties determine the information-theoretic capacity, i.e., the ultimate performance limit of wireless communications, and also determine how specific wireless systems behave. It is thus essential that we know and understand the wireless propagation channel and apply this knowledge to system design." [245, p.45].

Molisch [245, p.125ff] distinguishes three basic modeling approaches:

- 'Deterministic' channel modeling: solve Maxwell's equation to compute channel characteristics. Since solving the integrals or differential equations is time-consuming, ray-based methods are used in practice. This approach is often applied in network planning and system deployment.

- 'Stochastic' channel modeling: use models to estimate statistical properties of the channel characteristics. This approach is often applied when designing and comparing systems.

- Use of measurement data: use empirical data to describe the channel characteristics. The execution of a model is not necessary. This approach is often applied when measurement data is available for a specific scenario.

Channel models can be classified according to their temporal and spatial resolution [220, p.70]: so-called large-scale propagation models predict the average signal strength over the distance and are typically used to estimate the coverage. Small-scale or fading models model the signal strength at a fine local (wavelengths) and temporal (seconds) scale [220, p.70].

Four channel model categories are differentiated [245], see Fig. 2.14. Narrowband models describe the signal attenuation. Wideband models focus on the dispersion of the signal. Directional models capture directional properties of signal components, which is why these models are suitable for the simulation of multi-antenna systems [245, p.445]. 'Deterministic' models are based on Maxwell's theory of wave propagation.
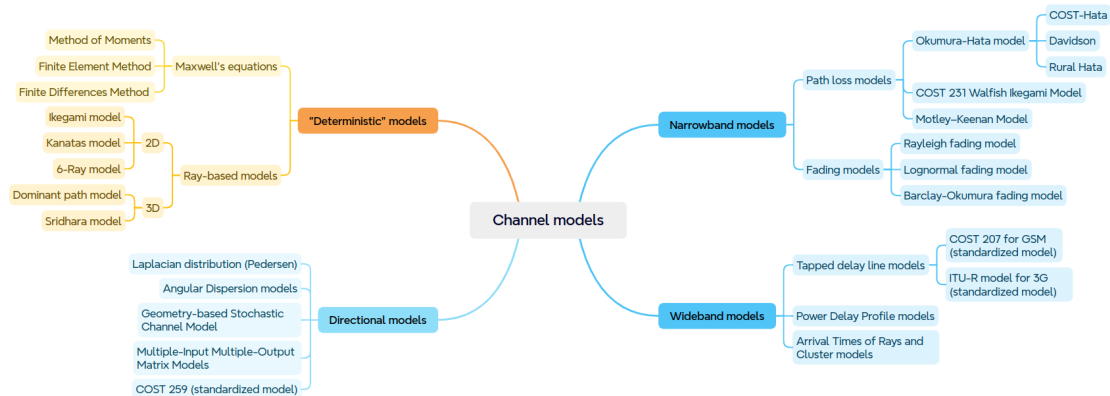

Figure 2.14.: Classification of channel models according to Molisch [245]. The model examples come from Molisch [245] and Phillips et al.[246]. Own graphic inspired by [245, 246].

Apart from that, there are different approaches how a channel can be represented as model. Phillips et al. [246] distinguishes several categories: (1) Theoretical models are derived analytically from the theory of ideal electromagnetic propagation. An example

is the free space path loss model according to Friis [247] that describes how the power decreases quadratically over the distance between transmitter and receiver. An extension of Friis' model is the Two-Ray Ground Reflection model that considers reflections from the ground [220, p.95]. The second category is formed by so-called (2) basic models that are derived from empirical data. The Okumura-Hata Model [248] describes the path loss depending on the distance between sender and receiver, the frequency, and the height of the base station or mobile station. The model is intended for scenarios with an elevated station, e.g., at a rooftop. Numerous extensions exist, such as the COST Hata Model, which is applicable even for frequencies above 1500 MHz [249].

For outdoor scenarios so-called (3) terrain models are used. They are similar to basic models but also consider diffraction. Stochastic fading models (4) add a random variable to capture fading. Models from this category are typically used to design the physical or link layer [246]. One of them is the Lognormal Shadowing Model, which can be considered a stochastic version of Friis model with a variable decay parameter. Stochastic fading models, such as the Rayleigh Fading model [250], model small-scale effects. Ray-based models (5) try to capture the effect of wave propagation in the entire space using rays. Supplementary models (6) adjust the results of other models. Many of these models focus on adjusting the signal propagation due to obstacles such as trees or walls, see e.g. [251–253]. The so-called ideal obstacle loss model [254, p.75] models the extreme case: If an obstacle is in the line-of-sight, the signal strength is zero. More model examples can be found in [246].

Standardized channel models are established models that are used by the mobile networks community to develop higher layer protocols and applications. They are released by standardization institutions such as 3rd Generation Partnership Project or IEEE, see Tab. 2.7. The models have been extensively calibrated or validated using empirical data and simulation data, see [255–264] and their references. I do not want to develop direct communication technologies further, but I want to test their suitability for crowd management applications. Therefore, I employ standardized models for my investigations.

2.3.3. Simulation frameworks

System-level simulators provide, in contrast to link-level simulators, implementations of the full protocol stack. An overview of simulators can be found in Tab. 2.8. Only the two simulation frameworks *OMNeT++* [265] and *ns-3* [266] have established themselves as scientific open-source software. The frameworks and their related model libraries provide implementations for standardized channel models including the channel. For example, the 3D channel model TR 36.873 [258] is available in the *simu5G* [258] simulator that belongs to the *OMNeT++* ecosystem [267].

Different studies demonstrate that the mobility behavior of mobile nodes has a major impact on the mobile network simulation: Wischhof et al. [30] demonstrated that the results of the mobile networks simulation depend on the accuracy of the locomotion

Standard	Year	Channel model	Gremium
802.11n	2004	802.11n channel model [255]	IEEE
LTE	2007	WINNER model [256]	WINNER project
LTE	2003	3D Spatial Channel Model (TR 25.996) [257]	3GPP
LTE	2017	3D Channel Model (TR 36.873) [258]	3GPP
LTE	2012	COST 2100 model [259]	COST 2100
5G	2017	ITU-R channel model [260]	ITU-R
5G	2018	3GPP TR 38.901 [261]	3GPP
5G	2015	METIS [262]	European Commission
5G	2017	mmMAGIC [263]	Frauenhofer
5G	2019	QuaDRiGa [264]	Frauenhofer

Table 2.7.: Examples of standardized channel models. Several models are available for modeling radio channels for different network technologies. These models are typically used for the development of protocols and applications at higher layers, such as the application layer.

Simulator	License	Language	Reference	Maintained
OMNeT++	Academic public	C++	[265]	
ns-3	GNU GPLv2	C++, Python	[266]	
GloMoSiM	Academic public	Parsec, Java	[268]	no
LTE-SIM	GPL-3.0 license	C, C++	[269]	no
Jist/Swans	CRF	Java	[270]	no

Table 2.8.: Overview of open source software for mobile network simulations. The list is based on Khan's software review [271]. I consider simulators as no longer maintained when the last commit was before Dec 2022 (checked on 25 Jan 2024).

model. Bai et al. [272] found that path duration in ad hoc networks depends on the accuracy of node movement. They tested different mobility models. Chancay-García et al. [29] found that information dissemination depends on the degree of motion and message size. For this reason, *OMNeT++* and *ns-3* have an interface that allows one to use pre-generated mobility traces in the network simulation. It is also possible to obtain position data online from the SUMO simulator over an interface [273].

The SUMO simulator [273] provides simple pedestrian models where agents move, for example, along lanes like vehicles. These models are not validated against empirical data from pedestrian dynamics research. It cannot be ensured that they produce realistic trajectories. Consequently, safety-critical pedestrian densities cannot be evaluated reliably. I conclude that it is essential to use a validated locomotion model in the investigation of a crowd guidance system. Because several crowd models and simulators are already available (Section 2.2.5), I look at model and simulator coupling in the next section.

2.4. Model composition and coupling

In the previous sections (Sections 2.2-2.3), I introduced crowd models and mobile network models. These models are separate models that are usually executed independently from each other. For the simulative investigation of a crowd guidance system, these models need to be connected. In this section, I present state-of-the-art approaches how separate models and simulators can be coupled. First, model coupling is discussed for different system types. Next, sequential update procedures are introduced. Then, it is discussed how the data exchange between simulators can be realized. Finally, state-of-the-art simulation frameworks and interfaces are presented.

2.4.1. Coupling models and simulators

With model coupling, two different component models are connected so that they can exchange information with each other. What a model coupling looks like depends strongly on the structure of the system, see Fig. 2.15. In partitioned multi-physics systems, the model coupling is located at a geometric boundary. Both components share a system state at the boundary. To couple models, the *monolithic approach* or the *partitioned approach* can be used. "The monolithic approach, on one hand, uses a single system of equations to describe and solve the coupled problem. On the other hand, partitioned methods use existing single-physics solvers and couple them to a simulation that strives to solve the overall coupled problem" [274, p.16].
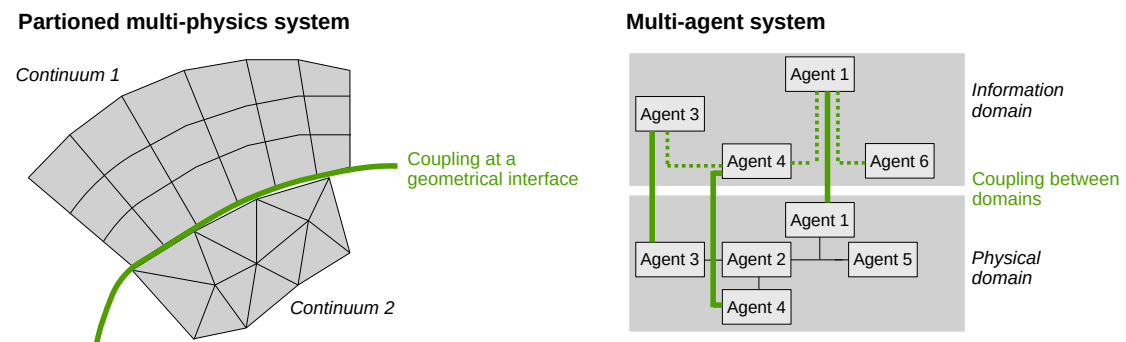

Figure 2.15.: Couplings for different system types. In a multi-physics system, interactions occur at a geometric interface (left). An example is an airplane wing: the wing is a mechanical structure that bends through the air that streams along its surface (geometric boundary). In a multi-agent system, the coupling connects the physical and the information domain. One example is a smart grid, where consumers and producers share information with the goal to improve the electrical supply. The topology of the two domains can differ (compare the two domains on the right). Own graphics; right side inspired by [275].

In a multi-agent system, the model coupling connects the physical domain and the information domain. This type of simulation is sometimes referred to as co-simulation [276]. The distinction of the two domains comes from the field of smart grid applications in which an electricity grid (physical domain) is controlled with the help of information transmitted via a network (information domain) [275, 277–279]. Crowd models have a similar hierarchical structure: The operational layer models the physics of crowd locomotion. Therefore it corresponds to the physical domain in Fig. 2.15. Providing route recommendations to a crowd at the tactical layer corresponds to the information domain.

Time stepping and update scheme

For any coupled system, the state needs to be exchanged between the models at certain times. There are various techniques for this, which differ in terms of their accuracy, stability and computational costs [280, p.44]. Both explicit and implicit methods are used in the partitioned approach, see [274, 280]. Explicit methods are widely applied to multi-agent systems [281]. Fig. 2.16 depicts two explicit techniques: the serial update scheme and the parallel scheme. Both methods are applied for multi-physics and multi-agent systems, compare e.g. [281] and [274]. In the sequential update, one model is executed first. The result is used to predict the state of the second model. With a parallel update scheme, both models are executed simultaneously. Explicit schemes can be unstable under certain conditions because one (or both) of the solvers use old values or an explicit prediction for the boundary values of the other solver [280, p.66]. Iterative techniques try to overcome this by introducing intermediate time steps. Nevertheless, they cannot ensure stability [280]. To my knowledge, it has not been investigated so far which update scheme is suitable for a composed crowd guidance system. It is, therefore, necessary to find a suitable procedure for coupling crowd and network models.
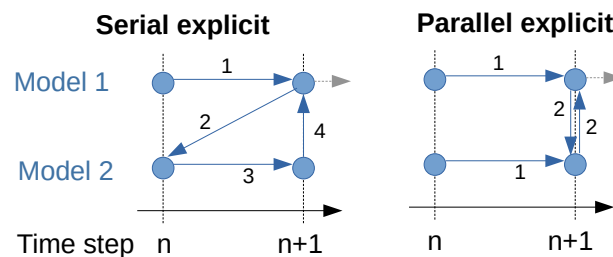

Figure 2.16.: Simple explicit update schemes used for multi-physics and multi-agent systems. In the serial update scheme (left), one of the models is evaluated first (1). The future state is passed to the other model (2) which uses the state for its prediction (3). The prediction is then passed back (4). In the parallel update scheme (right), the models are executed simultaneously.

Simulator coupling

If models are located in different simulators, communication between the two simulators must be realized in addition to model coupling. Following approaches to exchanging data between two coupled simulators exist [280]:

- File communication: simulators exchange data over files written to the harddisk.

- Socket communication: send and receive data over a network using communication protocols such as Transmission Control Protocol (TCP).

- Message Passing Interface: intended for the communication of parallelized software in distributed systems.

The advantage of the file communication approach is that it does not require external communication libraries. The disadvantage is that the speed of the data exchange depends on the harddisk properties. If a large amount of data needs to be shared, it can take an infeasible amount of time. Therefore, file communication is often used as a fall-back solution when other communication methods fail [280, p.153].

Socket communication means exchanging data in a network. To manage a socket connection so-called co-simulation frameworks can be used. They manage the connecting process of the sockets and the data exchange. If a middleware is used, the simulator coupling is called 'generic' [282]. Otherwise, the coupling is 'specific', see Fig. 2.17.
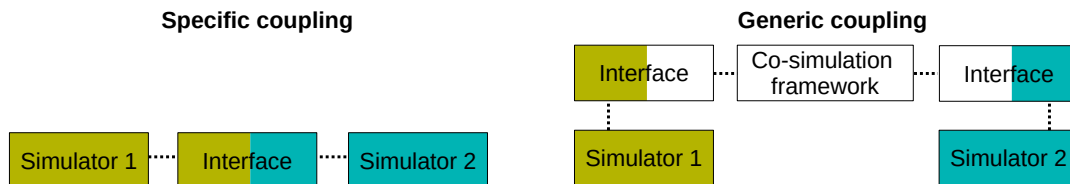

Figure 2.17.: Generic and specific coupling. With a specific simulator coupling, the sockets of the simulators communicate directly over sockets in a network (left). In a generic coupling, a middleware manages the inter-process communication between simulators over sockets (right). For both types of coupling, suitable interfaces must be implemented. Own graphics inspired by [282].

Validation of composite models

Finally, I would like to briefly address the validation of composed models, assuming that the reader is familiar with the concepts of verification, validation, and calibration. The so-called composability theory deals with the question "[...] whether validity actually is preserved by composition. In other words, if two models are separately valid, can it be assumed that their composition is necessarily valid?" [283, p.63]. In his dissertation,

Weisel [284] shows that a model, that is composed of two valid models, is not necessarily valid too. Only in some special cases, such as linear combinations, the composite model is valid under certain metrics [284].

Therefore, a composite model needs to be validated even when the component models have been validated. For this, conventional validation techniques can be used [283, p.76]. Nevertheless, validating a model requires that empirical data is available.

2.4.2. Co-simulation frameworks and interfaces

So far, one single crowd guidance system has been built using model and simulator coupling: CELLEVAC [149] for which the simulator AnyLogic was directly coupled with the commercial simulator Matlab.

Apart from that, several middleware frameworks have been developed, see Fig. 2.18. A middleware framework requires to implement interfaces, see again Fig. 2.17. Therefore, it does not have an advantage over directly connecting simulators in terms of software engineering effort. Coupling simulators directly keeps the number of software dependencies low. For this reason, many direct couplings can be found in practice, see Tab. 2.9.
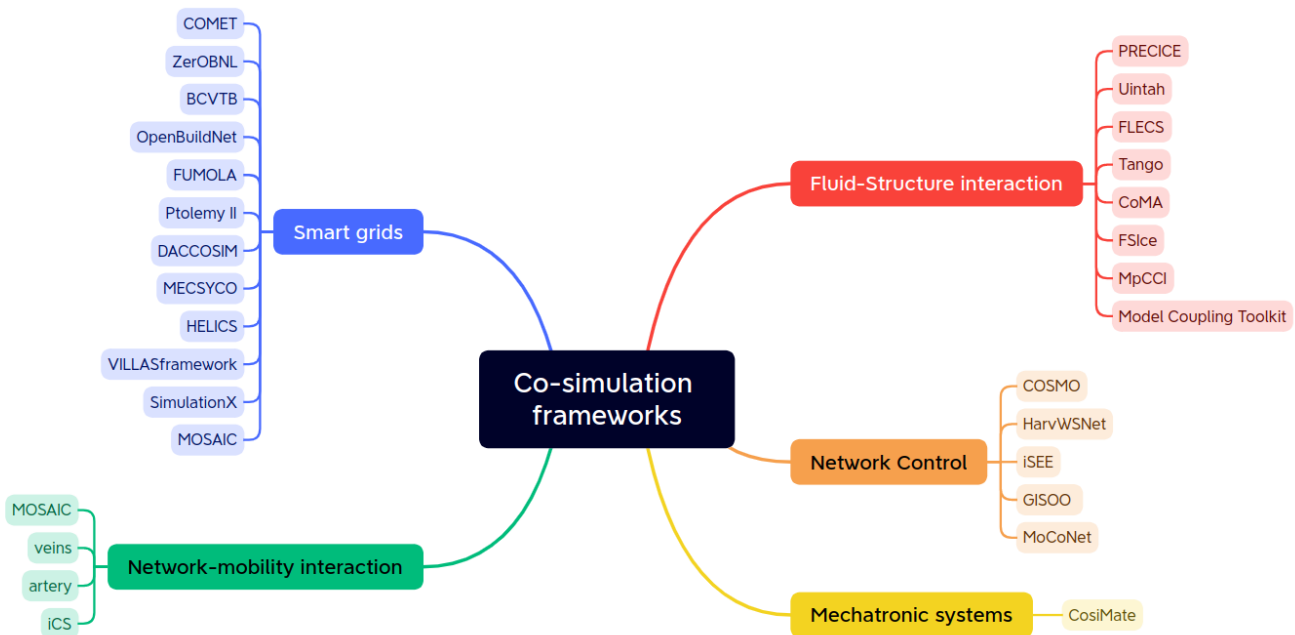

Figure 2.18.: Examples of co-simulation frameworks (middlewares). Co-simulation frameworks couple models from different simulation frameworks.

Framework	Mobility simulator	Network Simulator
Veins	SUMO	OMNeT++
Artery	SUMO	OMNeT++
iCS	SUMO	ns3
MOSAIC	SUMO, PTV Vissim	ns3, JiST/SWANS or OMNeT++

Table 2.9.: Simulator couplings. The MOSAIC framework employs a generic coupling approach. Artery, iCS and Veins implement the Traffic Control Interface in the respective simulators (specific coupling). There is also one coupling with the commercial simulator PTV Vissim.

For the specific coupling of mobile communication simulators and mobility simulators, the so-called Traffic Control Interface (TraCI) was introduced [273]. The Traffic Control Interface couples a mobility simulator with a mobile communication simulator. The simulators exchange data over the Transmission Control Protocol (TCP). The TraCI protocol defines the structure of a message that is composed of a header and the content, see Fig 2.19. The content is either a request for data (How many vehicles are in the simulation?) or a command (Change the velocity of vehicle with $id = 4$ to $20\,\text{km/h}$!).

So far, only couplings with the mobility simulator SUMO have been realized, see Tab. 2.9. The mobility models of the SUMO simulator are intended for large-scale scenarios: They cannot provide detailed trajectory data at a local scale. However, detailed data is necessary to evaluate the performance of a crowd guidance system. Therefore, the simulation frameworks listed in Tab. 2.9 are not suitable for my investigations. I conclude that a novel simulation framework needs to be created that fits my needs.
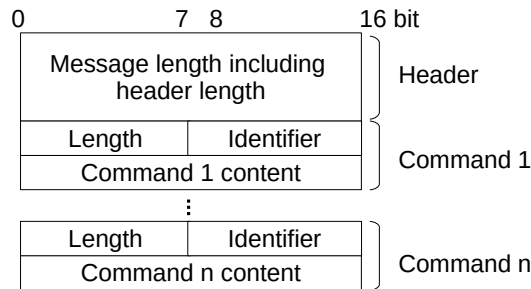

Figure 2.19.: Traffic Control Interface (TraCI) message. The message is communicated between the sockets of the mobile communication simulation and the socket of the mobility simulator using the transfer communication protocol (TCP). TraCI messages are composed of a header and several commands. Own graphics inspired by [273].

2.5. Uncertainty quantification

When evaluating simulation results, it is crucial to consider errors and uncertainties. Therefore, I look at uncertainty quantification methods in this section. Several procedures exist to quantify uncertainties for which a wide theoretical foundation is available, see [285, 286]. One method is the Monte Carlo method [285]. The distribution of the uncertain parameter is approximated through a discrete sampling. The sample values are propagated through the simulation model which results in an output distribution, see Fig. 2.20. The Monte Carlo method requires many samples to approximate the distributions accurately, which is why applying it becomes infeasible when simulation runs take long [285]. This is the case for mobile network simulations that usually take several hours. Therefore, I look at alternative uncertainty quantification methods that require less model evaluations.

In this section I look at surrogate-based uncertainty quantification with polynomial chaos expansions. I focus on two uncertainty quantification methods:

- Forward propagation: How uncertain is the model output due to uncertain parameters?

- Sensitivity analysis: How influential are the uncertain parameters on the model output?

The section is structured as follows: First, theory on polynomial chaos expansions is presented. In particular, it is explained how the coefficients of a polynomial chaos expansions can be determined. Then it is shown how uncertainty measures can be derived analytically from these coefficients.
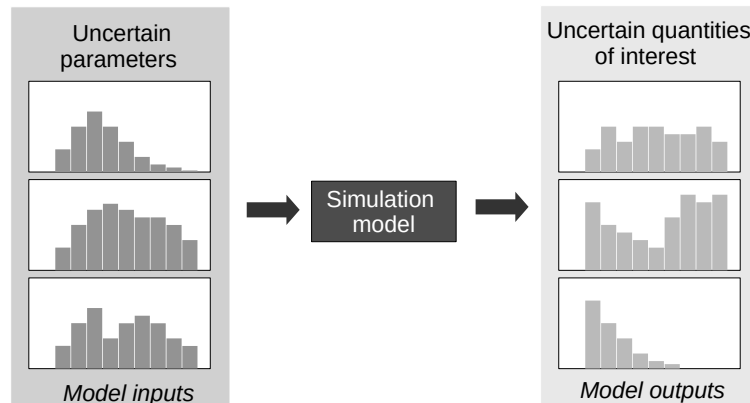

Figure 2.20.: One method for uncertainty quantification is the Monte Carlo Method [285]. It can be used for forward propagation that aims to quantify the uncertainty of an output distribution. The uncertain input distributions (left) are propagated through a simulation model. Therefore, one gets an output distribution for which statistical moments can be computed.

2.5.1. Polynomial chaos expansion

Polynomial chaos expansions, also known in literature [285] as spectral expansion, approximate an unknown model function f that depends on the variable t and the random variable $Z \in R^d$. Spatial, temporal, spatio-temporal, or any other deterministic dependencies are contained in t. The random variable Z corresponds to an uncertain parameter of the simulation model f. A polynomial chaos expansion $f_N(t, Z)$ approximates $f(t, Z)$ using polynomial basis functions [286, p.67]:

$$f(t, Z) \approx f_N(t, Z) = \sum_{i \leq N} \hat{f}_i(t)\phi_i(Z) \in P_N^d \qquad (2.2)$$

where P_N^d is the space of polynomials of Z of degree up to $N \geq 0$, $\hat{f}_i(t)$ are deterministic coefficients and ϕ_i are M orthogonal polynomials that form a basis for the random component of the solution [285, p.209].

For any $t \in T$, the mean of f, denoted as $\mu_f(t)$, can be approximated by [286, p.67]:

$$\mu_f(t) \approx E[f_N(t, Z)] = \int \left(\sum_{i \leq N} \hat{f}_i(t)\phi_i(z) \right) dF_Z(z) = \hat{f}_0(t) \qquad (2.3)$$

where $F_Z(z)$ is the distribution of the random variable Z: $F_Z(z) = P(Z \leq z)$, see [286, p.64]. Note that $\hat{f}_0(t)$ is the coefficient of the zero-order polynomial and has a similar function to the intercept in linear regression.

For any $t \in T$, the variance of f can be approximated by [286, p.67]:

$$var(f(t, Z)) \approx \sum_{0 < i \leq N} \hat{f}_i^2(t)E[\phi_i^2(Z)] = D_{PC} \qquad (2.4)$$

To determine the coefficients $\hat{f}(t)$, the Galerkin method, the collocation method, or the spectral method (projection) can be used. The Galerkin method is intrusive, which means that the simulation model needs to be adjusted, while the others are non-intrusive [285]. The spectral method requires fewer model equations than the regression-based collocation method as it only evaluates the model at particularly useful sample points. The errors of the mean and the variance convergence exponentially to zero over the number of expansion terms [287].

Each distribution type is assigned a specific polynomial type. This assignment is also known as the Wiener-Askey scheme, which extends the polynomial chaos expansions of Gaussian distributions to arbitrary distributions [287], see Tab. 2.10. The sample values are taken from quadrature tables and transformed to the respective parameter range. Therefore, the number of model evaluations equals the number of quadrature points.

	Uncertain parameter Z	Type of polynomial	Support
Continous	Gaussian	Hermite	$]-\infty, \infty[$
	Gamma	Laguerre	$[0, \infty[$
	Beta	Jacobi	$[a, b]$
	Uniform	Legendre	$[a, b]$
Discrete	Poisson	Charlier	$\{0, 1, 2, ...\}$
	Binomial	Kryawtchouk	$\{0, 1, 2, ..., N\}$
	Neg. binomial	Meixner	$\{0, 1, 2, ...\}$
	Hypergeometric	Hahn	$\{0, 1, 2, ..., N\}$

Table 2.10.: Distribution type of the uncertain parameter and referring type of polynomial [286, p.59].

Standard polynomial expansions require that the model function f and its derivatives are smooth [288, 289]. Otherwise, the system behavior is not captured accurately due to Gibbs phenomena, that is, overshoots occur at transition points [289]. To overcome this issue, variable transformation has been proposed: "The goal of the variable transformation is to create a new mapping which enables the response to be well approximated by a low-order polynomial chaos expansion." [289]. However, it is unclear how this can be implemented in practice if no information about f or its differentiability is available.

2.5.2. Forward propagation and global sensitivity analysis

Forward propagation aims to quantify the uncertainty of a quantity of interest due to uncertain parameters [285]. The uncertainty of the quantity of interest is quantified using the statistical moments [285, p.187]. For a given polynomial chaos expansion, forward propagation cuts down to evaluating Eqs. (2.3)-(2.4) that describe how mean and variance depend on the coefficients of a polynomial expansion, see Fig 2.21.

Sensitivity analysis aims to quantify the influence of each parameter on a quantity of interest [285]. Global sensitivity analysis evaluates the parameter influence over the entire parameter range rather than the effect of local perturbations. One technique is the variance-based approach that can be applied to linear and non-linear functions f [285]. The influence of each parameter on a quantity of interest f is quantified by so-called Sobol or sensitivity indices. Sudret et al. [290] showed that sensitivity indices can be determined analytically from a given polynomial chaos expansion.

The first-order sensitivity index S_i is defined as the ratio of the variance D_i that is caused by the parameter i only and the total variance D of the quantity of interest:

$$S_i = D_i/D \tag{2.5}$$

The total-effect index S_{T_i} is defined as ratio
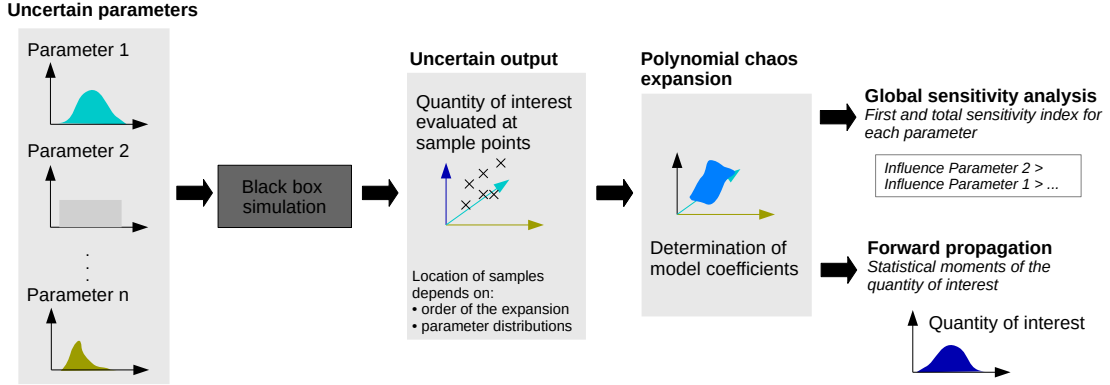
$$S_{T_i} = D_{T_i}/D \tag{2.6}$$

Figure 2.21.: Uncertainty quantification based on polynomial chaos expansions. Uncertain simulation parameters (left) are propagated through a black-box simulation model. The model evaluations are used to construct the polynomial expansion. With the coefficients, forward propagation and sensitivity analysis can be performed analytically.

where D_{T_i} is the variance caused by the parameter i and its interactions [291] with the other parameters. The sensitivity indices can be derived analytically from its coefficients for a given polynomial chaos expansion. For this purpose, it is necessary to introduce the multivariate polynomial chaos expansion $f_N(t, \mathbf{Z})$ [292]:

$$f_N(t, \mathbf{Z}) = \sum_{\alpha \subset N^d} \hat{f}_\alpha(t) \phi_\alpha(\mathbf{Z}) \tag{2.7}$$

Note that the above equation is similar to Eq. (2.2) with the exception that the random variable Z has been replaced by a set of random variables $\mathbf{Z}$ and the index i has been replaced the d-dimensional index $\alpha = \alpha_1 \alpha_2 ... \alpha_d \in N^d$ where d is the number of parameters. The first sensitivity index S_i can be represented as a function of the coefficients of the polynomial chaos expansions [292]:

$$S_i \approx \frac{\sum_{\alpha_i > 0} \hat{f}^2_{(0...\alpha_j...0)}}{\sum_{\alpha_i \subset N^d} \hat{f}^2_\alpha - \hat{f}^2_{(0...0)}} \tag{2.8}$$

Similarly, the total order indices can be represented as [292]:

$$S_{T_i} \approx \frac{\sum_{\alpha_i \subset N^d : \alpha_i > 0} \hat{f}^2_\alpha}{\sum_{\alpha_i \subset N^d} \hat{f}^2_\alpha - \hat{f}^2_{(0...0)}} \tag{2.9}$$

The index value evaluates the influence of a parameter. An index value of 1 means a high influence, and a value of 0 means a low influence. Standard libraries, such as the Python package Chaospy, provide implementations for the construction of the polynomial chaos expansion and the computation of the sensitivity indices.

2.6. Summary

In this chapter I presented and analyzed state-of-the-art approaches for crowd management. A definition for crowd management was provided, differentiating it from crowd control. Then metrics for evaluating pedestrian traffic were discussed. To redirect crowds efficiently, I looked at psychological theories about crowd behavior. I found that social identities affect crowd behavior and, therefore, should be considered when providing guidance information.

My review of crowd-sensing procedures showed that a variety of methods exist, including methods that utilize mobile communication. However, none of the approaches can measure flows or the spatial distribution of pedestrians using direct communication technology. I concluded that new methods are needed.

The review of algorithms that compute signals to guide crowds based on current environmental information showed that the proposed algorithms were either customized to scenarios and, therefore, not transferable or were tested under the assumption that people always follow instructions, which is wrong. I concluded that algorithms are needed that can be applied to arbitrary scenarios. They should be tested under realistic conditions, that is, not all people follow instructions.

In the second part I looked at modeling and simulation approaches. My review of crowd models showed that several validated models exist for modeling the movement behavior. However, no generally applicable route choice model is available for modeling route choice behavior. Therefore, I concluded that a new modeling approach is needed to model route choice in a crowd guidance system. In addition I found that although there are several approaches to modeling psychological behavior and communication, none of them have been used to redirect crowds in a crowd guidance system. Therefore, transferring these methods is necessary.

Then I looked at the modeling and simulation of direct communication in mobile networks according to four mobile standards that specify direct communication: The two IEEE standards 802.11p and 802.11bd that enable direct communication in WLAN ad hoc networks, and the LTE and 5G standards of the 3rd Generation Partnership Project intended for cellular communication with or without base station that allocates resources. I looked at different modeling approaches for the radio channel and gave an overview of standardized channel models released by the mobile communication community. These models can be employed for my investigations, and no further development is necessary.

Since a crowd management system is made up of various components, I next looked at approaches for coupling models and simulators. I presented two types of model couplings and discussed time-stepping and update procedures. I found that, due to the fact that a crowd guidance system based on direct communication technology is novel, a suitable update scheme needs to be found. Then I looked at the inter-processes communication,

that is, how simulators exchange data. I found that there are middleware simulation frameworks and standardized interfaces available, but there is no simulation framework available that allows one to simulate a crowd guidance system using direct communication technology. Therefore, I concluded that a simulation framework needs to be created that I can use for my investigations.

Finally, I presented a procedure for quantifying uncertainties in simulation results due to uncertain simulation parameters that I use in my investigations.

3. Software for crowd guidance systems based on direct communication

In this chapter I develop the scientific software *CrowNet* (<u>Crow</u>ds in <u>Net</u>works) that I use for the investigation of my research questions in Chapter 4. The chapter is structured as follows: First I give an introduction into the simulation framework *CrowNet*. I define its scope, requirements, quality measures and I present the software architecture. Since *CrowNet* has been developed in collaboration with other researchers (see Appendix A), I address which parts of the software were developed by me and which parts of the software I use but which were developed by other researchers. I then go into the development of the individual modules. Some modules are based on existing simulation frameworks known from the previous chapter. I show how these frameworks have been extended for their integration into *CrowNet*. Other modules provide novel models or procedures for which no software was available. I will present the respective concepts and their implementations. In this, I will focus on functionalities and features that are mandatory for the investigation of my research questions. A description of all available features of the *CrowNet* framework can be found in the software documentation. At the end of the chapter I give an overview of the functionalities that *CrowNet* provides for the investigation of crowd guidance systems which I will draw on in the following chapter.

> **Availability of the simulation software**
>
> The *CrowNet* software is free and open source (LGPL-2.1 license) and publicly available on github. Please find more information in Appendix A.

3.1. Introduction to CrowNet

3.1.1. Scope

The scope of *CrowNet* is to enable the simulation of a crowd guidance system as depicted in Fig. 3.1. The so-called controller contains the route recommendation algorithm, which provides route recommendations for redirecting the crowd at certain times. If the algorithm requires current traffic data as input, a mobile application based on direct communication technology is used as a sensor. The controller forwards the route recommendation to a second mobile application, which disseminates the information in the crowd. Importantly the recommendation for a specific time is the same for all crowd

members so that groups do not split up. The crowd reacts to a route recommendation, causing densities and flows to change. In the control loop, the crowd represents the dynamic system that should be influenced. Importantly, the behavior of the crowd is neither forced nor controlled. Rather, the crowd forms a system in which people can but do not have to follow instructions. This general approach allows to simulate both crowd control and crowd management applications (see Section 2.1.1). Therefore, systems — covered by the *CrowNet* software — differ fundamentally from systems where the crowd is considered as controllable fluid. Also, they differ from navigation systems that provide route recommendations for individuals not for crowds.

CrowNet enables the simulation of an overall crowd guidance system, an isolated subsystem and individual components. Importantly, multiscale systems can be simulated: A crowd moves much slower than information is disseminated over the mobile network.
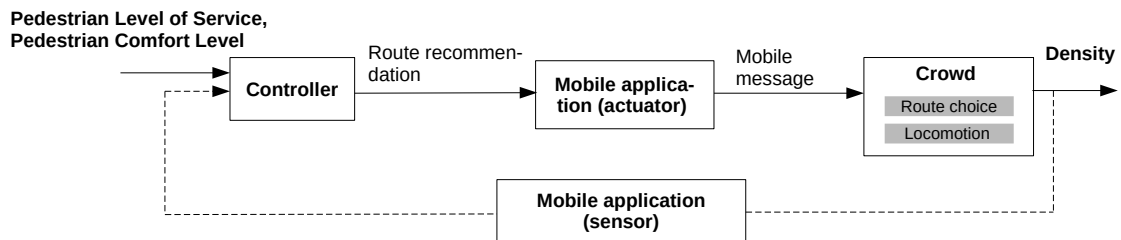

Figure 3.1.: Scope of *CrowNet*. With *CrowNet*, a crowd guidance system based on direct communication can be simulated. The crowd guidance system is represented as a control loop where the algorithm, implemented in the controller, generates route recommendations. The recommendation is disseminated through a mobile application based on direct communication (actuator). As input the algorithm uses density measurements provided by another mobile application (sensor).

3.1.2. Software architecture

CrowNet follows a component-based architecture, see Figure 3.2. The models come from simulators and model libraries that communicate over the Traffic Control Interface (TraCI). Component models can be easily exchanged or omitted to enable the investigation of isolated subsystems. Through its modular structure it is also possible to integrate additional simulators like, for example, the SUMO simulator (see [293]).

The five main modules of *CrowNet* fulfill the following purpose:

- The module *crownetutils* enables the execution of coupled simulations and provides pre- and post processing methods (Section 3.2).
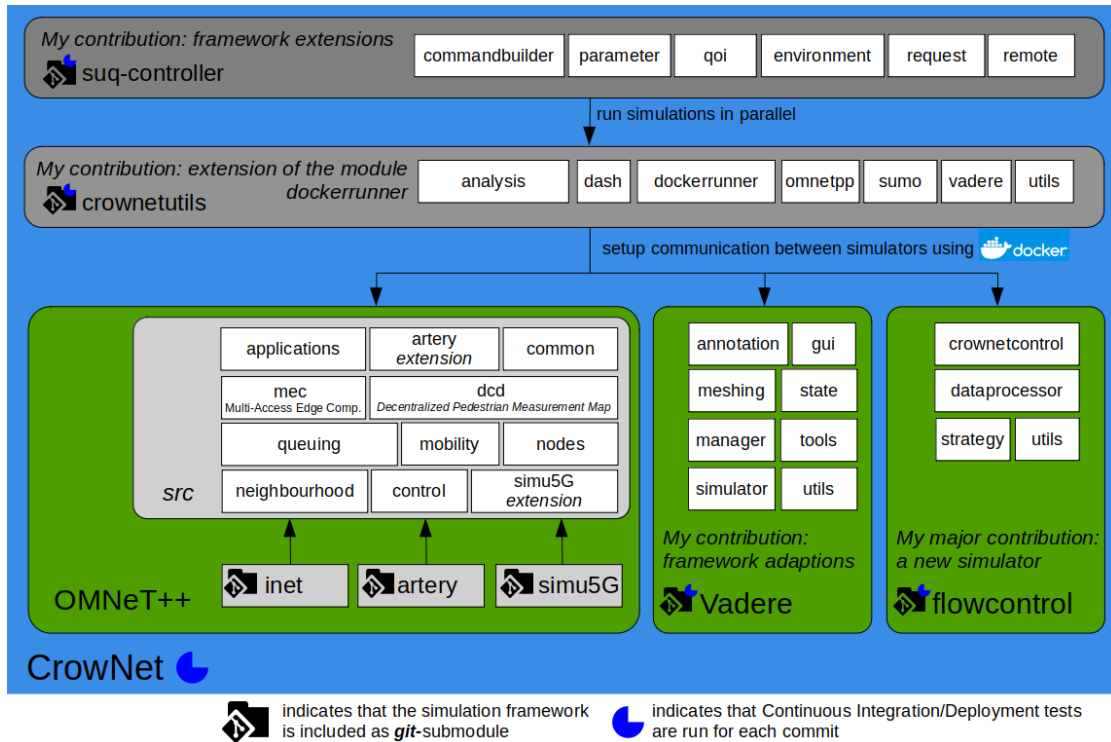
Figure 3.2.: Software architecture of *CrowNet* and my contributions. *CrowNet* is based on several simulators that are included as modules. The *OMNeT++* module (inet, artery, simu5G) provides mobile network models used to develop applications based on direct communication (src) such as sensoring crowds (dcd). The *Vadere* module provides validated models for crowd behavior. The *flowcontrol* modul is a novel Python simulator that provides route recommendation algorithms (my major contribution). The modules *crownetutils* and *SUQ-controller* enable simulations in parallel. I list my contributions for each *CrowNet* module separately in Appendix A.

- The module *SUQ-controller* enables parameter studies in which coupled simulations are executed in parallel. It extends the existing *SUQ-Controller* (Section 3.3).

- The module *OMNeT++* provides new mobile applications for the detection [4, 294] and redirection [295] of crowds (Section 3.4).

- The module *Vadere* provides models for crowd behavior. It is based on the Vadere simulator [209]. For its integration into *CrowNet*, numerous changes were made (Section 3.5).

- The module *flowcontrol* provides route recommendation algorithms. *flowcontrol* is a novel Python framework that I have developed in this thesis (Section 3.6).

3.1.3. Requirements, quality measures and continuous testing

Requirements

The *CrowNet* software is developed based on principles from software engineering. First, functional and non-functional requirements are defined, see Tab. 3.1-3.2. The requirements define usability, testing, availability, model validity, and execution functionalities. For each requirement, a design decisions is derived that defines how the respective requirement is implemented. The design decisions listed in Tab. 3.1-3.2 are implemented in the five main modules of the *CrowNet* software.

Functional requirement	Design decision
It must be possible to build a crowd guidance system according to Fig. 3.1 using existing component models.	Provide component models and manage their interaction.
It must be possible to exchange or extend models.	Re-use existing simulators that provide models; do not reinvent the wheel.
It must be possible to integrate models in novel frameworks to improve the sustainability.	Use the Traffic Control Interface [273] for data exchange between simulators.
It must be possible to run the simulation using a single execution script.	Provide an execution script that collects all needed executables, libraries and dependencies to run a coupled simulation.
It must be possible to conduct parameter studies and run simulations in parallel.	Extend the Python framework SUQ-controller (https://gitlab.lrz.de/vadere/suq-controller).
It must be possible to run a coupled simulation in debug mode.	Use git sub-modules that provide the source code of simulators and model libraries.
It must be possible to export and visualize results.	Choose simulators with visualization and export functionalities.
It must be ensured that the change of a component only affects component-related properties and results.	Continuously test the software using unit and fingerprint tests.
The simulation framework must be accessible to everyone.	Make CrowNet publicly available on github: https://github.com/roVer-HM/crownet.

Table 3.1.: Requirements and corresponding design decisions for functional requirements. Please see Tab. 3.2 for the non-functional requirements.

Non-functional requirement	Design decision
The installation effort should be low.	Employ continuous deployment. Provide software components as Docker images on github: https://github.com/orgs/roVer-HM/packages.
The provided component models should be validated.	Select simulators and model libraries carefully. Use libraries with validated models.
The provided component models should be maintained.	Select simulators and libraries that are actively maintained by checking the number of associated simulators (and forks).

Table 3.2.: Requirements and corresponding design decisions for non-functional requirements. Please see Tab. 3.1 for the functional requirements.

Quality measures

Quality measures are introduced to ensure the quality of the software, and thus, that other researchers can understand, check and use the source. The quality measures should also help to increase the sustainability of a software in the sense of Open Science (see Appendix A).

I adhere to the following principles of software development:

- Employ a test-driven approach. I create the test method first and then implement the actual method.

- Use a Version control system. I use git 2.25.1, which makes it possible to track the development process and jump back to older software states. This also enables researchers to reproduce study results.

- Follow the principle of open science and make the software publicly accessible. This increases confidence in the software and avoids duplicate implementations. The code is publicly available on github: https://github.com/roVer-HM/crownet.

- Provide releases and provide the current state of the software. I provide a stable version (release) and a main branch (current development) on github. The main branch provides new features that can be tested by external users before they are released.

- Follow the guidelines for clean code [296]. I defined traffic engineers, crowd managers and researchers from the field of pedestrian dynamics as target groups. I use comments to explain the functionality of classes and methods. This is mandatory as terms have different meanings between the target groups.

- Provide documentation and tutorials. I document the code in a way that both developers and users can understand and use the code. I also provide tutorials and quick start examples.

- Report document bugs and open tasks. I use issues boards to point out bugs and open tasks and to fix them. There is an issue board for developers and one for communication with external users that is publicly accessible: https://github.com/roVer-HM/crownet/issues

- Apply the practice of continuous integration and continuous deployment. A push to the remote repository triggers a continuous integration pipeline, see Fig. 3.3.

Testing and continuous integration

The continuous integration pipeline consists of four stages: build, test, uq-test and deploy, see Fig. 3.3. The build stage (1) tests whether the *OMNeT++* module can be built (job: build-for-sim). It also tests whether the modules *SUQ-controller* and *crownetutils* can be installed in a virtual Python environment. The Python environment serves for the tests of the following stages.
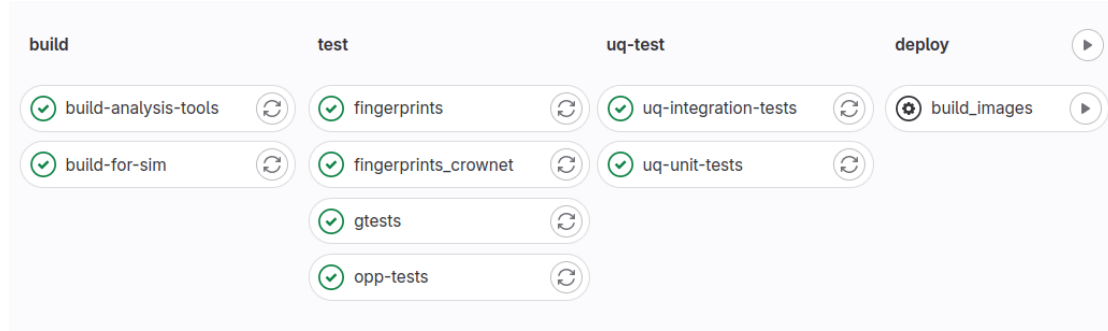

Figure 3.3.: Continuous Integration pipeline of the *CrowNet* simulation framework. The snapshot was taken from the pipeline of the private gitlab repository. In the build stage, *CrowNet* gets compiled (`build-for-sim`) and a virtual Python environment is generated (`build-analysis-tools`) that is used to run *CrowNet* tests. The next stages contain unit tests and fingerprint tests. The manually triggered deployment stage (right) generates Docker images.

In the test stage (2), unit tests, regression tests, and integration tests are carried out. For this purpose existing testing frameworks are employed: in `opp-tests` the `opp_test` tool of the *OMNeT++* simulator [254] is used to run integration tests. The `opp_test` tool tests mobile network applications. "Message creation and sending as well as result recording and other *OMNeT++* utilities are available just like during simulation execution" [254, p.232]. Plain C++ code is tested in a separate testing framework as recommended

in [254], using the `GoogleTest` and the `GoogleMock` unit testing frameworks. Note that `gtests` stands for 'google tests' in Fig. 3.3.

Regression testing is conducted to ensure that changes do not accidentally break existing functionalities: Fingerprint tests compare a hash value computed from the simulation output with a reference fingerprint. Only if the hash values are identical, the test is successful. Fingerprint tests are available for the *OMNeT++* module (`fingerprints` in Fig. 3.3) and for the coupled simulation that includes the update scheme and the data exchange between simulators (`fingerprints_crownet` in Fig. 3.3).

The stage `uq-test` tests whether parameter studies can be carried out in parallel. In `uq-unit-tests` the generation of samples is tested. Parameter studies including post-processing methods are carried out in `uq-integration-tests`. This includes my simulation studies of Chapter 4.

Docker images are used to setup and run simulations quickly and easily. The images can be deployed in the `deploy` stage (4) by a manual trigger. The images are generated based on the current commit. The tag of the Docker image is named after the commit for version control purposes. This helps researchers and developers to find a specific software state for reproducing results.

In addition to the continuous integration pipeline shown in Figure 3.3, testing is performed for each *CrowNet* module. Changes in a specific module trigger the continuous integration pipeline of the corresponding module. Only if the module-specific pipeline has been passed successful, the *CrowNet* pipeline (Fig. 3.3) is run in a second step. This two-step procedure simplifies and accelerates the development of individual components, as implementation errors are already detected in the module-specific tests.

For the *Vadere* module, the existing continuous integration pipeline [155] is used. Novel pipelines are setup for the modules *crownetutils*, *SUQ-controller*, and *flowcontrol*. As each module is integrated as a `git sub-module` in *CrowNet*, the corresponding pipelines can be triggered directly through the *CrowNet* repository.

3.2. Module crownetutils: model and simulator coupling

The *crownetutils* module has the following functionalities:

- Configuration of the model structure of a crowd guidance system.

- Execution of the coupled simulation.

- Provision of pre- and postprocessing functionalities for coupled simulations.

3.2.1. Model composition and simulator coupling

The functionalities for model composition and simulator coupling are implemented in the package `dockerrunner` of the *crownetutils* module. The core class of the package is the `BaseSimulationRunner` that has a `run()` method for the execution of a coupled simulation. The `run()` method is called by the *CrowNet* executable: the `run_script`. Each simulation has its own `run_script`, which is stored in the corresponding simulation directory, see Fig. 3.4.
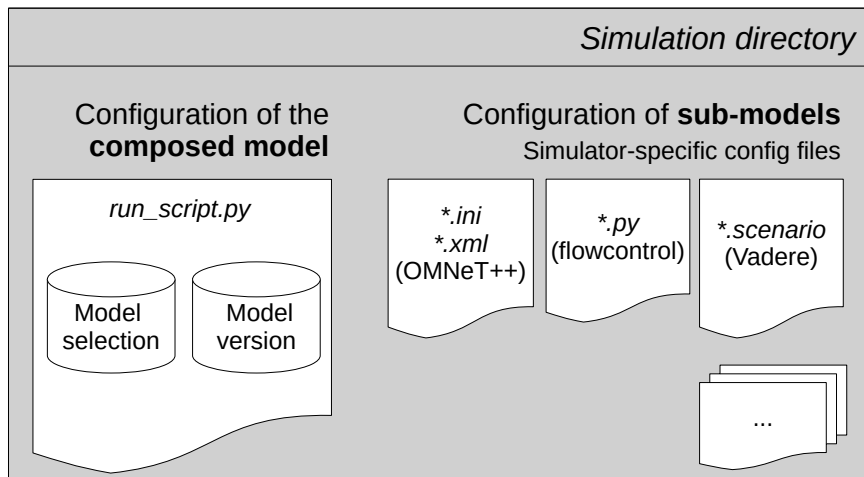

Figure 3.4.: Configuration of a simulation. The model structure of the crowd guidance system is specified in the `run_script.py`. The crowd simulation (*Vadere*) is configured in the *.scenario file. The *.ini and *.xml files are config files for the mobile network simulation (*OMNeT++*). The route recommendation algorithm (*flowcontrol*) is specified in a Python file (*.py).

The structure of the crowd guidance system is defined in the `run_script.py`, see Listing 3.1. Parameter values are defined in the simulator-specific config files. Parameters of the crowd model are defined in the Vadere-specific `*.scenario file`. Parameters of the communication simulator *OMNeT++* are defined in the configuration file `omnetpp.ini`. The route recommendation algorithm, implemented the simulator *flowcontrol*, is specified in `control.py`. Which simulator-specific files need to be included depends on the structure of the crowd guidance system.

```python
from crownetutils.dockerrunner.simulationrunner import
    BaseSimulationRunner

if __name__ == "__main__":

    ...
    settings = [
    # configuration of the composed model
        'vadere-control', # start Vadere and flowcontrol only
        '--with-control', # specify the controller
        'control.py', # file that contains guiding strategy
        '--ctrl.controller-type',
        'AvoidShortAlgorithm', # select guiding strategy from control.py
        '--scenario-file',
        'vadere/scenarios/three_corridors.scenario', # scenario
        '--vadere-tag', # version of the docker images
        'latest',
        '--control-tag',
        'latest',
    # simulation configuration
        '--write-container-log',
        '--experiment-label',
        'myfirstcrownetexperiment',
        '--run-name',
        'sample_0', # unique name for distinguishing parallel runs
    ]

    runner = BaseSimulationRunner(working_dir, settings)
    runner.run() # start the simulation
```

Listing 3.1: `run_script.py`: executable and configuration file. Arguments specify the structure of the model and how the simulation is executed. Simulator-specific tags (–*-tag) ensure that research results can be reproduced. The tags correspond to a certain commit of the *CrowNet* repository or use its current ('latest') version.

In the *crownetutils* module, several simulator couplings have been implemented. With this, the modeler can freely decide which parts of the crowd guidance system should be simulated. The flexible model structure brings a performance advantage: Instead of simulating the entire system, isolated subsystems can be simulated which saves computational effort. Also it keeps the model as simple as possible.

An overview of possible simulator combinations can be found in Fig. 3.5. The simulators exchange data using a socket communication. The crowd simulator *Vadere* is always the server. The mobile communication simulator *OMNeT++* is always the client. Depending on the combination of simulators, *flowcontrol* acts either as a server or client. The required simulator coupling can be selected through sub-commands that are passed over the command line interface.

vadere-opp

OMNeT++	→ ←	vadere

client   *server*

vadere-control

flowcontrol	→ ←	vadere

client   *server*

vadere-opp-control

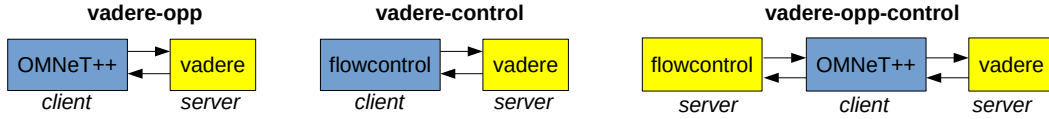| flowcontrol	→ ←	OMNeT++	→ ←	vadere

server   *client*   *server*

Figure 3.5.: Simulator couplings. The crowd dynamics simulator *Vadere* [209], the mobile networks simulator *OMNeT++* [265] and the novel route recommendation simulator *flowcontrol* share data through socket communication.

The execution of a coupled simulation is managed in the `dockerrunner` module. The `BaseSimulationRunner` class starts the simulators in the correct order: It creates a Docker network in which the individual simulators communicate through a socket connection, see Fig. 3.6. Using a Docker network ensures that the simulator communication processes is separated from the host network and does not influence processes of the operating system. To setup the socket connection the `BaseSimulationRunner` first starts the server(s) that wait for the client to establish a connection.

Each simulators is started inside the respective Docker container, see again Fig. 3.6. The `DockerRunner` of each server ensures that the simulation inside the container is ready before the next container is created. Therefore, delays during the starting process have no effect. The well-tested Python Docker API is used for the implementation.

When coupled simulations are run in parallel, it must be ensured that the correct Docker containers connect with each other. To achieve this, the name of the Docker container is utilized as a host name.
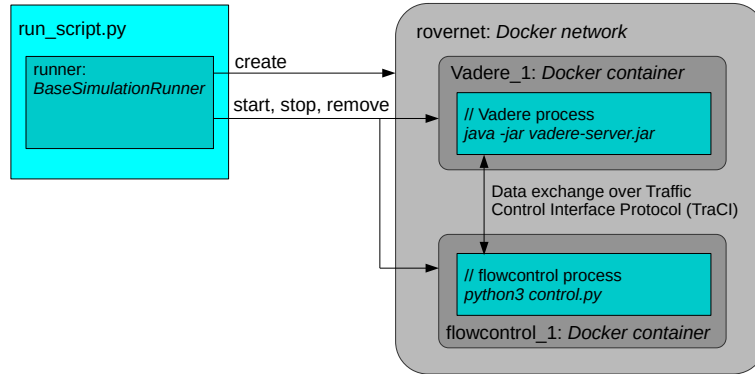

Figure 3.6.: Example for simulator communication in the Docker network. The `run_script.py` creates the Docker networks in which the simulators are executed in Docker containers that exchange data over a socket connection.

The `BaseSimulationRunner` uses the `SimulationDispatcher` protocol to start and stop the required containers, see Fig. 3.7. The method of the `SimulationDispatcher` protocol is selected from a sub-command provided over the command line interface. For example,

if the string `vadere-control` is passed, the method `run_simulation_vadere_ctl()` is selected. The advantage of decoupling dispatcher and runner is that further simulator couplings can be easily implemented, while at the same time, all classes implementing the respective methods can be called through the `SimulationDispatcher` protocol. To start a simulation the main method of the `run_script.py` calls the run method of `SimulationRunner`. Therefore, the `run_script.py` has a dual function: Configuring the model structure and executing the simulation.

Once a socket connection has been established, the simulation states are exchanged via the Traffic Control Interface (TraCI) [273]. In the *Vadere* simulator the Traffic Control Interface was implemented in [297]. *Vadere* has the role of the server and waits for requests that are provided via the network, see Fig. 3.8. *OMNeT++* and *flowcontrol* use subscriptions to request the simulation state from the other simulator.

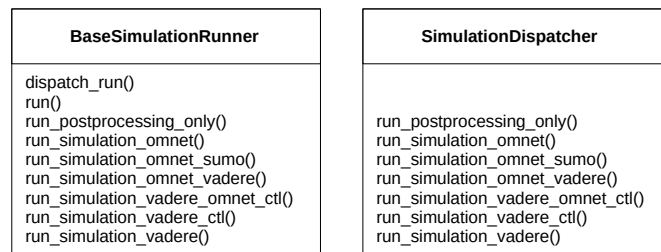| **BaseSimulationRunner**	**SimulationDispatcher**
dispatch_run() run() run_postprocessing_only() run_simulation_omnet() run_simulation_omnet_sumo() run_simulation_omnet_vadere() run_simulation_vadere_omnet_ctl() run_simulation_vadere_ctl() run_simulation_vadere()	run_postprocessing_only() run_simulation_omnet() run_simulation_omnet_sumo() run_simulation_omnet_vadere() run_simulation_vadere_omnet_ctl() run_simulation_vadere_ctl() run_simulation_vadere()

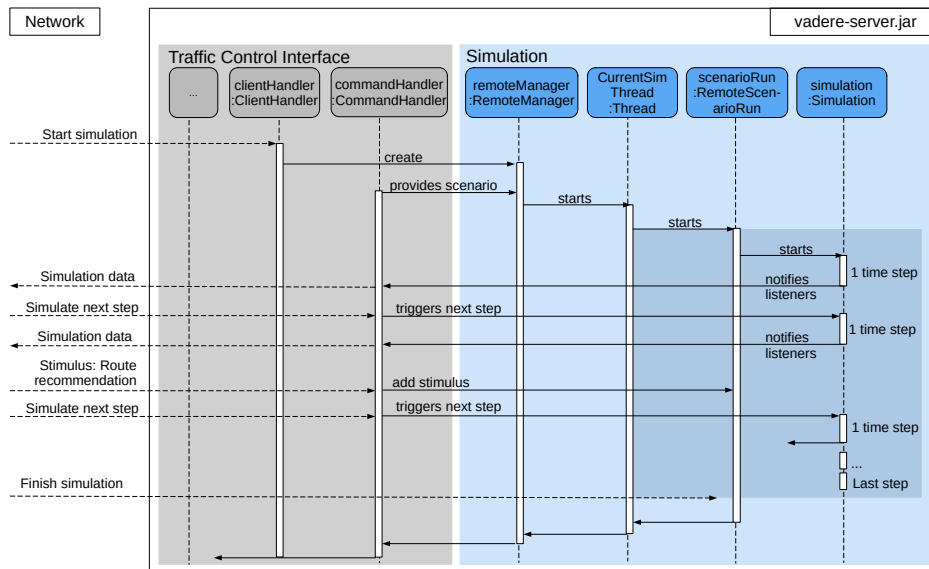Figure 3.7.: `SimulationRunner` and `SimulationDispatcher`: simplified class diagrams.

Figure 3.8.: Traffic Control Interface and simulation loop of *Vadere*. The interface was implemented in [297] to enable a communication with the mobile network simulator *OMNeT++*.

3.2.2. Explicit update scheme with position interpolation

An explicit update scheme is implemented to update the state of the component models, see Fig. 3.9. The global time step size corresponds to the time step size of the crowd simulator *Vadere*. *Vadere* provides mobility data for the simulator *OMNeT++*.

Figure 3.9.: Explicit update scheme with position interpolation. The route recommendation algorithm, the crowd model and the mobile network model exchange data at certain times. The route recommendation algorithm provides route recommendations every m time steps with $m \geq 1$ (left). The position data in the mobile network simulation is interpolated linearly (right).

In *OMNeT++*, the position of agents is interpolated using the current and the future state. The route recommendation algorithm (*flowcontrol* simulator) is evaluated every m time steps where $m \geq 1$.

The interpolation of the agent position is exact if the node velocity of an agent (i) is constant within one time step. The speed of an agent (i) is estimated from the current and future node position:

$$v^i = \begin{cases} \frac{X^i_{n+1} - X^i_n}{\Delta t} & n > 1 \\ 0 & n = 1 \end{cases} \tag{3.1}$$

where $n \in \mathbb{N}^+$ denotes the time step, X^i denotes the position of agent (i), and t_e is the time in between the event and the time t_n. The position of agent (i) at a specific point in time (e) is:

$$X^i_e = X^i_n + v^i t_e \tag{3.2}$$

Importantly, there is no physical coupling between the mobile network simulation and the crowd simulation. Only the current agent positions play a role in the mobile networks simulation. Therefore, one can assume that the update scheme does not lead to any stability problems. I check this in my investigations.

3.2.3. Pre- and Postprocessing methods

The *crownetutils* module provides a feature with which pre- and post-processing methods can be implemented. Is is based on annotations, see `process_as` in Listing 3.2. The pre- and post-processing methods are selected using the command line parameter `--qoi` of the `run_script` . Each pre- and post-processing method is assigned a priority value to specify the order of execution: see the properties `type` and `prio` in Listing 3.2. The higher the `prio`, the sooner the method is executed. The package `analysis` provides implementations of standard metrics for the evaluation of the mobile communication simulation.

```python
1  from crownetutils.dockerrunner.simulationrunner import
       BaseSimulationRunner, process_as
2
3  class SimulationRun(BaseSimulationRunner):
4
5      @process_as({"prio": 1, "type": "pre"}) # before sim start
6      def prepare_1(self):
7          ...
8
9      @process_as({"prio": 1, "type": "post"}) # after sim finished
10     def result_1(self):
11         ...
12         dataframe.to_csv(result_1.csv)
13
14     @process_as({"prio": 2, "type": "post"}) # after sim finished
15     def result_2(self):
16         ...
17         dataframe.to_csv(result_2.csv)
```

Listing 3.2: Adding pre- and postprocessing to a *CrowNet* simulation using annotations.

3.3. Module SUQ-Controller: extensions for coupled simulations

The *SUQ-controller* module builds on the Surrogate and Uncertainty Quantification controller (SUQ-controller) that is an open source Python framework for executing parameter studies in parallel, see Appendix A. The *SUQ-controller* was intended for conduction parameter studies with *Vadere*. It created a copy of *Vadere*'s config file (*.scenario) for each sample and overwrote the parameter values.

In *CrowNet*, however, the simulation is specified via several configuration files. In addition, simulators are coupled. Therefore, the *SUQ-controller* needs to be extended for conducting parameter studies with coupled simulations.

3.3.1. Enabling coupled simulations through new process calls

The *SUQ-controller* is extended to execute coupled simulation by introducing the new module `CommandBuilder`, see Fig. 3.10. The core of the module is the class `Command`.
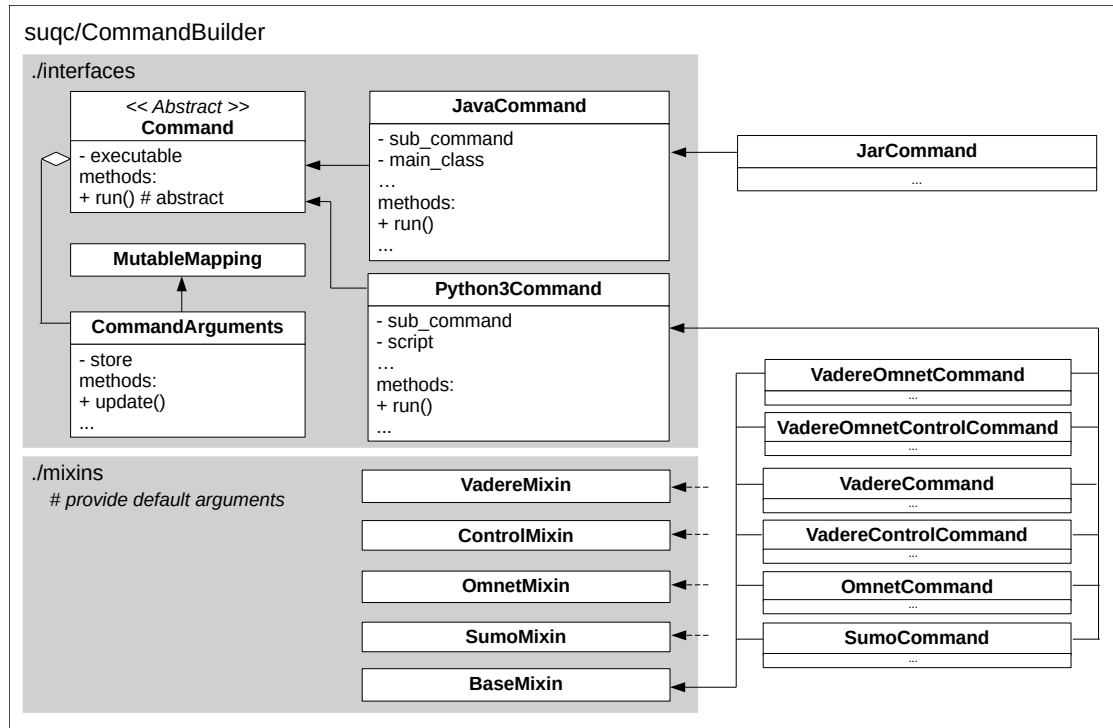
Figure 3.10.: Extension of the SUQ-Controller: The module `CommandBuilder` provides commands to conduct simulation studies with coupled simulations.

The `Command` class defines how an executable is started. It has two child classes: `JavaCommand` and `Python3Command`. The former relates to Java executables, such as `vadere-console.jar`. The latter specifies Python executables, such as `run_script.py`. Each child class implements a corresponding `run()` method. The `run()` method of the `JavaCommand` child class is used to start the *Vadere* application via the command `java -jar vadere-console.jar`.

The `run()` method of the `Python3Command` class calls the `runscript.py`. `Python3Command` has numerous child classes, see Tab. 3.3. With the different child classes, parameter studies can be conducted with individual components, isolated subsystems or the entire crowd guidance system.

The child classes inherit from the `mixins` package, see again Fig. 3.10. The `BaseMixin` class defines general simulation settings such as the log level. The simulator-specific `mixins` enable the configuration of simulator releases and simulator-specific settings.

Command	Parameter study with
VadereOmnetControlCommand	... a crowd guidance system according to Fig. 3.1.
VadereOmnetCommand	... a crowd guidance system without dynamic route recommendation provision.
VadereControlCommand	... a crowd guidance system without mobile communication.
OmnetCommand, VadereCommand	... an individual component of a crowd guidance system.

Table 3.3.: Child classes of the `Python3Command`. They serve as wrapper commands.

3.3.2. Executing parameter combinations in parallel

The simulator-specific config files (see again Fig. 3.4) contain the model parameters. To conduct a parameter study, the parameter values in these files need to be changed. The *SUQ-controller* reads the respective config files and parses them into a dictionary. To parse the *.ini file of the *OMNeT++* simulator, the Python package *omnetinireader* has been developed, see Appendix A. It is open source and is included as external library in the *SUQ-controller*.

Parameters of the *flowcontrol* simulator are changed over the `--controller-type` argument that is passed to `run_script.py`. The *SUQ-Controller* requires that parameter values are stored as a list of dictionaries. Each list element corresponds to one sample. The parameter values are specified for each simulator separately, see Listing 3.3.

```
sample = {'omnet': { "*.misc[0].app[0].repeatInterval": "1.0s",
                     "**wlan[*].radio.transmitter.power": "2.0mW" },
          'vadere': {"sources.[id==1].spawner.distribution.updateFrequency": 2.0}}
```
Listing 3.3: Sample definition (Python dictionary) in a coupled simulation.

Since the *Vadere* and *OMNeT++* simulators contain stochastic processes a `SeedManager` is implemented, see Fig. 3.11. It generates reproducible seed values that are fed into *Vadere* and *OMNeT++*.

The execution of a parameter study with coupled simulators is shown in Fig 3.12. The executable is defined via a `Command` object. The parameter combinations are passed as a list of dictionaries to the `Command` object. The *SUQ-controller* then creates simulation directories and manages the execution of simulations in sub-processes. To reduce the overall runtime, the processes are started in parallel. To ensure that simulator instances are correctly assigned, the Docker containers receive a post-fix with the corresponding sample number: The container `flowcontrol_Sample_00` only communicates with the containers `omnet_Sample_00` and `vadere_Sample_00` but not with `vadere_Sample_01` or `omnet_Sample_02`.
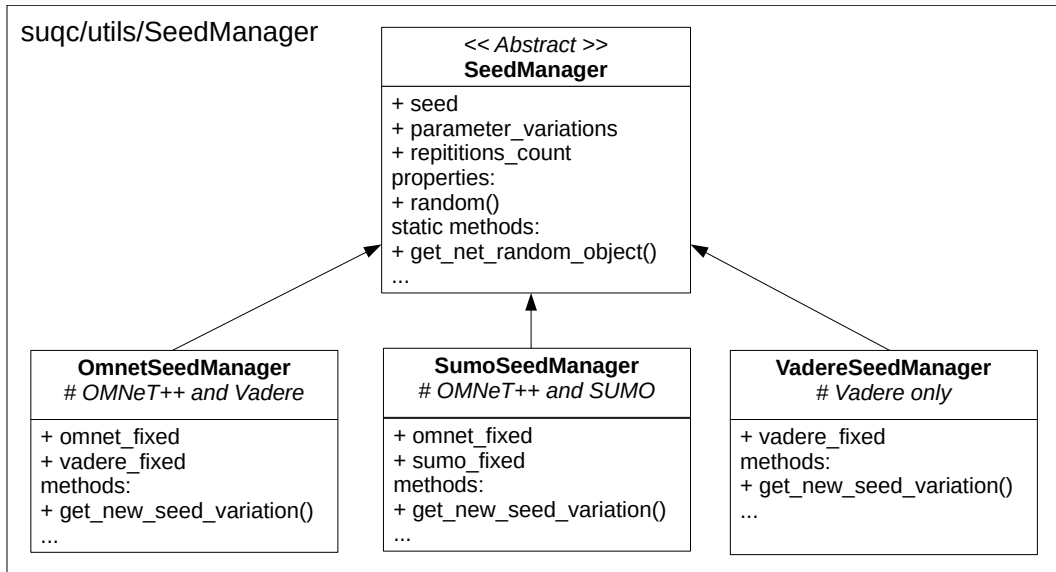
Figure 3.11.: `SeedManager` of the *SUQ-controller* module. Seeds are generated and fed into the *OMNeT++* simulator and the *Vadere* simulator. This ensures that the repetition of stochastic simulations leads to the same results.
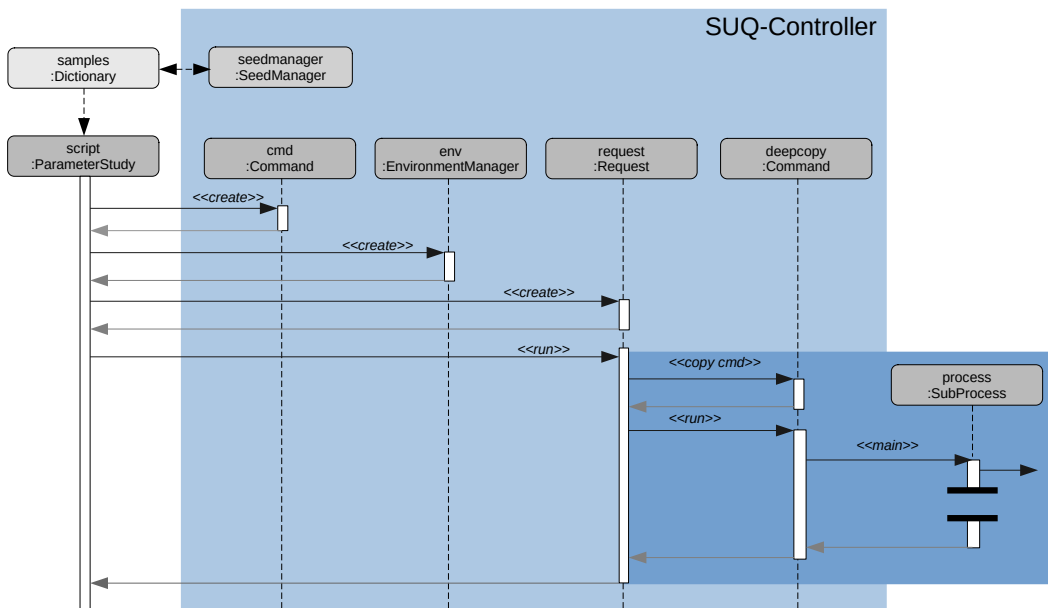

Figure 3.12.: Conducting a parameter study using the *SUQ-controller* module. The `SeedManager` generates seeds for the samples. Before the simulation is started in a sub-process, several objects are generated that specify or manage the simulation (`cmd`, `env`, `request`).

3.4. Module OMNeT++: provider for direct communication

The *OMNeT++* module provides mobile network models and executes the mobile network simulation. It contains implementations of crowd sensing procedures and mobile applications that disseminate information based on direct communication. I use the module, but I did not contribute to it.

3.4.1. Employing OMNeT++ and its ecosystem

In the *OMNeT++* module, several frameworks are integrated to simulate direct communication. The frameworks have been developed externally and have inter-dependencies, see Fig. 3.13. Some of them enable the simulation of WLAN-based communication and some others enable cellular communication. All frameworks are based on the *OMNeT++* simulation engine for the execution of their models. The *Vadere* crowd simulator was integrated in the *OMNeT++* ecosystem in 2019 [297]. The aim was to provide realistic pedestrian mobility data for the mobile communication simulation.
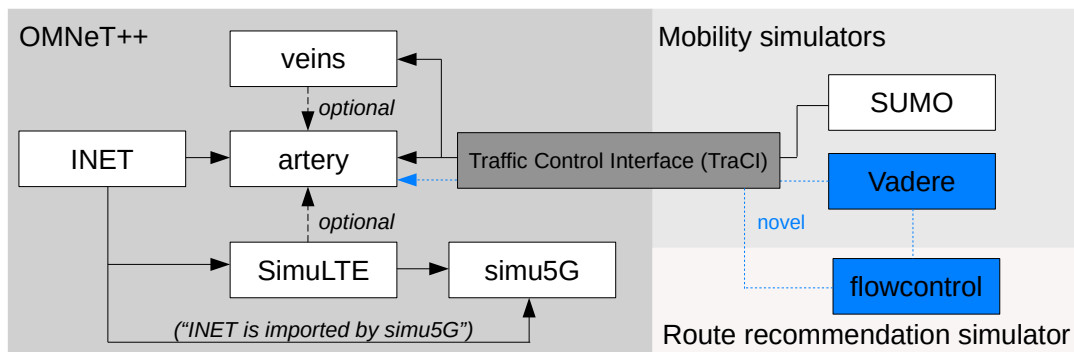
Figure 3.13.: Package dependency of the *OMNeT++* module. The *OMNeT++* simulation engine is related to several simulators and model libraries. Only simulators that are integrated in the *OMNeT++* module are depicted (*inet*, *artery*, *SimuLTE*, *simu5G*, and *Veins*). The pedestrian dynamics *Vadere* was integrated in [297]. The novel simulator *flowcontrol* is integrated as part of this thesis. Own graphics inspired by [254, p.369].

The *OMNeT++* module is named after the simulation engine *OMNeT++* [265]. The engine is event-based and intended for arbitrary distributed systems. *OMNeT++* is licensed under the Academic Public Licence. The core of *OMNeT++* is the C++ simulation kernel that handles the execution of event-based simulations. This includes the processing of input files, the scheduling of events, collecting results, and handling the optional online visualization. *OMNeT++* has a generic component architecture. Model components are termed 'modules' that "communicate via message passing, either directly or

via predefined connections. Messages may represent events, packets, commands, jobs, or other entities depending on the model domain" [265]. So-called simple modules use the simulation class library and are implemented in C++. Modules can represent different objects, such as agents, sinks generating network traffic, or network interface cards. So-called compound modules are composed of at least two simple modules. They represent more complex systems such as hosts or routers. For the setup of the network topology so-called Network Topology Description (NED) files are used. Model parameters and simulation kernel settings are defined in the *.ini file.

The *INET* framework [298] serves as model library for the simulation of direct communication according to the 802.11p standard. *INET* is an established library that contains standard communication protocols for network simulations. It is licensed with the LGPL license and is used in research, academia, and industry. *INET* has started out as "a model suite for the Internet protocol stack (e.g., IPv4, TCP, and UDP), and has gradually expanded to other areas, in particular: Ethernet, IEEE 802.11, QoS mechanisms, MPLS, DiffServ, Internet routing protocols, node mobility, MANET routing protocols, further wireless MAC protocols, IPv6, and SCTP, just to name a few. Recent advancements include physical layer and physical medium modeling, advanced visualization features, and improved network emulation capabilities." [298]. *INET* provides transceiver models including antenna, models for the transmission medium and obstacle and terrain models. Each model can be individually configured or extended. The simulation frameworks *simuLTE, simu5g* and *artery* — all used by the OMNeT++ module — build on *INET*. See again Fig. 3.13.

The *simuLTE* [299] framework and the *simu5G* framework [267] enable the simulation of direct communication in cellular networks. *simuLTE* is a system-level simulator that enables the simulation of 4G LTE and Long Term Evolution Advanced (LTE-A) networks. It also serves as a model library for node models and base stations. *simuLTE* provides the full LTE protocol stack that is composed of the physical layer, the medium access layer, the Radio Link Control (RLC) and the Packet Data Convergence Protocol (PDCP). The provided receiver models consider simultaneous transmissions in the computation of the Signal-to-Interference-plus-Noise-Ratio (SINR) which allows the analysis of interference. *simuLTE* enables infrastructure-mode communications and sidelink communications in mode 3 (controlled mode) where the base station allocates resources. *simu5G* extends the *simuLTE* framework by enabling the simulation of 5G networks. The source code of *simuLTE* is fully provided in the *simu5G* framework. Like *simuLTE*, *simu5g* only supports the controlled mode (mode 3), that is, resources are allocated by a base station while data is exchanged over sidelinks.

The *veins* [300] framework is an open source library that provides models for vehicular communication according to the 802.11p standard. It provides models for the IEEE 802.11p Medium Access Control (MAC) layer, the IEEE 1609.4 Multi-Channel, and the IEEE 802.11p physical layer. For the upper layers, it uses the WAVE architecture that is used in the United States. Veins does not provide any models for modeling the mobility behavior of vehicles, but accesses position data from the SUMO simulator via the Traffic Control Interface (TraCI). The coupling via the Traffic Control Interface is bi-directional. With this, one can simulate traffic rerouting [254, p.218].

Like *veins* the *artery* [301, 302] framework enables vehicular communication. It was originally intended as a module of the *veins* framework. Meanwhile *artery* has become a separate framework licensed under the GPLv2 license. *artery* builds on models provided by the *INET* framework. It extends the 802.11p physical and access layers with ad-hoc mode functionalities. The scope of *artery* is the simulation of application scenarios in European countries. Hence, the radio channel bandwidth and the carrier frequency are set to frequencies of 10 MHz and 5.9 GHz. Like *veins*, *artery* uses the Traffic Control Interface for receiving mobility data. *artery's* implementation of the Traffic Control interface is used for the data exchange with the simulators *flowcontrol* and *Vadere*.

3.4.2. Pedestrian density map application for crowd sensing

The package `applications` provides implementations of mobile applications. One of them is the 'decentralized pedestrian density map application' that was first proposed in [294] and further developed in [4]. The application aims to measure the spatial and temporal distribution of the density of people based on a cell-grid, see Fig. 3.14.
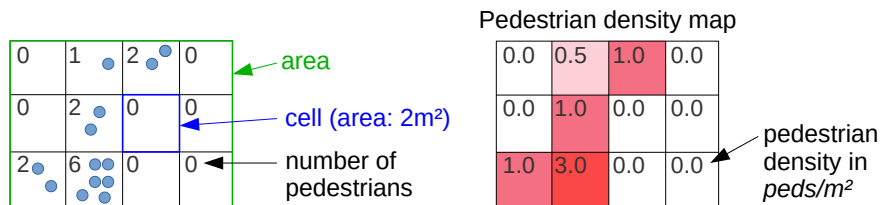
Figure 3.14.: Decentralized pedestrian density map. The topography is discretized with a cell-grid. The density is estimated from the number of received beacons per cell. The beacons are assigned to a specific cell via a position date.

The idea of the density map application is to count people by counting 4G/5G sidelink beacons [294]. Devices regularly send beacons with position data via the so-called beacon service, see Fig. 3.15. Each device counts the number of received beacons for each cell. Based on the count and the cell size, the density can be derived for each cell. Since the range of a beacon is limited, the position data is only locally available. To receive density information from distant cells, density data is exchanged between devices over

a second service: the density map service. The density maps are aggregated to improve the accuracy of densities, see Fig. 3.16.

Both beacons and density map application packets communicate using 4G/5G sidelink communication [4]. Because each device counts itself, the pedestrian counting is both local and decentralized. Data flow and storage times are, therefore, not covered by data protection [303].

The density map application has been tested in simulation studies for LTE in the controlled mode (mode 3) [4]. In the study the authors used a base station with 25 resource blocks and a carrier frequency of 2.6 GHz in combination with a heuristic path loss model [304]. Two filter algorithms were proposed for the aggregation of the count data [4]:

- Youngest measurement first. For each cell, sort the estimated values according to their age. Use the most recent estimate.

- Youngest measurement first plus distance. For each cell, normalize age and distance (distance between the cell and the cell of the device that estimated the number of persons) to values between 0...1. Calculate a weighted mean from the normalized values (suitable weights: 0.9 for the normalized age and 0.1 for the normalized distance, see [4]). Use the estimated value that is assigned to the smallest weighted mean.

If no beacon has been received for a cell, it is assumed that the cell is not occupied.
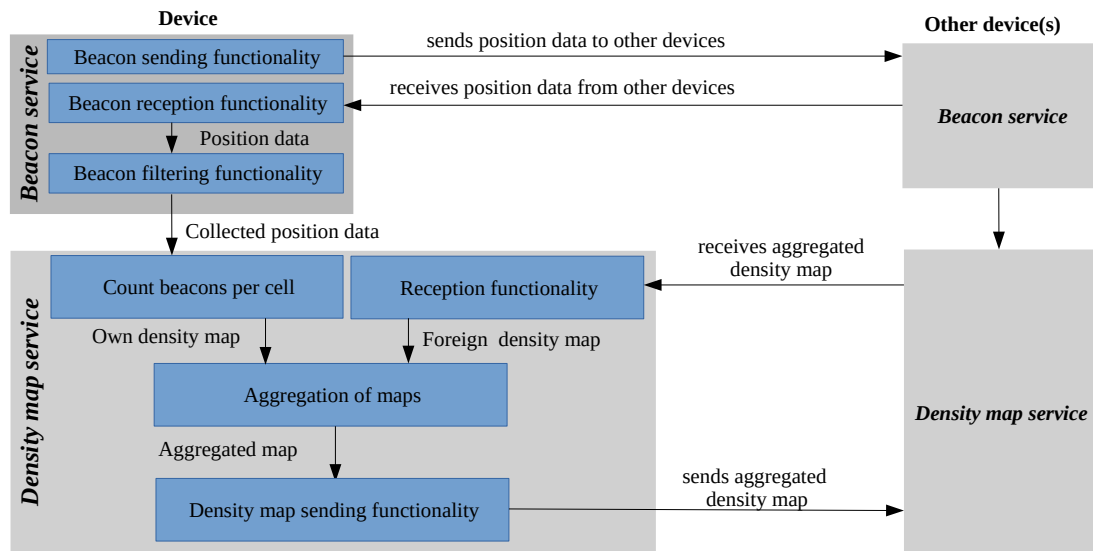
Figure 3.15.: Overview of services of the density map application [4]. Beacons and density maps are shared between devices using separate services. To improve the accuracy, density maps are aggregated.
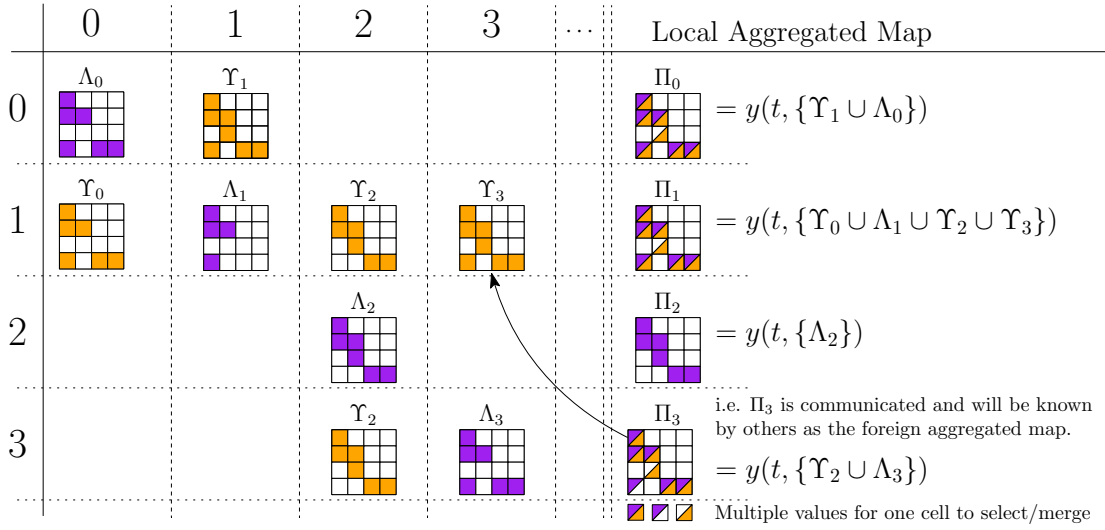
Figure 3.16.: Availability of density information for the aggregation. If a matrix element contains a map, a device (row) has received density information from another device (column). The diagonal is always occupied, as each device counts beacons (own estimate). The matrix is not symmetric because some nodes have more reception range than others. Graphics from [4]; permission granted by the creator S. Schuhbäck.

3.4.3. Applications for disseminating route recommendations

Two mobile applications were implemented in *OMNeT++* to disseminate route information via direct communication. The first application is implemented in the C++-class `UdpDetourAppVadere.cc`. It sends route recommendations at fixed intervals. The application is intended for simple crowd management applications where static detour information is sufficient and one can assume that people follow route recommendations. The recommended alternative route is changed directly in the crowd simulator by setting the target via the Traffic Control Interface, see Listing 3.4.

The second application enables the dissemination of dynamically generated route recommendations. As input, the app receives route recommendations provided by the *flowcontrol* simulator via the Traffic Control Interface, see Fig. 3.17. The application is implemented in the `BroadCastControlApp` application that is part of the compound module `CrownetUdpApp`. The `ControlManager` class processes the input data. The scheduler of the `CrownetUdpApp` schedules a packet that is immediately disseminated in the mobile network via the `BroadCastControlApp`. The reception of route recommendations is implemented in the `handleDataArrived()` method. As soon as a route recommendation is received, it is forwarded to *Vadere*, again using the Traffic Control Interface. Whether a node can send or receive route recommendations is configured through a boolean parameter.

```
1  void UdpDetourAppVadere :: actOnIncident (
2      IntrusivePtr <const DetourAppPacket > pkt) {
3
4    ...
5    std :: string blocked = std :: string (pkt ->getClosedTarget ());
6    std :: vector <std :: string > targetLists = ctrl ->getTargetList ();
7    if (std :: find (targetLists.begin (), targetLists.end (), blocked) !=
8        targetLists.end ()) {
9      std :: vector <std :: string > newTargetlist {};
10     for (int i = 0; i < pkt ->getAlternativeRouteArraySize (); i++) {
11       newTargetlist.push_back (pkt ->getAlternativeRoute (i));
12     }
13     ctrl ->setTargetList (newTargetlist); // change target in Vadere over
       TraCI
14   }
15 }
```

Listing 3.4: `actOnIncident()` method of the `UdpDetourAppVadere` application. The application is intended for simple crowd management applications with static detour information. It is assumed that all pedestrians follow route recommendations: When a route recommendation is received, the target of the agent is directly changed over the Traffic Control interface.
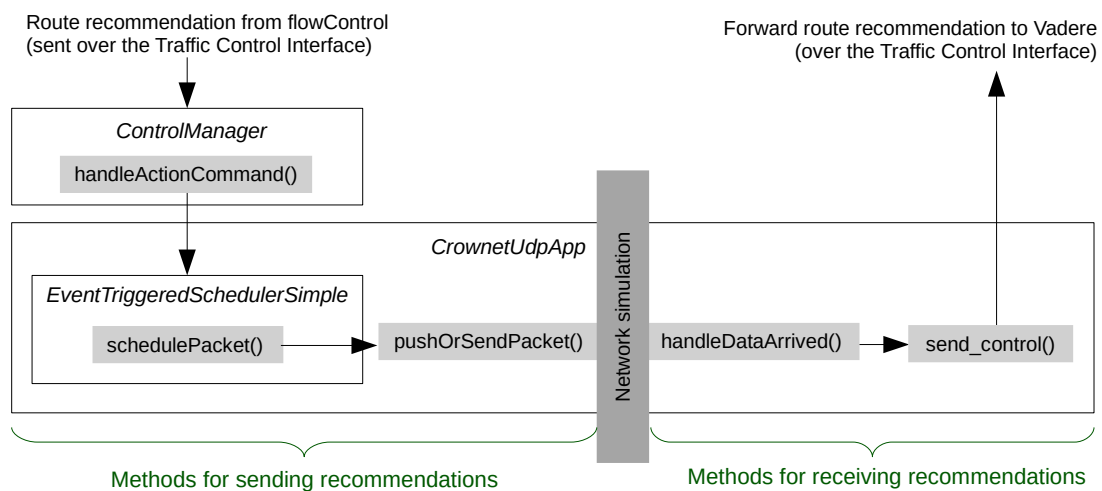
Figure 3.17.: `CrownetUdpApp` application and related classes. The class communicates via the Traffic Control Interface with the two simulators *flowcontrol* and *Vadere*. The `CrownetUdpApp` has sending and receiving functionalities. The `ControlManager` forwards route recommendations from the Traffic Control Interface to the `CrownetUdpApp application` (left). Received route recommendations are forwarded to the *Vadere* simulator (right). In the networks simulation the packet transmission is simulated.

3.5. Module Vadere: simulator extensions

The *Vadere* simulator [209] is used in *CrowNet* to simulate crowd behavior. *Vadere* has a psychology layer implemented which is essential for modeling behavioral changes in a crowd guidance system. *Vadere* contains several validated locomotion models, such as the Optimal Steps Model. To exchange data with other simulators, the Traffic Control Interface was implemented in [297]. In the following I analyze the state-of-the-art simulator. Then I describe the adjustments that were necessary to integrate *Vadere* in *CrowNet*.

3.5.1. State-of-the-Art simulator: analysis and required extensions

Vadere follows a model-view-controller software design pattern [209]. The `state` package ('model') contains the model classes that contain the state and the properties of objects, such as the size of an obstacle. The `simulator` package ('controller') is responsible for the execution of the models. The `gui` package ('view') contains *Vadere*'s 2D graphical user interface. The core of *Vadere* is the simulations loop, see Listing 3.5. At each simulation time step the layers of the psychology layer [155] are executed sequentially. Stimuli are specified in *Vadere*'s config file: the *.scenario file. It is not possible to generate stimuli dynamically and add them to the simulation online.

```
1 while (simulationIsRunning) {
2   ...
3   perceptionModel.update (agents, stimuli); // Perception
4   cognitionModel.update(agents); // Cognition
5   locomotionModel.update(agents,time); // Locomotion
6   time ++;
7   ...
8 }
```
Listing 3.5: Vadere simulation loop including psychology layer [305].

For each time step, the `StimulusController` extracts stimuli from the `ScenarioStore` object. Stimuli are always addressed to the entire population of agents. It is not possible to address stimuli to an individual agent. Therefore, it is not possible to simulate how an agent receives a mobile message.

At the perception layer the most important stimulus is selected which is then passed to the cognition layer where a self-category is selected in dependency of the most important stimulus and environmental conditions. Since each self-category is assigned exactly one specific behavior to keep it simple as simple as possible [155, p.95], one cannot simulate that people show different behaviors. In a crowd guidance system, some people follow the route recommendations and some others do not. Therefore, adjustments are necessary.

Parameters of the psychological models are hard-coded. Parameter studies are, therefore, only possible by changing the source code. Systematic uncertainty quantification is not possible.

I conclude that the state-of-the-art *Vadere* simulator (release 1.6, 2022) needs to be adapted. I derive the following requirements, see Tab 3.4.

Requirement	Design decision
Uncertainty quantification must be possible for psychological models.	Parametrize the psychology layer (see Section 3.5.2).
It must be possible to model and simulate the effect that agents respond differently to the same information.	Extend the functionalities of the psychology layer (see Section 3.5.3).
The reception of individual stimuli (mobile messages) must be possible.	Generalize the stimulus definition and processing (see Section 3.5.4).

Table 3.4.: Required extensions for the *Vadere* simulator release 1.16. To use *Vadere* as mobility provider in a crowd guidance system several adjustments are necessary. The design decisions (right) describe how the required adjustments (left) are implemented.

3.5.2. Parametrization of the psychology layer

To enable parameter studies with psychological models, *Vadere*'s psychology layer is parameterized. For this purpose, I introduce attribute classes for the models of the perception layer and the cognition layer. Each model class is assigned a unique attribute class. If the scenario is set up using the `VadereGui`, the corresponding attribute class is automatically inserted. Each attribute class has a corresponding `json` counterpart so that parameter values can be specified in the scenario file.

Listing 3.6 shows an example how a psychological model can be parameterized over the *.scenario file. The parameter values of the `ProbabilisticCognitionModel` are defined in the respective attribute class. The idea of the model is that agents take a route with a certain probability when they receive a route recommendation. The route probabilities are specified as vector, see `targetProbabilities` in Listing 3.6. Tests ensure that the sum of probabilities equals 1.

3.5.3. Enabling behavioral repertoires

In reality, people react differently even with the same environmental information due to their individual experiences and daily behavior. To allow different behaviors, the cognition layer is extended by the abstract class `AProbabilisticModel`, see Fig. 3.18.

```
1  "attributesPsychology" : {
2      "psychologyLayer" : {
3          "perception" : "SimplePerceptionModel",
4          "cognition" : "ProbabilisticCognitionModel",
5          "attributesModel" : {
6            " org.vadere.%.perception.AttributesSimplePerceptionModel" : {
7              "priority" : { // control priority (assign integers: 1, 2, 3)
8                "1" : "Threat",
9                "2" : "Wait",
10               "3" : "Information",
11           }},
12           "org.vadere.%.cognition.AttributesProbabilisticCognitionModel"
     : {
13             "routeChoices" : [ {
14             "instruction" : "Please take the long route (id=31)",
15             "targetIds" : [ 11, 21, 31 ],
16                          // control route probabilites
17             "targetProbabilities" : [ 0.25, 0.25, 0.50 ]
18           } ], ...
```

Listing 3.6: Parameterization of psychological models in Vadere's scenario file. The
          `targetProbabilities` parameter is a vector of probabilities. Each probability
          is assigned to a certain target (`targetIds`). If the long route is recommended
          (`instruction`), 25% of the agents take target 11, 25% take target 21. The
          majority (50%) takes the long route (target 31).
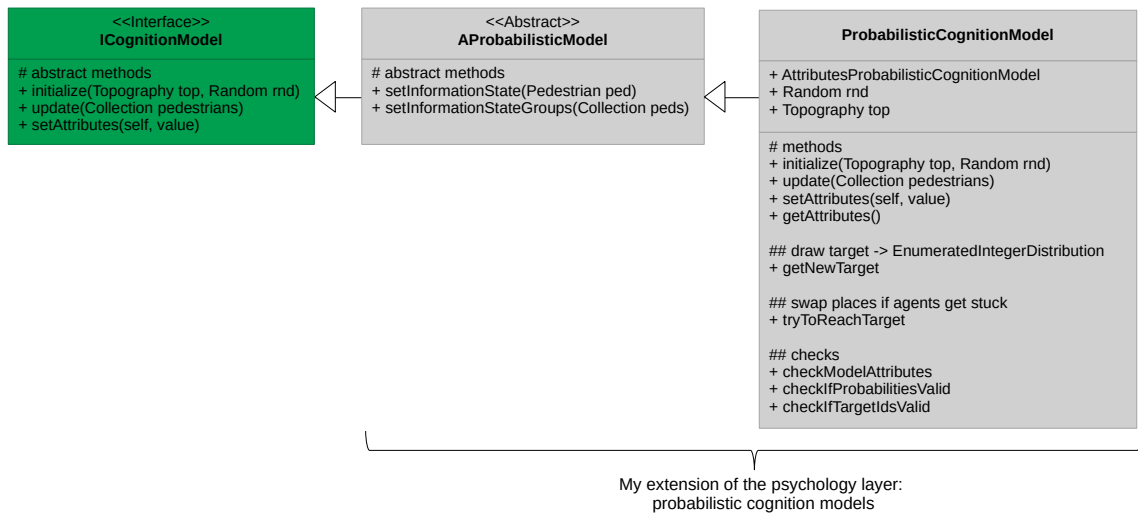
Figure 3.18.: Extending *Vadere*'s cognition layer. The abstract class `ProbalisticModel` is
          added to enable probabilistic choice modeling.

Instead of deriving the self-category from the most important stimulus, the self-category
is determined probabilistically: For each behavior a certain probability is assigned. An
example is the `ProbabilisticCognitionModel`, see again Listing 3.6. It models the route

choice behavior probabilistically. The routes 11 and 21 are taken with a probability of 25 % each and route 31 with a probability of 50 %.

For any probabilistic model, it is assumed that an agent makes a decision only when new information comes in. The decision-making is realized by drawing from a random distribution. If the same information is received again, it is ignored because it has been already processed. To achieve this, the information state of an agent must be stored. I introduce the `InformationState` that is an `enum` with the values `NO_INFORMATION`, `INFORMATION_RECEIVED`, `INFORMATION_CONVINCING_RECEIVED`, `INFORMATION_UNCONVINCING_RECEIVED`. Another enum item is `FOLLOW_INFORMED_GROUP_MEMBER` that is intended for group members who make decisions together. To display the `InformationState`, the GUI is extended.

3.5.4. Generalization of the stimulus definition and processing

The `SimulationCommandHandler` is introduced to add route recommendation stimuli dynamically to the simulation. The class handles `StimulusInfo` objects that are received over the Traffic Control Interface. It extracts the `StimulusInfo` object from the bit stream and adds the object to the `StimulusInfoStore`. Therefore, `StimulusInfoStore` contains static stimuli from the *.scenario file and dynamically received stimuli, see Fig. 3.19.
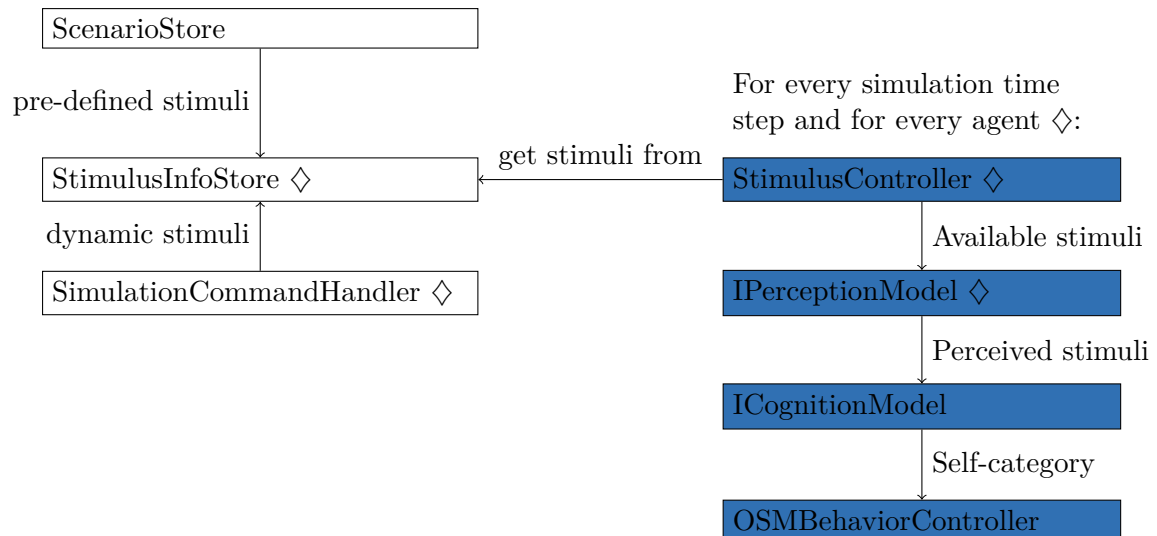
Figure 3.19.: Processing dynamic stimuli in *Vadere*. Novel or adjusted classes are marked with ◇. Stimuli are dynamically received over the Traffic Control Interface implemented in [297]. The adjusted `SimulationCommandHandler` adds the stimuli to the `StimulusInfoStore`. Since agents receive route recommendations at different times, it is necessary to address stimuli to specific agents. To achieve this, the filtering functionalities of the `StimulusController` are extended. Also, the `update` method of the `IPerceptionModel` is adjusted to enable the processing of agents-specific stimuli.

Due to delays, pedestrians may receive route recommendations at different times via their mobile applications. Therefore, it is mandatory that a route recommendation stimulus can be addressed to an individual agent. To achieve this, a `SubpopulationFilter` is added to the existing `StimulusInfo` class. The `SubpopulationFilter` has a list attribute that contains agents ids. For agents in the list, the stimulus is forwarded to the perception layer.

Route recommendations should be only displayed in areas where they are actually needed. In real life, the mobile application would check whether information should be displayed. In the simulation, this functionality is provided by a novel spatial filter: The `Location` attribute ensures that a stimulus is only generated for agents in a certain area. If the agent is in a certain area, the stimulus if forwarded to the perception layer.

The `SubpopulationFilter` and the `Location` are added to the `StimulusInfo` as attributes. The extended class has four attributes:

- `stimuli` of type `List<Stimulus>`: route recommendations, waiting signals, bangs, ... (unchanged)

- `timeframe` of type `Timeframe`: defines the time slot in which stimuli are active (unchanged)

- `location` of type `Location`: defines the spatial area in which stimuli are active (novel attribute)

- `subpopulationFilter` of type `SubpopulationFilter`: defines which agents have access to the stimuli (novel attribute)

The objects `timeframe, location` and `subpopulationFilter` are employed for filtering stimuli for space and time and address them to specific agents. The `StimulusController` and the interface `IPerceptionModel` are adjusted for this purpose. The filtering functionalities of the `StimulusController` are extended to allow spatial and agent-specific filtering. Note that the original implementation only supported a time-based filtering.

Because the provision and the perception of a stimulus are two separate processes, I introduce two stages:

- First the availability of a stimulus is checked by the `StimulusController`: Has the mobile device received a route recommendation and is this route recommendation displayed to the virtual pedestrian?

- Second the perception is modeled at the perception layer: How is the message perceived? Are there any other stimuli?

The separation of concerns is achieved by adjusting the `update` method of the `IPerceptionModel` interface. Instead of using a list that contains stimuli perceived by the entire population, a map is used. Each agent has an individual list of available stimuli assigned: `void update (HashMap<Pedestrian, List<Stimulus>> pedSpecificStimuli);`.

3.6. Module flowcontrol: novel framework for route recommendation algorithms

3.6.1. Scope, software architecture and requirements

flowcontrol is a novel Python simulation framework that I developed as part of this thesis. It is my major contribution to the *CrowNet* simulation framework. The scope of *flowcontrol* is to provide algorithm that generate route recommendations for crowd models. In *CrowNet*, *flowcontro*l is used in combination with the *Vadere* simulator. *flowcontrol* has a modular architecture and follows a model view controller pattern, see Fig. 3.20.
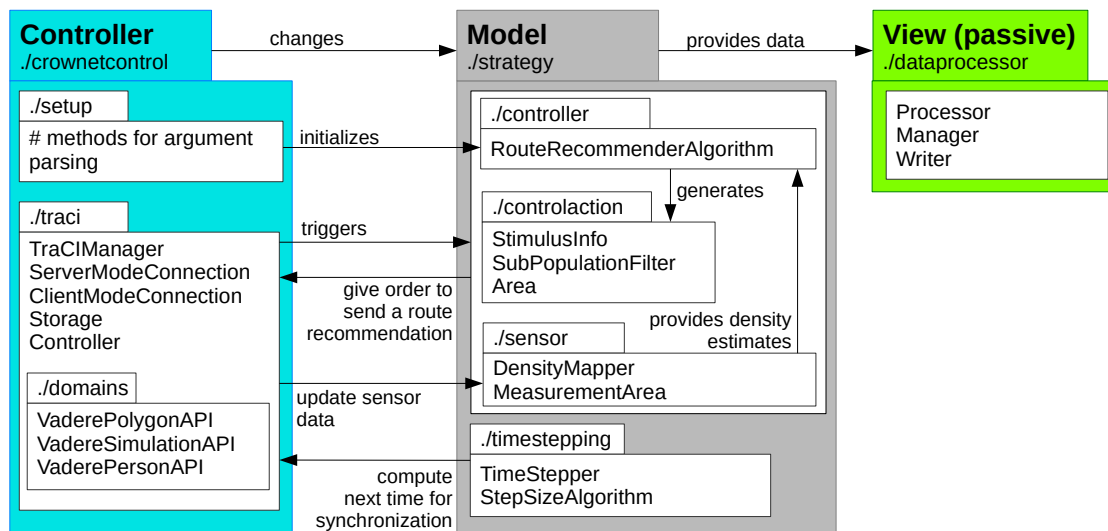
Figure 3.20.: Architecture of the novel simulation framework *flowcontrol*. The scope of *flowcontrol* is to provide route recommendations for pedestrian dynamics simulators, such as *Vadere*. Route recommendation algorithms, time stepping procedures and methods for mapping sensor data are implemented in the `strategy` package ('Model'). The data exchange with other simulators is handled in `crownetcontrol/traci` ('Controller'). *flowcontrol* uses the graphical user interface of the crowd simulator: The `dataprocessor` package ('View') provides additional data processors.

The `strategy` package ('Model') contains the core elements for generating route recommendations. It contains several sub-packages, see again Fig. 3.20. The `controller` subpackage contains the route recommendation algorithms. The `timestepping` sub-package

contains methods to control the time between two route recommendations. The sub-package `controlaction` provides representations for route recommendations. For example, the *Vadere* simulator requires that a route recommendation is a `StimulusInfo` object in JSON format. To couple *flowcontrol* with another pedestrian dynamics simulator, one simply implements the simulator-specific representation of a route recommendation.

The package `crownetcontrol/traci` ('Controller') manages the transfer of route recommendations over the Traffic Control Interface. If the complete crowd guidance system should be simulated, the recommendation is sent to *OMNeT++*. *OMNeT++* simulates the transmission of the route recommendation in the mobile network. When the packet is received, the route recommendation is forwarded to *Vadere*. When the network simulation is not needed, *flowcontrol* sends the route recommendation directly to *Vadere*. Depending on the simulator combination, *flowcontrol* is either the client or the server, see again Fig. 3.4 in Section 3.2.

As most pedestrian simulators *Vadere* provides a graphical user interface and visualization tools which is why *flowcontrol* does not provide a user interface. The processors and export functionalities in the `dataprocessor` package ('View') only serve to verify the results.

For the development of *flowcontrol*, I adhere to the principles of software development: I use code quality measures (see Section 3.1.3) and define requirements, see Tab. 3.5. *flowcontrol* has an own continuous integration pipeline that is triggered with every commit. Unit tests verify the correct implementation of methods. Integration tests ensure that the simulator coupling works.

Requirement	Design decisions
Route recommendation algorithms for crowds must be provided that can be transferred to arbitrary scenarios; adding algorithms should be easy.	Provide algorithms that can be transferred to arbitrary scenarios; establish a reusable software structure by using suitable interfaces and a strategy pattern (see Section 3.6.2).
It must be possible to control the time between two evaluations of the route recommendation algorithm (time interval).	Separate the time stepping procedure from the logic of the algorithm and provide time stepping procedures (see Section 3.6.3).
It should be possible to use density information from the decentralized density maps application (Section 3.4.2) as input for the route recommendation algorithm.	Implement a procedure that extracts the necessary density information from the application data (see Section 3.6.4).

Table 3.5.: Main requirements for the novel *flowcontrol* simulator. The design decisions (right) describe how the requirements for the simulator (left) are implemented.

3.6.2. Route recommendation algorithms: the logic of guidance

The redirection logic of the route recommendation algorithm is implemented in the `strategy` module. To enable developers to easily extend the simulation framework, a strategy software design pattern is used. With this, new algorithms can be simply added by inheriting from the abstract class `RouteRecommendationAlgorithm`, see Fig. 3.21. The static helper method `get_controller_from_args` creates and initializes an object of the type `RouteRecommendationAlgorithm` defined in `run_script.py` (`--ctrl.controller-type`). The separation of algorithm logic and object initialization allows to extend the package easily which increases the re-usability and sustainability of the software.

As I have outlined in the state of the art, several algorithms have been proposed for guiding crowds (see Section 2.1.5). However, these are neither transferable to arbitrary scenarios, nor have they been tested under realistic conditions. Therefore I develop algorithms myself and implement them in the `strategy` package.

I suggest and implement the following algorithms:

- `AlternateTargerAlgorithm`: alternates the routes sequentially.

- `AvoidCongestionAlgoritm`: recommends an alternative route whenever the density on the direct route is higher than on the alternative route.

- `MinimalDensityAlgorithm`: recommends the route where the density is the lowest. If there are only two routes, the algorithm is identical to `AvoidCongestionAlgorithm`.
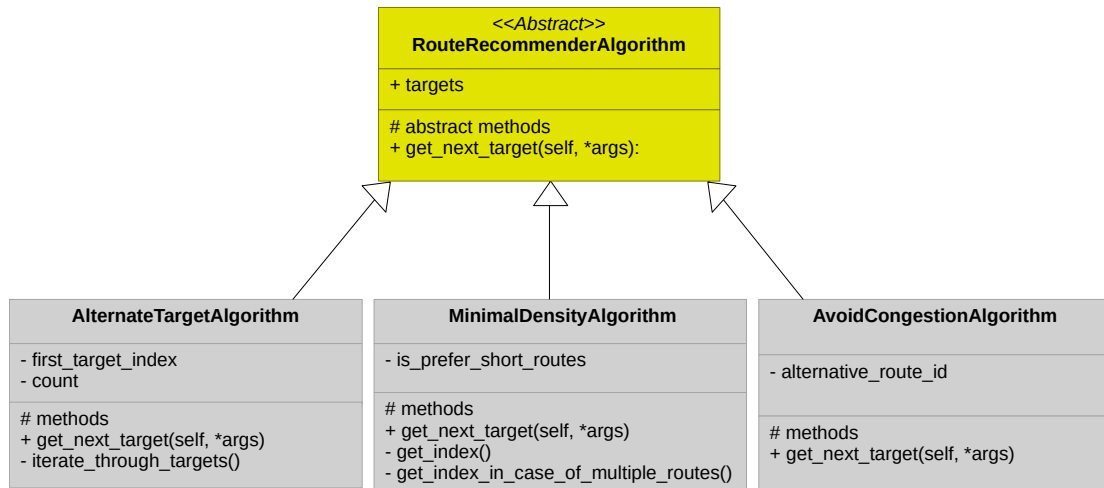
Figure 3.21.: Important classes for the implementation of route recommendation algorithms in the `strategy` module. The core of the module is the abstract class `RouteRecommendationAlgorithm`. The logic of a route recommendation algorithm is implemented in the `get_next_target()` method of the respective child class.

3.6.3. Time stepping procedure

A core functionality of a dynamic crowd guidance system is the scheduling of route recommendations. In *flowcontrol* this functionality is implemented in the `timestepping` module. *flowcontrol* offers different types of time stepping procedures that trigger the generation of route recommendation at regular or irregular time intervals, see Fig. 3.22.
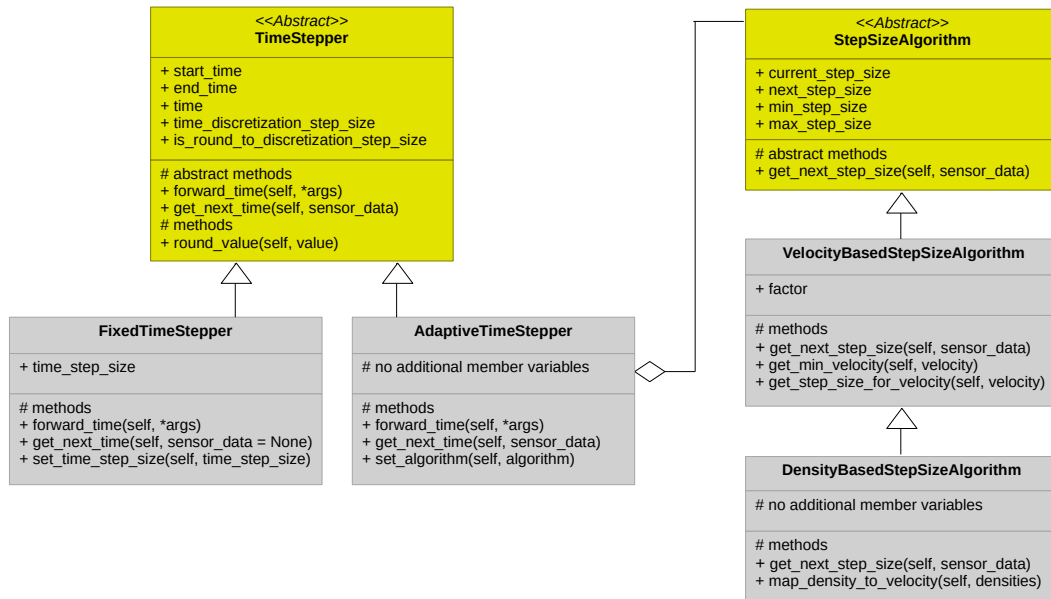

Figure 3.22.: Time stepping procedure. The abstract class `TimeStepper` controls the time in between two route recommendations. The time interval can be either fixed (`FixedTimeStepper`) or dynamically adjusted during the simulation (`AdaptiveTimeStepper`). For the latter, a `StepSizeAlgorithm` is required that computes when the route recommendation algorithm should be evaluated next.

With a static stepping procedure, the time between two route recommendations is always the same. A dynamic stepping procedure adjusts the time between two route recommendations dynamically. A dynamic adjustment may help to reduce the number of redirection measures. This might be particularly helpful when the number of pedestrians changes dynamically over time: At a low arrival rate few redirection measures may be needed (large time interval) to dissolve congestion. At a high arrival rate (small time interval), pedestrians may need to be redirected more often. Therefore, the `StepSizeAlgorithm` interface is introduced. Its core method is the abstract method `get_next_time_step_size()` that has to be implemented by the child classes.

3.6.4. Extracting density information from density maps

Some route recommendation algorithms require density measurements. Densities are measured in areas where the maximum densities occur. I want to measure densities based on direct communication technology which is why I propose to use the decentralized density map application [4]. However, a measurement area does not always correspond to a specific cell of the map, see Fig. 3.23.
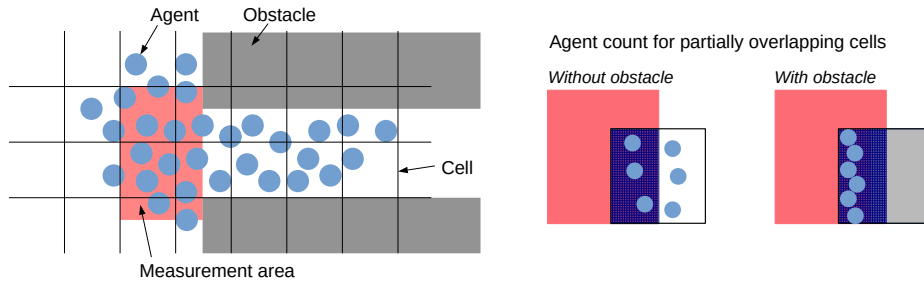

Figure 3.23.: Mapping of density data. Since the measurement area is does not correspond to one cell of the density map application [4], a mapping is required. If obstacles cover parts of the cell, the reference area is small which leads to high densities (right).

The `sensor` package provides a mapping procedure that computes the average density from cell-based data. The averaging procedure is implemented in the `DensityMapper` that has information about the cell layout, the measurement areas and the obstacles, see Fig. 3.24. Since the density map application does not consider obstacles, densities could be locally underestimated which is not acceptable for crowd management applications. A simple correction procedure is implemented: The percentage of available space is determined for each cell. Then the density values are divided by the percentage values, given that a cell is not fully occupied by an obstacle.
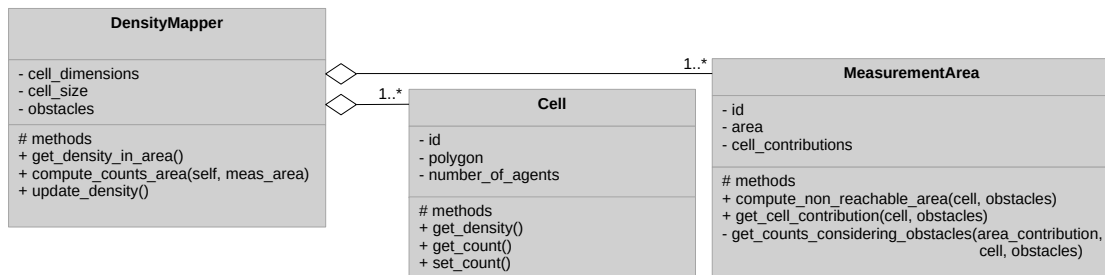

Figure 3.24.: Important classes for the density estimation. The `DensityMapper` has information about the measurement area (position, size) and the cell-grid (cell size, cell position). The `DensityMapper` estimates the crowd density in a measurement area from a density map [4].

3.7. Summary

With *CrowNet*, I created a new open-source framework to simulate a crowd guidance system where crowds are sensed and redirected using direct communication technology. First, I modeled a crowd guidance system composed of a crowd, a controller containing the route recommendation algorithm, a mobile application for sensing the crowd, and a mobile application for communicating route recommendations. Based on the model structure, I designed a modular software architecture. For the implementation, I defined functional and non-functional requirements and made design decisions to be realized in the corresponding modules. In addition, quality measures were defined to ensure software quality and sustainability. A continuous integration and deployment pipeline was introduced to test the source code during the development process. The core functionalities of the software were then implemented in the five main modules.

The *crownetutils* module provides implementations for coupling models and simulators. An explicit update procedure was proposed and implemented, interpolating agent positions in the mobile communication simulation. To separate the inter-process communication of simulators from parallel executions and from the host system, the coupled simulation are executed in a containerized fashion, employing Docker networks. The state exchange between simulators is realized via the Traffic Control Interface. Several simulator couplings have been implemented for the simulation of the entire crowd system and for isolated subsystems.

The *SUQ-controller* enables to conduct parameter studies with coupled simulations. The module is based on the existing Python package *SUQ-controller*. New process calls were added in order to be able to run coupled simulations. The definition of parameter combinations has also been extended. Parameters can now be changed in various simulator-specific config files. To ensure that only Docker containers communicate that refer to a specific parameter combination, a naming convention for the containers has been introduced.

The *OMNeT++* module enables the simulation of direct communication. Existing simulators were employed as model libraries for the lower layers of the network. To sense crowds, a novel method based on LTE sidelink communication was implemented: the decentralized pedestrian density map application. Two further mobile applications were implemented to disseminate route recommendations, which differ in their static or dynamic information content.

The *Vadere* module provides crowd models. The analysis of the state-of-the-art simulator showed that several extensions were required for the integration of *Vadere* into *CrowNet*: First, I generalized the stimulus definition and the processing of the stimuli in *Vadere*'s psychology layer to enable dynamically generated route recommendations to be processed. Second, I introduced an interface for modeling probabilistic behavior in the cognition layer to enable probabilistic route choice. Third, I parameterized the

psychology layer to permit parameter studies with behavioral models. With these extensions, route recommendations can be dynamically processed in *Vadere*'s psychology layer.

The module *flowcontrol* has been developed for the provision of route recommendations. For this novel Python framework I first defined requirements from which I derived design decisions. Since state-of-the-art route recommendation algorithms did not fulfill my requirements, I proposed novel algorithms and implemented them. To control the time interval between two route recommendations, a generic time-stepping procedure has been implemented. To handle sensor data, a method was implemented that maps cell-based density values, provided by the density map application, to measurement areas. The *flowcontrol* framework can be combined with other crowd simulators by implementing appropriate interfaces.

In the following chapter, I will use the *CrowNet* software as a simulation tool to investigate my research question.

4. Guiding crowds using direct communication technologies

In this chapter, I investigate how crowds can be redirected using direct communication technologies. For readers who are only interested in how I answer the main research question (RQ), I recommend to go directly to Section 4.4. If they have questions related to previous findings, that the study is based on, they can jump back. I answer my research question following a step-by-step approach, see Fig. 4.1.

I first address my research sub-questions that relate to isolated components of my crowd guidance system. In Section 4.1, it is investigated how reliably route recommendations are disseminated using direct communication in a crowd (sub-question RQ-1). I introduce a criterion to evaluate the reliability of information dissemination in a crowd and employ it in a simulation study. In Section 4.2, the suitability of route recommendation algorithms is assessed under the condition that not all crowd members follow instructions (sub-question RQ-2). The effect of three heuristic algorithms on safety and comfort-related quantities, such as the level of service, is compared. From this study, I select the most suitable algorithm for my crowd guidance system. In Section 4.3, the third research sub-question (RQ-3) is investigated, that is, how route recommendations should be designed in mobile messages so that crowd members follow instructions. The Sections 4.1-4.3 are structured in the same way. First, the research scenario is presented, followed by the study design. Then, the results are discussed. At the end of each section, a summary and a conclusion ('Lessons learned') are provided where I point out what the findings mean for the design of a crowd guidance system.

All the individual scenarios presented in Sections 4.1-4.3 were inspired by an application scenario at the metro station Münchner Freiheit in Munich, Germany. Since I had to abstract and adapt the application scenario to create laboratory conditions to properly answer my research sub-questions, the scenarios differ in the plot, in the topography and in the resulting pedestrian streams. Readers who are interested in the application scenario, find a detailed description in Section 4.3.

In Section 4.4, I answer the main research question (RQ). I propose and investigate a complete crowd guidance system. To test my concept, I assess the effect of the crowd guidance system for the application use case Münchner Freiheit (Munich, Germany), where footballs fans are guided from the bus to the train to resolve congestion.

All simulation studies presented in this chapter were conducted with the *CrowNet* software (Chapter 3). For the survey, ethical approval was obtained from the ethics committee of the Munich University of Applied Sciences, see Appendix F.
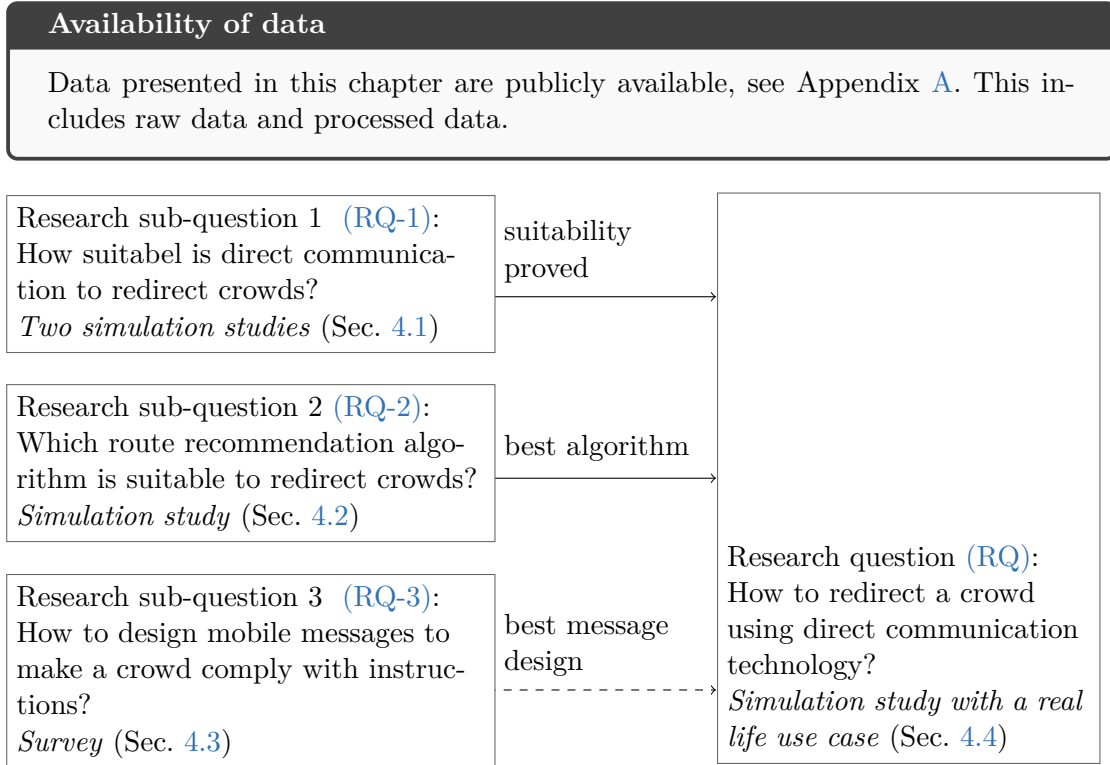
> **Availability of data**
>
> Data presented in this chapter are publicly available, see Appendix A. This includes raw data and processed data.

Research sub-question 1 (RQ-1): How suitabel is direct communication to redirect crowds? *Two simulation studies* (Sec. 4.1)	suitability proved →	
Research sub-question 2 (RQ-2): Which route recommendation algorithm is suitable to redirect crowds? *Simulation study* (Sec. 4.2)	best algorithm →	Research question (RQ): How to redirect a crowd using direct communication technology? *Simulation study with a real life use case* (Sec. 4.4)
Research sub-question 3 (RQ-3): How to design mobile messages to make a crowd comply with instructions? *Survey* (Sec. 4.3)	best message design - - - →	

Figure 4.1.: Structure of the chapter. In the Sections 4.1-4.3, I answer research sub-questions. My proposal of a crowd guidance system and a proof of concept can be found in Section 4.4.

4.1. Information dissemination in a mobile crowd

In this section, it is investigated how reliably route recommendations are disseminated through direct communication technologies in a mobile crowd (sub-question RQ-1). To answer the question, I design a worst-case scenario. If the information dissemination succeeds, I expect that it will be reliable in many real-life use cases. The worst case is that interference and shadowing lead to a breakdown of the information dissemination. While interference might be controlled through protocols, shadowing is determined by the topography: Physical obstacles not only limit the line of sight but also hinder mobile communication.

The crowd and the mobile network form a complex socio-technical system. If the crowd size is loose and crowd members may be isolated, shadowing affects the information dissemination and, thus, the crowd dynamics. In dense crowds, the effect of interference will be dominant. To better understand the system's dynamics I divide the study into two parts. The first part focuses on the effect of shadowing when the crowd is loose. The second part focuses on the reliability of the information dissemination process in

a dense crowd when there are many crowd members to relay the information around obstacles. The questions to be answered are:

- Sub-study 1 (in the presence of shadowing): How many crowd members are necessary in the worst-case scenario that shadowing does not disturb the information dissemination?

- Sub-study 2 (no impairment through shadowing): How fast is the information disseminated? What is the effect of uncertain simulation parameters?

The section is structured as follows: First, a worst-case research scenario is designed where the mobile communication in a crowd is blocked by obstacles. Then a criterion is proposed that evaluates the reliability of the information dissemination in a mobile crowd. In particular, I discuss why microscopic quantities of interest from the research field of mobile networks are not suitable for my investigations. Then, the model of the crowd guidance system is presented, where a crowd should receive static detour information about a closed gate. In both simulation studies, the same scenario, model and implementation is used. The difference between the studies is the set of analyzed parameters and the research methodology. At the end of the section, the results are summarized and discussed. Then, I evaluate what these findings implicate for the design of my crowd guidance system.

4.1.1. Research scenario

Imagine a complex building, such as a train station with multiple entrances and exits, where an entrance or exit is suddenly closed. Since walking routes are not directly visible due to walls or other obstacles, it is necessary to inform passengers about the detour. For this purpose, a mobile application based on direct communication technology should be used. When information is disseminated over the mobile network, pedestrians might be separated by obstacles (shadowing). Due to that, detour information may not be disseminated successfully over the mobile network. To investigate this effect, I pose several requirements for the research scenario:

1. Pedestrians have a limited view: they cannot see the closure of the route directly.

2. The information about the closure is distributed via a mobile application based on direct communication.

3. Communication between shadowed pedestrians is only possible via intermediate nodes or by them changing position, which resolves shadowing.

4. Only people who have personally seen the closure on site generate and send redirection information (there are no sensors that detect the closure automatically).

The last point implicates extreme conditions: The closed gate is located in a niche so that pedestrians, who provide information, are completely shadowed from the area where the information is needed. Hence, the conditions are more restrictive than in the scenario from my related publication [295], see Appendix D.

The topography is depicted in Fig. 4.2. It covers an area of $176 \, \text{m} \times 129 \, \text{m}$. Crowd members at a train station take different routes to get to the train. Suddenly, the left path is closed, forcing pedestrians to take a different route. The agents are generated from four sources at the north and walk along different routes to the destination at the south. The number of agents can be controlled through a source parameter. After $100 \, \text{s}$, an agent next to the closed gate decides to inform others about the closure. It generates a route recommendation and sends it via a mobile application as a groupcast to all agents in the scenario. The technology used is direct communication based on the 802.11p WLAN standard. Agents that receive the detour information forward it.

As soon as agents receive the rerouting information, they take the alternative route. This is based on the assumption that all people follow the route recommendations, which is optimistic. In this study, I will refrain from examining the pedestrians' compliance to follow route recommendations because it would be computationally too expensive, and I refer to Section 4.2, in which the effect of compliance is investigated.

4.1.2. Reliability of information provision

In the research field of mobile communication, there are several metrics and quantities for evaluating the performance of a network, such as latency or jitter. These quantities are microscopic and refer to individual packet transmissions which is important to look at when designing network technologies. However, these metrics do not consider the information state of the crowd as a whole. Therefore, they are not suitable for my investigations.

Moreover, mobile communication and crowd locomotion have different temporal and spatial scales. Packets in a mobile network are transmitted within a few milliseconds over hundreds of meters. The average pedestrian is much slower with a typical speed between $1 \, m/s$ and $1.5 \, m/s$. Also, pedestrians need time to process information. An information delay of a few milliseconds does not affect the locomotion of the crowd as a whole.

I propose the following macroscopic quantities to evaluate the information process in a mobile crowd:

- *Information degree p*: percentage of crowd members that have received detour information

- *Dissemination time t_{diss}*: time between the point of time when the route recommendation was sent the first time and the time when a certain proportion p of the crowd members has been informed. I propose $p_{threshold} = 100 \, \%$ for closed system

(no arrivals, no departures). For open systems where people continuously arrive or leave the scenario, I propose to choose $p_{threshold} \leq 100\,\%$ depending on the arrival and departure rate. In the two sub-studies, $p_{threshold} = 95\,\%$ is used.
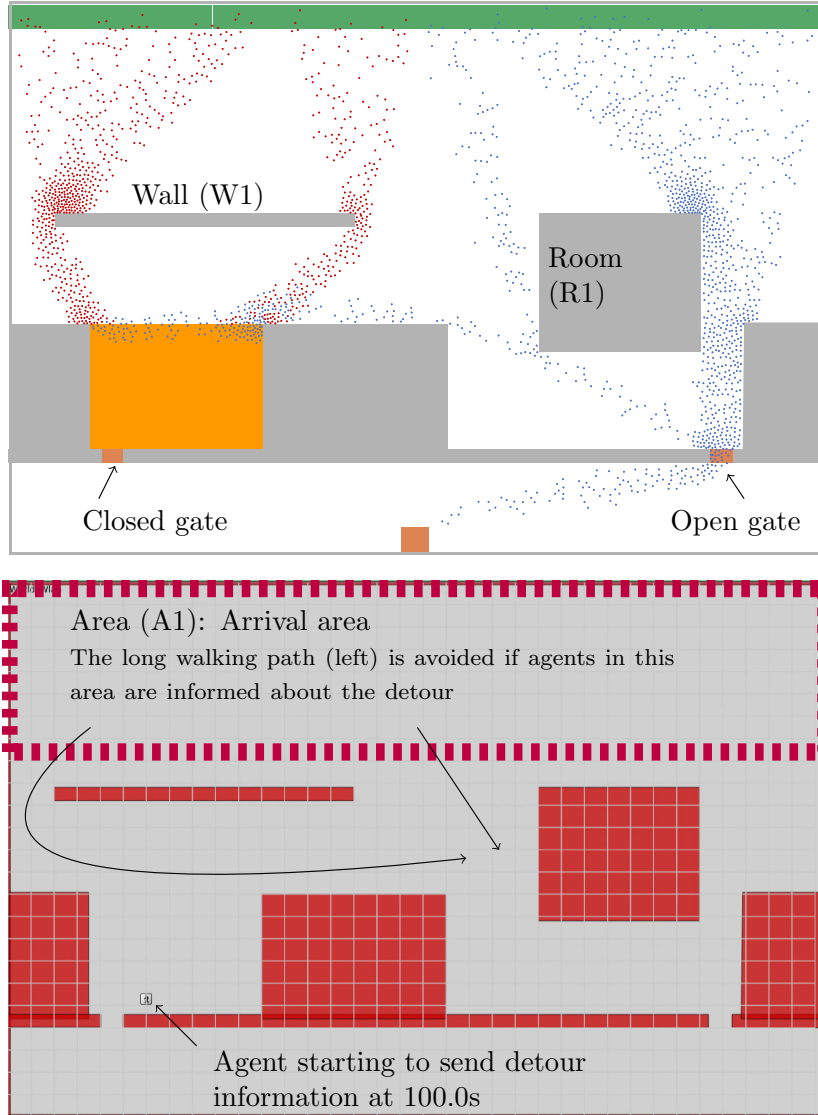

Figure 4.2.: Worst case scenario for investigating information dissemination in a mobile crowd. Representation of the scenario in *Vadere* (top) and *OMNeT++* (bottom). The size of the scenario is $176\,\text{m} \times 129\,\text{m}$. The grid with $5\,\text{m}$ cell width provides orientation (bottom). The agents (blue and red circles) walk from the green source (top) to the target (bottom) on different routes. Agents marked red take a detour, since the left gate (brown area) is closed. The orange area is the area where the closure is in their line-of-sight, so that they start to detour.

To evaluate the reliability of information dissemination, I propose the following criterion:

- The information dissemination is reliable if the *dissemination time* t_{diss} does not exceed the threshold value $t_{threshold}$: $t_{diss} \leq t_{threshold}$. Otherwise, if $t_{diss} > t_{threshold}$, information propagation is not reliable.

As threshold value I choose $t_{threshold} = 10\,\text{s}$ because within $10\,\text{s}$ even running pedestrians $(3.0\,\text{m/s})$ receive the information within the arrival area $(176\,\text{m} \times 30\,\text{m})$ where a change of the route choice would not increase the length of the walking path.

To check the plausibility of the simulation results, I compare the *dissemination time* with the statistics of the microscopic quantity of interest *packet lifetime*. The *packet lifetime* describes how long the transmission of a packet takes. I expect that the maximum value of the *packet lifetime* and the *dissemination time* have the same order of magnitude. I do not expect that they are equal, since information is transmitted over intermediate nodes. If so, multiple packets are necessary, and, the time it takes to receive information becomes the sum of all *packet lifetimes.*

4.1.3. Model and implementation of the crowd guidance system

The simulation model of the crowd guidance system is composed of several component models, see Tab. 4.1. In the following, the component models are briefly introduced. It is explained why a certain model is chosen for the investigation and hot it is implemented in the *CrowNet* simulator.

Crowd behavior

I assume that all people follow the recommendation to take the alternative route immediately. This assumption may be optimistic, but it should not affect the macroscopic quantities of interest: Even if only a part of the crowd followed the route recommendation, the left stream in Fig. 4.2 would become wider because non-compliant agents continue to walk downwards while compliant agents turn right. Nevertheless, the agents form a connected graph within the arrival area. As a consequence, the topology of the network would be similar. The transmission of individual packets might change, but this should not affect the macroscopic *dissemination time*.

After receiving detour information, the target of agent is directly changed without modeling the perception or cognition. The targets are set in the `UdpDetourApp` (*OMNeT++*) when an agent receives the information over the mobile network, see Tab. 4.1. The `UdpDetourApp` then triggers the target change in *Vadere* via the Traffic Control Interface (TraCI). To model the locomotion behavior, the Optimal Steps Model is used that is implemented in the *Vadere* simulator.

Component	Sub-component	Model	Implementation (Simulator)
Crowd	Locomotion model	Optimal Steps Model	OptimalStepsModel (Vadere)
	Perception model	Assumption: info is always perceived	impl. in UdpDetourApp (CrowNet)
	Cognition model	Assumption: 100% compliance	UdpDetourApp (CrowNet/app.)
Controller	Time stepping alg.	Fixed interval (default: 1s)	UdpDetourApp (CrowNet/app.)
	Route recommendation alg.	None (static information: Gate closed)	Not needed
Network	Application layer	Detour application	UdpDetourApp (CrowNet/app.)
		Receive and forward information	UdpDetourAppVadere (CrowNet/app.)
	Transport layer	Udp	Udp (inet)
	Network layer	Ipv4	Ipv4NetworkConfigurator (inet)
	Data Link layer	acc. IEEE 802.11	Ieee80211MgmtAdhoc (inet)
	Physical layer → Channel model	acc. IEEE 802.11	Ieee80211DimensionalRadio and Ieee80211DimensionalRadioMedium (inet)
	→ Obstacle Model	Ideal Obstacle model	IdealObstacleLoss (inet)

Table 4.1.: Component models for investigating information dissemination in a crowd. If not specified, I use parameter values and settings for the component models. The mobile network model is composed of the protocol stack and the channel model. Importantly, the ideal obstacle model is used to model transmission failures due to shadowing: The transmission fails if agents are not in line-of-sight.

Route recommendation generation

In the scenario, there is no need for a sophisticated route recommendation algorithm because there is only one route left: This route is recommended all the time. Thus, there is no need to integrate the *flowcontrol* simulator. Instead, a simple redirection model is implemented in the `UdpDetourApp` application of the *OMNeT++* module, see Fig. 4.3. The redirection model sends static information ('The left gate is closed. Please use the right gate').

Information dissemination through direct communication

In my investigations I want to test the worst case. Therefore, I suggest to use direct communication according to the IEEE 802.11p standard. If one can inform a mobile crowd based on this technology, it is likely that more advanced technologies based on IEEE 802.11bd or cellular communication may also be suitable. Therefore, models for the 802.11p standard are selected on the link and physical layer. The link layer model
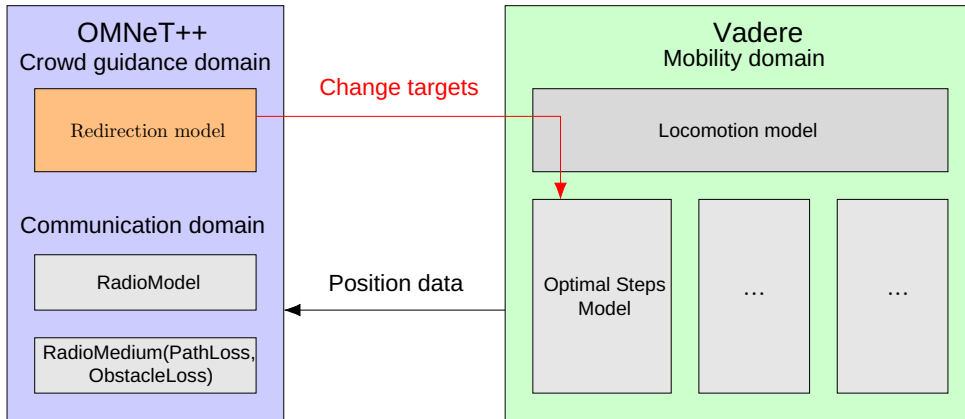
Figure 4.3.: Simulation approach in the information dissemination study. The simulators *OMNeT++* and *Vadere* communicate over the Traffic Control Interface. When detour information has been received in the mobile networks simulation (left), the targets are changed in the crowd simulation (right). Agents' positions from the crowd simulation are shared at the simulation time step of the *Vadere* simulator (0.4s). Image as in my publication [295].

handles the sending and reception process of data frames that are passed from device to device. There are no control or management frames. As implementation the module *Ieee80211MgmtAdhoc* from the *INET* framework is used.

The signal modeling is specified at the physical layer. To model the signal power in the analog domain, a models are used that describes the change of the signal power over time and frequency: the *Ieee80211DimensionalRadio* model in conjunction with the *Ieee80211DimensionalRadioMedium* model from the *INET* library.

To capture the signal attenuation caused by obstacles, I choose the Ideal Obstacle Model to model the worst case: If there is an obstacle in line-of-sight, there is no signal strength at all. In a real-life use case, the signal would be attenuated. Therefore, I believe that information dissemination is successful in real life, when it is successful in the simulation because conditions are milder. Using the Ideal Obstacle Model has another advantage: No permeability of the obstacles needs to be defined which keeps the dimensionality of the parameter space low. As implementation the *IdealObstacleModel* from *INET* is used.

The sending of the route recommendation is modeled as a mobile application on the application layer. When the application is started, it sends the redirection recommendation at fixed time intervals. Packets are broadcasted via a link local broadcast. No routing protocol is used. The application is implemented as a finite-state machine in the *OMNeT++* module of the *CrowNet* simulator, see Fig. 4.4. Note that in the scenario only one agent next to the closed gate sends route recommendations. Other agents only receive and forward the information.

Every agent has an application assigned that receives and forwards detour information. The application manages the change of the targets over the Traffic Control Interface. As soon as the detour information is received, the destination contained in the route recommendation is set for the agent. Then, the packet is forwarded via broadcast. An internal timer ensures that the route recommendation is only processed once. This helps to avoid a broadcast storm. As implementation, the *UdpDetourAppVadere* application is implemented in the *OMNeT++* module.
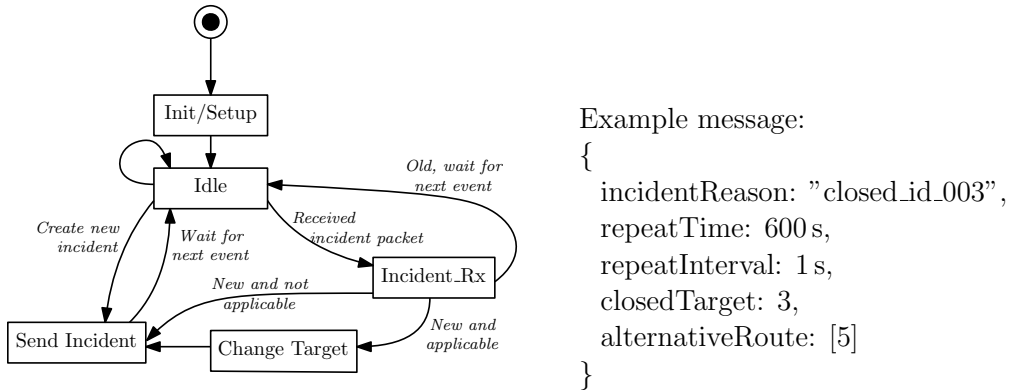
Figure 4.4.: Controller model generating and forwarding detour information. Since the left gate (target id=3) is closed, agents are redirected to the right gate (target id=5). The information is provided in time intervals of 1 s. Image as in my publication [295].

4.1.4. Minimal crowd size to avoid shadowing

In the first sub-study, I want to find out the minimum number of crowd members necessary in my scenario to ensure that the dissemination of route recommendation does not fail due to shadowing. I use the minimum number in the next sub-study as lower bound for an uncertain parameter when conducting an uncertainty quantification study.

Methodology: varying the crowd size

To find out how many crowd members are needed to avoid shadowing, the parameter *number of agents* is varied between 20 and 150. I choose a lower bound of 20 agents due to the following consideration. To prevent shadowing, at least 8 nodes (2×4) are needed: four agents around the wall (W1) and four agents around the room (R1), see Fig. 4.2. Another node is required to transport the information to the target area. Since agents' positions are continuously changing, I choose 20 nodes to be on the safe side. I choose an upper bound of 150 agents because one can assume that agents form a connected

chain which prevents the effect of shadowing. In total, 500 samples are generated which requires to run the simulation 500 times.

Since the number of agents is still low, it is likely that the information distribution fails due to shadowing rather than interference. Therefore, I assume, that shadowing is present when the information dissemination fails, that is, when less than 95 % of the agents are informed after 10 s: $p = 0.95, t_{threshold} = 10$ s.

The information dissemination fails due to shadowing

To ensure that shadowing causes the dissemination failure I randomly select samples with $t_{dissemination} > 10$ s. I observe that the crowd is indeed temporarily separated by obstacles during the information dissemination process (simulation time $\in [100$ s, 110 s]), see Fig. 4.5. One can observe three separated crowds: A crowd on the left, a crowd on the right and a crowd at the bottom, each represented by blue connecting lines between them. The left crowd is in line-of-sight with the crowd on the right which allows them to pass on information. However, the crowd at the bottom is still shadowed by a wall, so that the four agents close to the target are not informed within 10 s. As a result, the information degree is only 81 % after 10 s (17 of 21 agents have been informed). The visual check strongly indicates that the information dissemination indeed fails due to shadowing when the parameter *number of agents* has a low value.
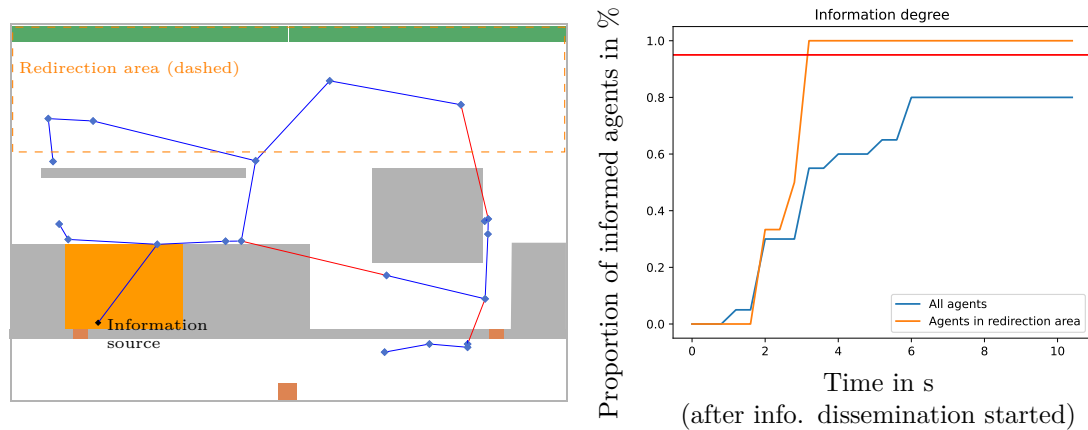
Figure 4.5.: Partially informed crowd due to shadowing. The crowd is separated by obstacles (see red lines through obstacles on the left). As a consequence, only 17 out of 21 (81 %) agents in total are informed after 10 s (right). Agents in the arrival area have been informed successfully after 3.2 s (intersection of the orange line and the red threshold line on the right) because agents form a chain in this area (see blue lines connected to the information source on the left).

Probability for shadowing in dependency of the crowd size

To determine the probability for shadowing in the scenario, the values of the parameter *number of agents* are grouped into bins. For each bin, it is counted how often shadowing was present. To obtain the probability the count is normalized by the total number of samples. The result is depicted in Fig. 4.6.
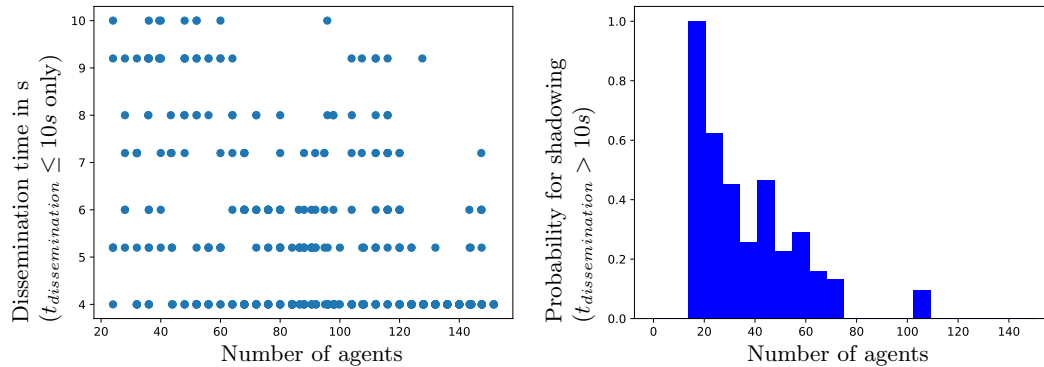

Figure 4.6.: Information dissemination time and probability for shadowing. The information dissemination is successful when the dissemination time is $\leq 10\,\mathrm{s}$ (left). Note that the information dissemination time is a multiple of the time step size $0.4\,\mathrm{s}$. The more agents, the less likely shadowing is (right). In the research scenario, shadowing is no longer present when the crowd has more than 110 members.

As expected, the probability of shadowing decreases over the uncertain parameter *number of agents*. In the scenario, there appears to be no more shadowing with more than 110 agents. Below this threshold, shadowing is likely to occur. However, one cannot predict whether shadowing occurs for a specific sample, since the crowd locomotion is stochastic: Agents' walking speeds are randomly drawn from a normal distribution. A change of the seed or a slightly different value of a parameter changes agents positions. One can observe that communication can even succeed if the *number of agents* is small: For 30 agents only, the *dissemination time* is less than $5.2\,\mathrm{s}$, see Fig. 4.6 (left). However, this is a stochastic effect that cannot be controlled.

Summary of the shadowing study: shadowing is prevented with a sufficient dense crowd

The first sub-study suggests that with enough well distributed agents present, loss of information through shadowing can be completely avoided. In the scenario 110 agents sufficed to spread detour information in a crowd in less than 10 seconds. I conclude this is enough time to inform agents about a closure before they take a closed route.

4.1.5. Effect of uncertainties on the information dissemination

Motivation for quantifying uncertainties

The crowd size in real-life scenarios, like at a train station or a festival, is often uncertain. Apart from the crowd size, there are additional unknowns, such as the transmitter power of smartphones. I aim to understand how such uncertainties impact the reliability of the information dissemination. Therefore, uncertainty quantification methods are applied. I choose a surrogate-based approach with polynomial chaos expansions to reduce the computational effort because a single simulation run can take several hours.

Forward propagation is employed to assess the effect of uncertain parameters on the quantities of interest. A global sensitivity study is conducted to quantify the influence of parameters.

The sub-study is structured as follows. First, the set of parameters and quantities of interest is introduced. Second, the simulation pipeline is described which comprises the construction of the polynomial chaos expansions. Finally, the results are discussed. I expect the reader to be familiar with the theory on polynomial chaos expansions, forward propagation and global sensitivity analysis (see Section 2.5).

Parameter selection for the uncertainty quantification

I consider two uncertain parameters that I think are particularly interesting for a crowd guidance system: the *number of agents* and the *transmission interval*, see Tab. 4.2. The parameter *transmitter power* only serves as a control.

Parameter	Unit	Distribution	Sample points
Number of agents n	1	U(200,2500)	415, 1325, 2235
Transmitter power p	mW	U(2.0, 20.0)	4.03,11.00, 17.97
Transmission interval si	s	U(0.8,2.8)	1.0, 1.8, 2.6

Table 4.2.: Uncertain parameters in the information dissemination study. The three parameters are used in the forward propagation and in the global sensitivity analysis. Since the shape of the distributions are unknown, I use uniform distributions. Three sample points for each parameter are needed to construct a polynomial chaos expansion of order $= 2$. In total, there are $27 (= 3^3)$ samples.

The first parameter is the *number of agents*. The previous part of the study demonstrated how communication fails due to shadowing when the *number of agents* is low. A high *number of agents* can, in turn, lead to interference in the mobile network or congestion

in passenger traffic. Thus, this parameter controls the dynamics of the complex socio-technical system composed of the crowd and the mobile network. Since I do not know the distribution type of the parameter, I use a unit distribution. As lower bound I use a *number of agents* of 200 (the previous study required at least 110 to prevent shadowing). As upper limit I choose 2500. More agents would lead to a congestion where agents do no longer move at all. I want to avoid a change of the crowd dynamics because this opens the question whether changes in the information dissemination process are caused by a change of the crowd dynamics or by a change of the mobile network.

The second parameter is the *transmission interval*. At the simulation time 100 s, one agent next to the closed gate (see Fig. 4.2) starts to send detour information repeatedly over broadcast. The *transmission interval* is the time in between two broadcast rounds. The repetitions increase the probability that information is successfully disseminated even if the communication locally or temporarily breaks due to interference. However, the smaller the transmission interval, the higher the risk for collisions and interference. Therefore, I choose a lower bound of 0.8 s. As upper bound I choose 2.8 s which means that the broadcast is re-started four times within 10 s: 0 s, 2.8 s, 5.6 s, 7.8 s. I argue four chances should be enough to disseminate information.

The third parameter is the *transmitter power* of the smart phone's antennas. In combination with other parameters such as the receiver sensitivity of the antennas, this parameter determines the range for communication. It serves as a control because this parameter does not have an effect: The scenario is sufficiently small so that agents are always in communication range (see Appendix C). I choose a parameter range [2.0 mW, 20.0 mW] which is typical for many types of smartphones.

Selection of quantities of interest

I am interested in six scalar quantities of interest:

- *TimeAll*: Dissemination time ($p = 0.95$) referring to all agents: How long does it take until 95 % of the agents are informed about the detour? Accuracy: time step size 0.4 s. *SI-Unit: s*

- *TimeArrArea*: Dissemination time ($p = 0.95$) referring to agents in the arrival area: How long does it take until 95% of the agents are informed in the arrival area? Accuracy: time step size 0.4 s. *SI-Unit: s*

- *NumAgents*: I use the number of agents as control (similar to the control parameter transmitter power; here the number of agents is a measurement quantity not a parameter). *SI-Unit: 1*

Please note that the *dissemination times* are always a multiple of the simulation step size of the *Vadere* simulator (0.4 s).

The information is successfully disseminated when at least $p = 95\%$ of the agents have been informed. A proportion of $p = 95\%$ ensures that the information degree is not underestimated due to short-term fluctuations caused by informed agents that reach the target or leave the reference area.

To check the plausibility of the simulation results, I compare the *dissemination time* with the statistics of the microscopic quantity of interest *packet lifetime*. The *packet lifetime* describes how long it takes that a packet is transmitted in the mobile network from a sender to a receiver.

- *MaxPackLifetime*: maximal packet lifetime. *SI-Unit: s*

- *MedPackLifetime*: median packet lifetime. *SI-Unit: s*

- *MinPackLifetime*: minimal packet lifetime. *SI-Unit: s*

I expect that the *maximal packet lifetime* and the *dissemination times* have the same magnitude. I further expect that the *maximal packet lifetime* is higher than the information dissemination time since the *dissemination time* refers to the $95\,\%$ of the agents who get informed first. I check both expectations when analyzing the results.

Methodology: forward propagation and sensitivity analysis using polynomial chaos expansions

I use polynomial chaos expansions for forward propagation and global sensitivity analysis. In total, I construct 6 polynomial chaos expansions: one for each quantity of interest, see Fig. 4.7. The same order is chosen for all polynomial chaos expansions. Therefore, the same sampling can be used for their construction. For the polynomial degree, I choose a quadratic approach (order $= 2$) to capture linear and quadratic behaviors of the quantities of interest in dependency of the uncertain parameters. To model plateaus or rapid changes, higher orders would be necessary. However, I cannot further increase the order since the number of samples grows exponentially and running one simulation can take several hours. To construct a polynomial chaos expansion of order 2, three grid points are required for each parameter. The grid point values can be found in Tab. 4.2. In total, this makes 27 samples or simulation runs.

Due to the long simulation duration of up to four hours, I do not carry out any repetitions, although random effects are present in my system. I argue that the influence of random effects is negligible because due to the *number of agents* in the system the crowd forms a coherent communication network. I expect that there are local changes in the communication due to stochasticity. For example in one case agent (1) communicates directly with agent (2) and agent (3). In another case, agent (1) communicates with agent (2) who passes the information on to agent (3). As long as the network is coherent, I claim that such local effects have a negligible effect on the macroscopic
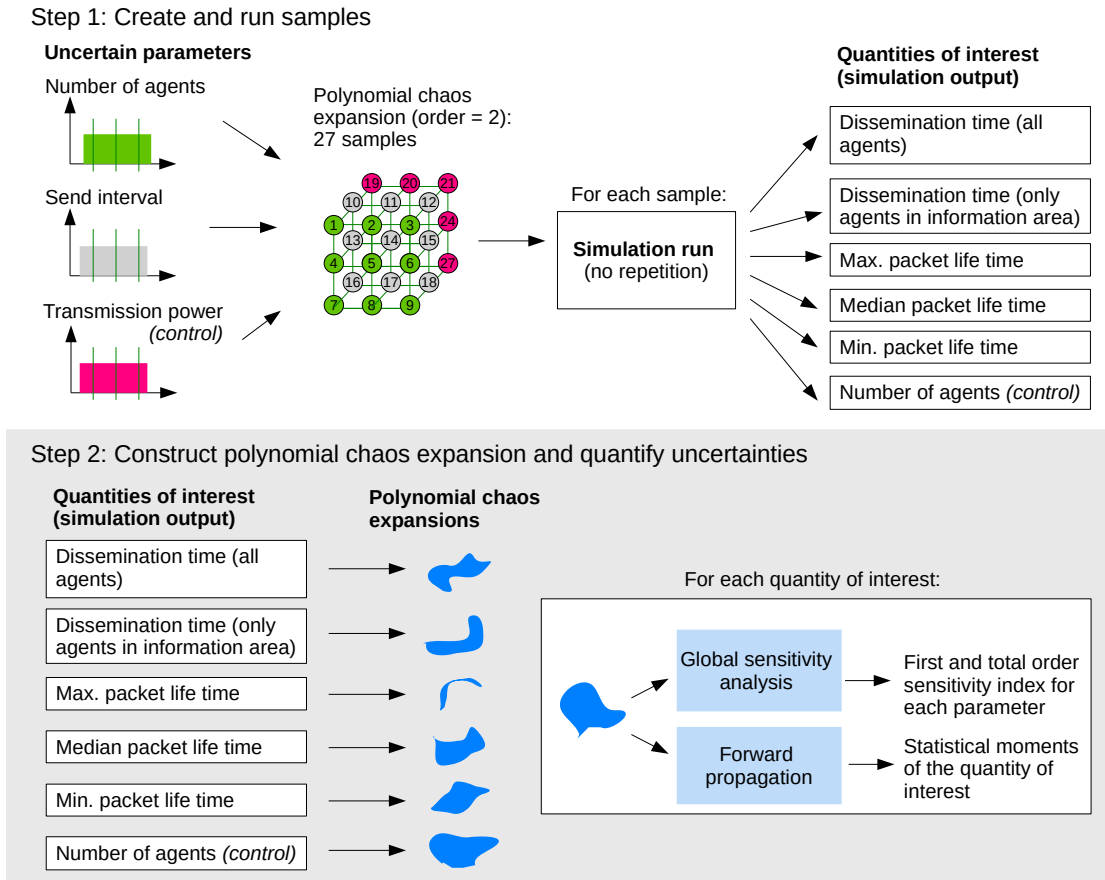
Figure 4.7.: Parameters and quantities of interest for forward propagation and global sensitivity analysis. I use polynomial chaos expansions for forward propagation and sensitivity analysis. For each of the six quantities of interest, a polynomial chaos expansion of order 2 is constructed. With three parameters, 27 samples must be executed. From the expansions the first and total order sensitivity indices of the parameters and the statistical moments of the quantities of interest are computed.

dissemination times or the statistical moments of the *packet lifetime*. In fact, a preliminary study with one sample demonstrated that the quantities of interest do not differ qualitatively for the three tested seeds. This supports my claim, without being a general proof.

The simulation study is carried out with the *SUQ-controller* module of *CrowNet*. To reduce the total simulation time, 8 simulations are run in parallel using the virtual machine that is described in Appendix B.

Verification of the polynomial chaos expansions and plausibility check

First, the correct construction of the polynomial chaos expansions is verified. The first
control hypothesis is that the parameter *transmitter power* has no influence. Tab. 4.3
shows that this is indeed the case. Both the first and the total sensitivity index are
zero, indicating that the parameter has no influence. The second control hypothesis is
that the quantity of interest *number of agents* depends only on the uncertain parameter
number of agents. This is indeed the case: Tab. 4.3 shows that the first and total index
of the parameter *number of agents* is 1.0. All other indices are zero (see line *NumAgents*
in Tab. 4.3), which means that the other parameters have no influence.

Second, the simulation results are checked for plausibility. Tab. 4.4 depicts the statistical
moments of the quantities of interest. One can observe that the expected values of the
median packet lifetime (5.046 s) and the *dissemination time* (5.0 s ... 5.4 s) are equal for
an accuracy of 0.4 s. Importantly, they have the same order of magnitude. Therefore,
the simulation results are plausible.

Parameter Quantity of interest	Transmission interval	Number of agents	Trans. power (control)
TimeAll (all)	0.9 (0.9)	0.1 (0.1)	0.0 (0.0)
TimeArrArea (arrival area)	0.7 (0.8)	0.2 (0.3)	0.0 (0.0)
NumAgents (control)	0.000 (0.000)	1.000 (1.000)	0.000 (0.000)
MaxPackLifetime	0.636 (0.670)	0.330 (0.364)	0.000 (0.000)
MedPackLifetime	0.999 (0.999)	0.001 (0.001)	0.000 (0.000)
MinPackLifetime	0.743 (0.748)	0.252 (0.257)	0.000 (0.000)

Table 4.3.: First and total order sensitivity indices for the quantities of interest. The
indices are estimated using polynomial chaos expansions. The total effect in-
dices are in brackets. As expected, the control parameter *transmitter power*
does not have any influence (indices: 0.0). The quantity of interest *NumA-
gents* serves as a control: It only depends on the uncertain parameter *number
of agents.*

Influence of the parameters *number of agents* and *transmission interval*

Next the influence of the two parameters *number of agents* and *transmission interval*
is analyzed. Tab 4.3 shows their sensitivity indices for each quantity of interest. One
observes that the first and the total sensitivity index (shown in brackets) are similar.
This indicates that there are negligible interaction effects (second order effects) between
the parameters. Thus, I use the first sensitivity index from now on.

Quantity of interest		Expected value	Standard deviation
Dissemination time in s	*TimeAll*	[5.0 ... 5.4[	1.2
	TimeArrArea	[3.2 ... 3.6[	0.8
Packet lifetime in s	*MaxPackLifetime*	7.548	0.730
	MedPackLifetime	5.046	0.545
	MinPackLifetime	4.973	0.701

Table 4.4.: Statistical moments for the quantities of interest. The statistical moments are estimates, derived from polynomial chaos expansions. As the *information degree* is only measured every 0.4 s in *Vadere*, the *dissemination time* can only be specified with an accuracy of 0.4 s. Accordingly, I use 1 decimal place only and provide ranges for the expected values and standard deviations for the *dissemination times*. Note that the range limits of the standard deviations differ in the second digit only under the assumption that the error of the standard deviation is 0.4 s.

The parameter *number of agents* has only little influence on the information dissemination: For the *median packet lifetime*, the sensitivity index is almost zero. This indicates no effect at all. For the *minimum* and *maximum packet lifetime*, the indices are low with 0.252 and 0.330 respectively. In fact the true values might be even lower because the indices also encounter variance caused by the varying sample size. The more agents, the more packet transmissions, and the more, accurately the bounds of the *packet lifetime* distribution are captured. For the evaluation of the dissemination process, I, therefore, look at *dissemination times* that are independent of the sample size.

The dissemination of information referring to all agents seems only slightly influenced (*TimeAll* index: 0.1) by the *number of agents*, see Tab. 4.3. For agents in the arrival area, the influence is higher but still low (*TimeArrArea* index: 0.2). The low influence can be verified visually in Fig. 4.8. For each *transmission interval*, the *dissemination time* is plotted over the *number of agents*. The data points correspond to the samples used for the polynomial chaos expansion. One can observe that the *dissemination time* hardly changes over the *number of agents* (left). For the arrival area (right), the *dissemination time* changes over the *number of agents* for certain *transmission intervals*. I attribute this to the fact that fewer agents are in the arrival area than in the entire topography. Therefore, the local transmission of packets has a stronger influence on the *dissemination time*.

The parameter *transmission interval* has a great influence on all quantities of interest. The index is between 0.636 and 0.999 except for the control, see the column 'Transmission interval' in Tab 4.3. An explanation is that the smaller the *transmission interval*, the more often information is provided and the faster the crowd is informed. The influence of the *transmission interval* can be also observed in Fig. 4.8. The larger the *transmission interval*, the larger the *information dissemination*. The parameter influence is more dominant for the entire population than for pedestrians within the arrival area: The corresponding indices are 0.9 (*TimeAll*) and 0.7 (*TimeArrArea*), see again Tab. 4.3.
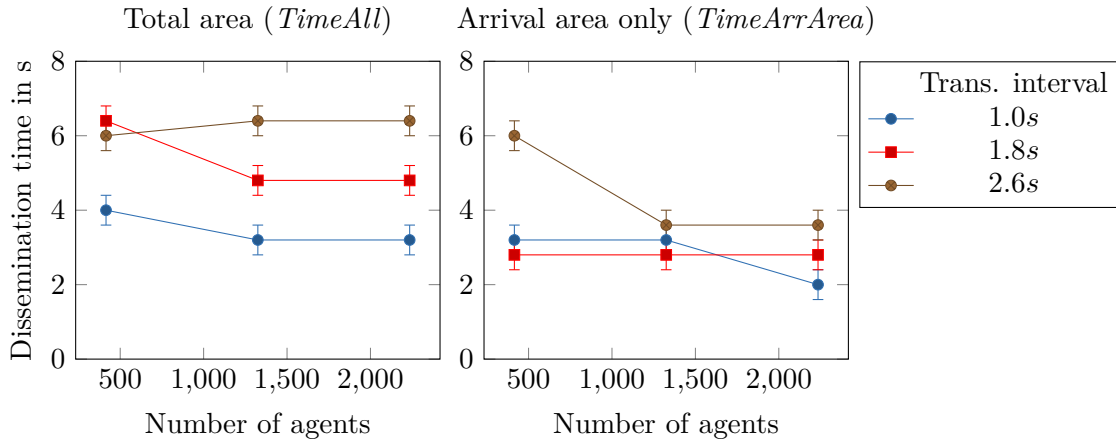
Figure 4.8.: *Dissemination time* over number of agents. Due to the time step size, the dissemination time has an accuracy of 0.4 s indicated by the error bars. The data points correspond to the sample points used to construct the polynomial chaos expansions. The dissemination times hardly change over the *number of agents*. The parameter *transmission interval* has indeed an effect (left).

To quantify the uncertainty of the *dissemination times*, I look at their statistical moments, see again Tab. 4.4. One can expect that the total crowd is informed within 5.0 s ... 5.4 s, while the population in the arrival area is informed in 3.2 s ... 3.6 s. The standard deviations are 1.2 s and 0.8 s respectively. Most importantly, none of the *dissemination times* is larger than 7 s even if the *number of agents* exceeds 2000, see again Fig 4.8.

Need of repetitive information provision

With an increasing number of pedestrians, the probabilities of interference and packet loss increase, see Fig. 4.9. However, interference seems to affect the information dissemination process only locally. Fig. 4.10 shows one sample where some agents have not been informed after 3.6 s. These agents are all located on the left-hand side near a dense crowd where the risk of interference is high. Despite inferences, agents are still informed within 6.4 s, see Fig. 4.10. I attribute this to the fact that information is provided repeatedly. If the information dissemination fails locally, the information is received in the next broadcast round. Since agents informed in the first round do not forward information, the risk of interference is reduced for the next rounds. This also explains the high influence of the parameter *transmission interval*.
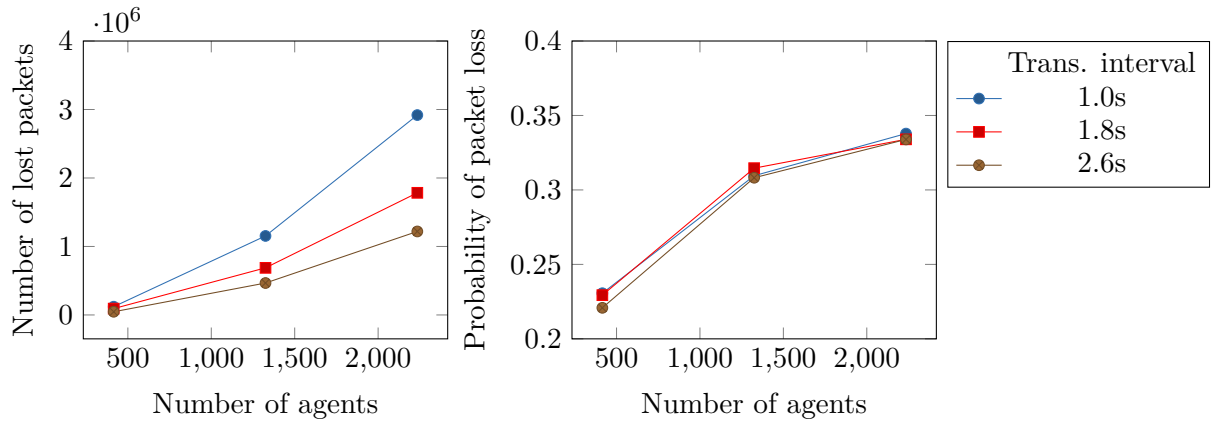
Figure 4.9.: Packet loss caused by interference. The number of lost packets increases over the *number of agents* (left). The lower the transmission interval, the more often the dissemination process is re-started and the higher the loss is. The probability for interference, estimated by the the number of lost packets divided by the number of total packets, also increases over the *number of agents* (right). The transmission interval has no effect on the probability.

Delays in the information dissemination process

As expected, the sub-population of agents in the arrival area is informed faster than the total population. Fig. 4.11 depicts the information degree for both populations. One can observe that the orange lines (sub-population) cross the 95 % threshold earlier than the blue lines (total population). This is plausible, since less agents need to be informed. Interestingly, the information dissemination process shows some delays, see the plateaus of the graphs in Fig. 4.11. These delays occur because the information is passed via a chain of agents around obstacles. If the communication chain breaks, the transmission is delayed for at least one *transmission interval*. Note that, for the arrival area, the decrease of the information degree in Fig. 4.11 (top left) is caused by informed agents leaving the arrival area.

Summary and findings from the uncertainty quantification analysis

The second sub-study indicates that information can be disseminated in a mobile crowd through direct communication within a few seconds. A global sensitivity analysis showed that the crowd size does not greatly affect the dissemination time, that is, the time it takes to distribute detour information in a crowd. This is fortunate since in a real life application one cannot control the crowd size. The transmission interval, that is the time between two repetitions when detour information is sent, had a big influence on the information dissemination process. The more often information was provided, the faster the entire crowd was informed.

Simulation time: 100.0 s
(Start to disseminate information)

Simulation time: 103.6s
(95 % of agents informed
in arrival area)

Simulation time: 106.4s
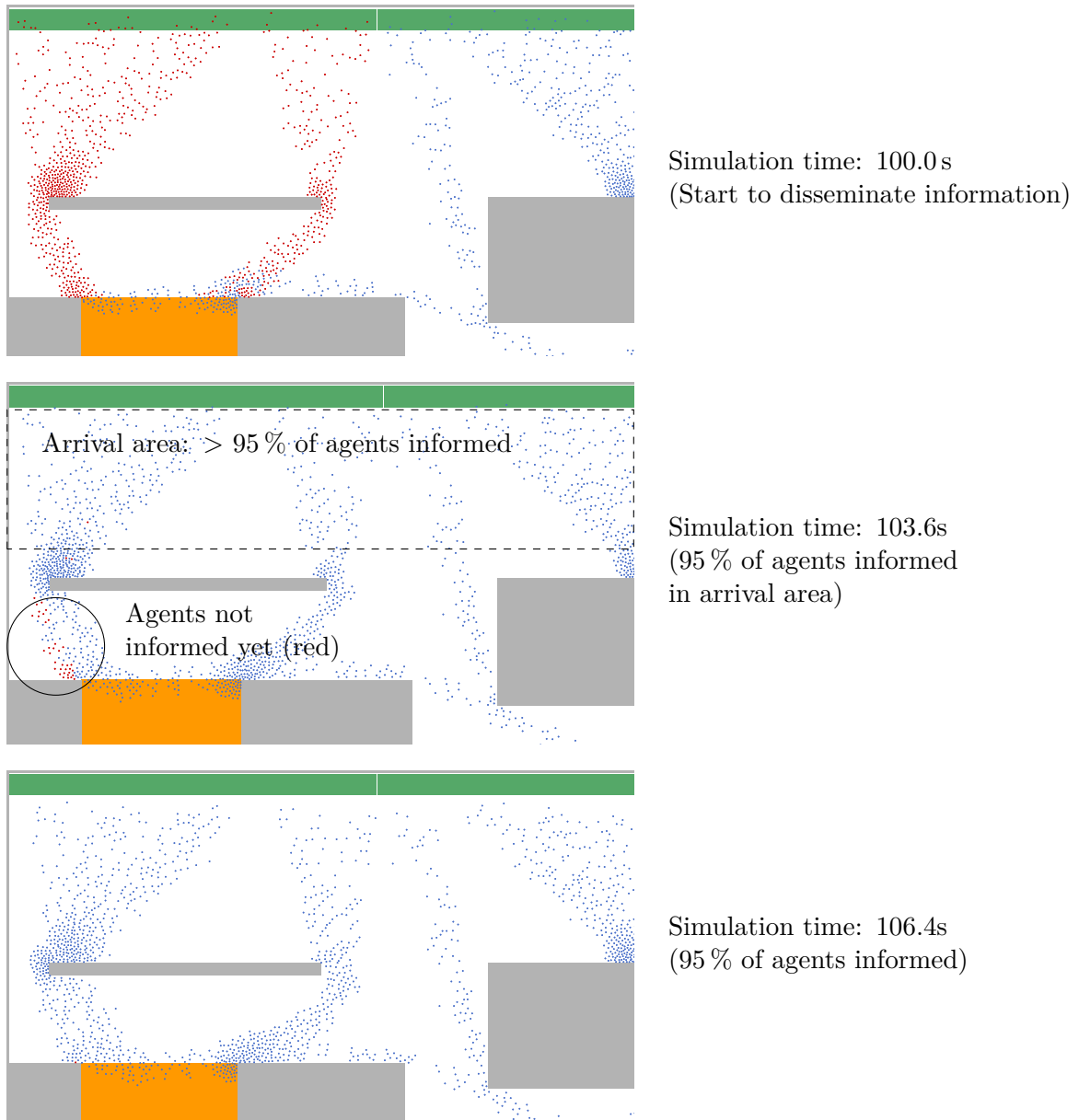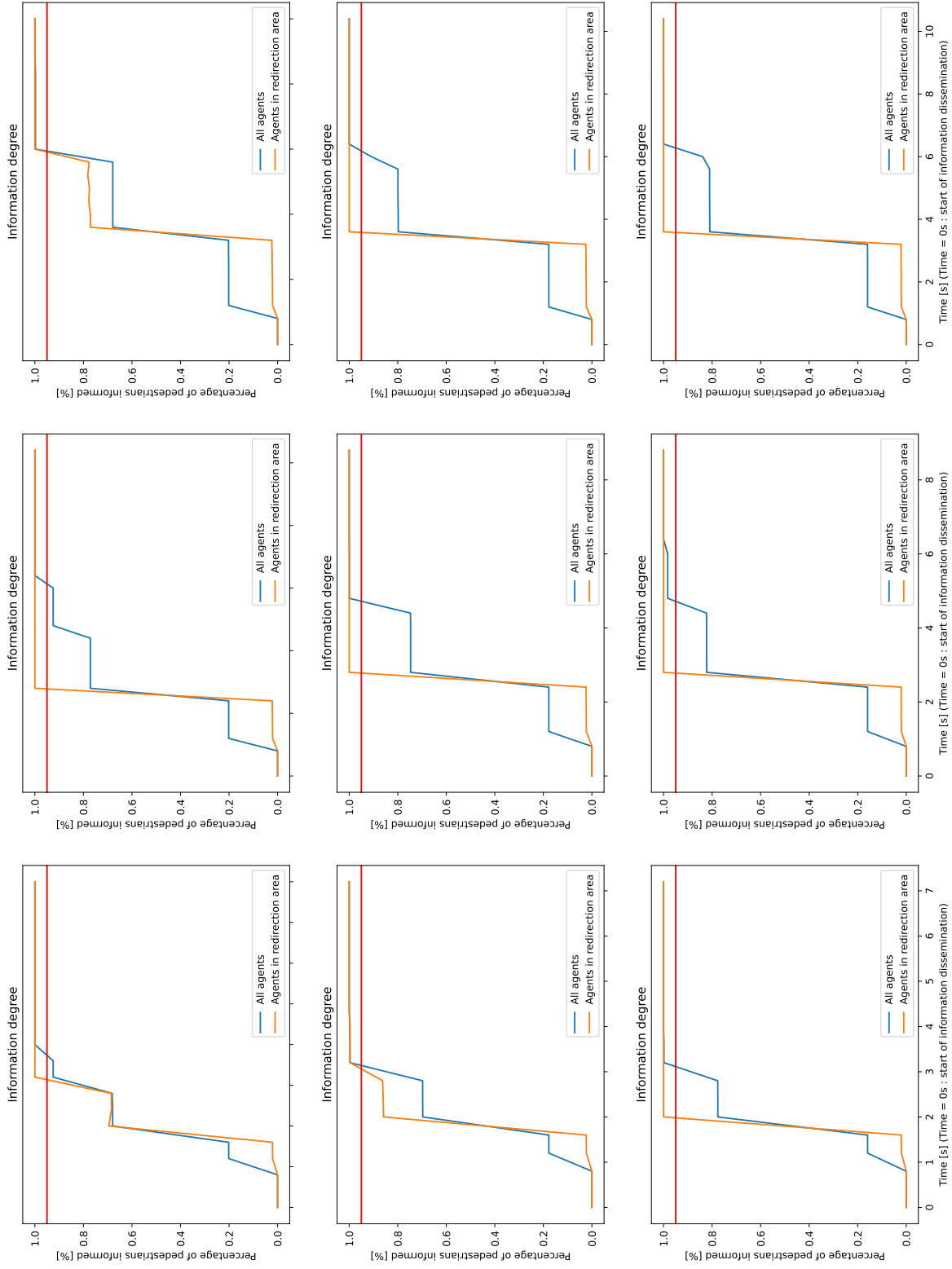(95 % of agents informed)

Figure 4.10.: Uncertainty quantification study: Simulation snapshots for the design with 2235 agents, a transmitter power of 19.97 mW and a transmission interval of 2.6 s. The information dissemination starts at 100 s. Red agents are unaware about the closure of the gate. After 3.6 s, the agents in the upper part have been informed. After 106.4 s 95 % of the total population has been informed.

Figure 4.11.: Information degree over time for a fixed radio transmitter power (p = 11 mW). The information dissemination time is the time that elapses until 95 % of the agents are informed in the arrival area (orange curve) or the total area (blue curve). The value p = 95 % (red line) ensures that temporal fluctuations, caused by informed agents leaving one of the reference areas, do not have an effect.

4.1.6. Summary, limitations, lessons learned

Summary of the two investigations

In this section I investigated my first research sub-question, that is, how reliably route recommendations are disseminated through direct communication technology in a mobile crowd. I proposed and applied a two-step procedure to investigate the information dissemination systematically when shadowing is present. First, one examines how shadowing depends on the number of agents in the scenario. More agents mean a closer distance between nodes in the mobile network. Second, one assesses how quickly information is transmitted when there are sufficiently many crowd members in the system. This procedure allows to separate the effect of shadowing and interference and, thus, helps to better understand the behavior of the socio-technical system composed of the crowd and the mobile network. To answer my research question, I designed a scenario that reflects worst-case conditions for the direct communication: A person sending the detour information was shadowed by walls and only pedestrians in line-of-sight received detour information through direct communication. To evaluate the reliability of the information dissemination process, I introduced a criterion that demands that the entire crowd must be informed within a certain time. In the first part of the study I varied the number of agents. I found that at least 110 crowd members are necessary to prevent shadowing in the scenario. In the second part, the uncertainty of the information dissemination process was quantified using forward propagation based on polynomial chaos expansions. To quantify the influence of parameters, global sensitivity analysis was applied, again using polynomial chaos expansions. I found that route recommendations were disseminated in a crowd within a few seconds. The number of crowd members had a low effect on the information dissemination when the crowd was dense enough. The transmission interval, that is the time in between two information broadcasts, had a large effect on the information dissemination. The more often information was provided, the faster the entire crowd was informed. In my scenario the fastest information dissemination was achieved with a transmission interval of one second.

In both studies I assumed that the communication breaks whenever the line-of-sight is broken. In reality, the information provision might be even better thanks to the permeability of materials and to reflections. In short the answer to the first research sub-question is:

> **How reliably are route recommendations disseminated using direct communication in a crowd? (RQ-1)**
>
> Using direct communication technology, route recommendations can be reliably disseminated in a mobile crowd within a few seconds, provided there are enough crowd members to counteract shadowing.

Limitations

I would like to point out some limitations of the study, some of them which I will address in my following investigations.

I assumed that all pedestrian immediately and always follow the route recommendation. In reality, pedestrians might not be aware of the mobile message or for some reason my not follow instructions. Therefore, I will assess the effect of compliance in Section 4.2.

The topography of the scenario was sufficiently small. Therefore, pedestrian were always in communication range. If the topography was larger, information dissemination may fail.

In the study, interference caused local packet loss which did not affect the overall information dissemination process. However, if the crowd size was denser, information dissemination might fail.

In the study I tested direct communication based on the WLAN 802.11p standard. I did not look at more advanced technologies, such as 802.11bd communication or cellular sidelink communication, because I wanted to assess the worst case. For example, I expect that the risk of interference is smaller for the sidelink communication in the controlled mode. Therefore, I am optimistic that other technologies are also suitable: I will test LTE sidelink communication in Section 4.4.

I excluded additional network traffic to keep the numbers of factors low in the study. As a previous study shows, network load has indeed an effect on the information dissemination, see Appendix D. I argue that a network load through direct communication is unlikely to occur in real life applications because people use mobile applications that request individual data. There is no need to share data between devices. Hence people use conventional cellular communication. Due to other frequency bands, the direct communication would not be affected.

I only analyzed the effect of three uncertain parameters that I identified as main model parameters. There is always the possibility that I may have overlooked an important uncertain parameter.

Lessons learned for my further investigations

The study showed that the crowd size slightly affects the information dissemination given that the crowd is dense enough and well-distributed. Route recommendation algorithms are in particular interest for dense crowds. From this, I conclude that such an algorithm does not need to take the crowd size into account.

In the study, the transmission interval was varied. Although it had an effect on the information dissemination, the crowd was always informed within a few seconds. I conclude

that an interval on a second scale is sufficient when information needs to be provided repeatedly due to an information loss.

In the study, the information dissemination took longer than the size of one transmission interval. Therefore, a route recommendation algorithm that generates recommendations dynamically must not change the information content during this period of time. Otherwise, contradictory information could be disseminated. I conclude that it is essential to check whether this functional requirement is fulfilled. I will present a procedure for this in Section 4.4.

4.2. Selection of a route recommendation algorithm

In this section I investigate my second research sub-question: Which algorithms are suitable for generating route recommendations for crowds, given not all people follow instructions? In the state of the art, I discussed why former algorithms were tested under unrealistic conditions: It was assumed that crowds behave like fluids. However, recommending a route is not the same as opening a valve to regulate fluid flows. People may or may not follow instructions that are suggested to them by signs, loudspeakers, mobile applications or staff. This is true even if they perceive and understand the suggestions. For this reason, it is necessary to consider the influence of compliance when investigating route recommendation algorithms. In view of the models' and, indeed, the true systems' many uncertainties, I argue that simple algorithms have a better chance to be implemented in a real setup than more intricate algorithms. I also expect algorithms to be more robust if they do not depend on precise measurements to achieve target values in some control loop. This is why I propose and test three heuristic algorithms for unknown compliance rates. The study presented in this section builds on my publication *Guiding crowds when facing limited compliance: Simulating strategies* [306] in which I examined two out of three route recommendation algorithms from this study. The careful reader will therefore find textual overlaps, as the studies have a similar structure. Since this study extends the findings from my previous study, it is also of interest for readers who are already familiar with the publication.

4.2.1. Research scenario

I propose a scenario inspired by a real-life use case at the metro station Münchner Freiheit in Munich, Germany. For the purpose of this study, it is sufficient to know that there are three routes of different lengths at Münchner Freiheit station to get from the bus or tram to the trains. The short route is usually more occupied than the two alternatives. This leads to congestion especially before football matches when fans change trains. The station site extends over two floors, which are connected through escalators, a ramp and stairs. To understand the effect of compliance and the difference between algorithms, effects caused by local topographical features such as twists and

turns, stairs or surface conditions, are excluded. Such local effects would introduce additional factors, which might prevent me from clearly assigning an effect to a factor. I, therefore, use an abstracted model of the topography.

The topography is depicted in Fig. 4.12. The outer dimensions of the area are 150 m × 25 m. The corridor widths and the ratio of the route lengths are similar to the real use case. The corridor width is w = 2.5 m for all corridors. Every two seconds, eight agents are spawned in the source on the left. The agents try to reach the (orange) targets placed at the ends of each corridor on the right. Without guidance, all agents take the shortest route, that is, the corridor on top. Such a behavior can be observed for football fans in the real world scenario. Congestion occurs along and in front of the short route.

To relieve the congestion, a crowd guidance system is implemented. The system provides route recommendation for agents within the information area. Agents follow a route recommendation with a probability c, that is, the compliance rate.
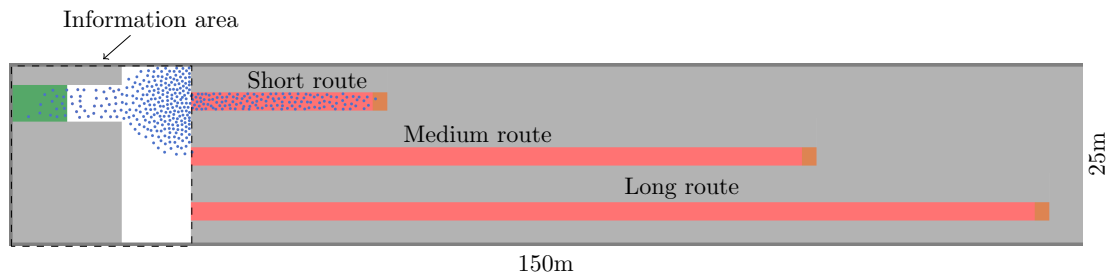
Figure 4.12.: Simplified topography. Agents are spawned in the source on the left (green) and walk to the targets (orange) on the right end of each corridor. Without guidance, all agents take the short route which causes congestion along and in front of the short route. In the setting with guidance, agents receive a route recommendation in the information area. Only a proportion of the crowd follows the route recommendation.

4.2.2. Setup of the simulation model and implementation

The simulation model of the crowd guidance system is composed of several component models, see Tab. 4.5. The validated Optimal Steps Model is used to model the locomotion behavior of the crowd. For the route recommendation algorithm and the route choice behavior, I propose novel models that are presented in the following.

Route recommendation algorithms

The route recommendation algorithm dynamically generates route recommendation at a certain time interval. Agents perceive these route recommendations only within the

Model	Sub-model	Model (simulator)
Crowd	Locomotion model	Optimal Steps Model (Vadere)
	Perception model	SimplePerceptionModel (Vadere)
	Cognition model	ProbabilisticCognitionModel (Vadere)
Controller	Time stepping alg.	FixedTimeStepper (flowcontrol)
	Route recommendation alg.	AlternateTargetAlgorithm (flowcontrol)
		OR
		MinimalDensityAlgorithm (flowcontrol)
		OR
		AvoidCongestionAlgorithm (flowcontrol)
Network	None	Assumption: information arrives immediately

Table 4.5.: Composed model for investigating guiding strategies.

information area that is composed of the green area of the source and the white area between the source and corridors, see Fig. 4.12.

I choose a time interval of $\Delta t =$ 2 s. The smaller the time interval, the more the algorithm is able to react to changes in the dynamics. In an ideal world, where agents would react immediately and reliably, one would expect a smaller Δt to improve the distribution of agents. However, through the delay caused by the provision of information and the fact, that it might be confusing for pedestrians when the information content changes rapidly, I limit the time interval to 2 s.

As I have outlined in the state of the art, several guidance algorithms have been proposed (see Section 2.1.5). However, these are neither transferable to arbitrary use cases, nor have they been tested under realistic conditions. Therefore, I propose and test three route recommendation algorithms that are based on heuristics:

- The *alternate route algorithm* sequentially alternates route recommendations every $\Delta t = 2$ s for a topography with n routes that all lead to the same target. After route n, the algorithm re-starts with the first corridor. Thus, the order is fixed. In our scenario, with $n = 3$, the algorithm provides the following recommendations: $t = 0$ s: short corridor, $t = 2$ s: medium corridor, $t = 4$ s: long corridor, $t = 6$ s: short corridor, ...

- The *minimal density algorithm* measures the density every $\Delta t = 2$ s and recommends the route where the density is minimal.

- The *simple density algorithm* recommends the long route only, when the density along the short route is higher than the density along the long route. The densities along other routes are not considered. Thus, the medium route (Fig. 4.12) is never recommended in the research scenario.

I implement the three algorithms in the *flowcontrol* simulator. All algorithms inherit from the abstract class `RouteRecommendationAlgorithm`, see again Fig. 3.21.

One could come up with the idea of addressing individual agents to different directions to dispense with the timing issue. Again, this does not work with humans because groups would have to split up, which they will not. Also, people are aware of others and might be confused if suggestions differ. The density-based algorithms do not suffer from that drawback, since one can assume the density not to fluctuate much, given that the measurement areas are sufficiently large. Thus, the algorithm can be implemented without a time schedule. This would simplify the realization of the algorithm in practice. Note that the two density-based heuristics can be also represented as a system of interdependent On-Off-controllers. I mention it because this connection might be interesting for future research.

Information provision

It is assumed that information is immediately available. One might argue that there are indeed delays in a real system where information is disseminated over the mobile network. I argue that these delays would be similar for the three algorithms. Therefore, they have a negligible effect when comparing them. Importantly, I can save computational effort because the mobile network simulation is not needed. This allows me to use a fine discretization of the compliance rate, which is a main factor of this study.

The availability of the route recommendation is restricted to the information area in front of the scenario's diverging corridors, see again Fig. 4.12. In a real-life application this could be the place where football fans get off the bus at the surface level of the real metro station.

Crowd model and compliance behavior

The crowd model comprises an operational and a tactical layer. To model the locomotion behavior, the Optimal Steps Model is used. It is implemented in the *Vadere* simulator.

At the tactical layer, I propose a novel model that describes how people react to route recommendations. The model builds on the perception and cognition layer structure that I presented in the state of the art. The agents receive route recommendations as stimuli, which are dynamically computed and provided by the algorithm. Only agents in the information zone perceive stimuli. Agents that receive multiple recommendations respond only to the first recommendation. To model the perception, I use the *SimplePerceptionModel* that is a perception model implemented in the *Vadere* simulator. The model extracts the most important stimulus. In the scenario, the route recommendation is the only stimulus. Therefore, the model only forwards the stimulus to the cognition layer. For the cognition process, I propose a novel model that reflects the decision of the population, that is, link choice. The idea is that agents decide with a certain probability whether to follow a route recommendation or stick to their initial route choice. This behavior is modeled with a probability distribution which has a probability

parameter that corresponds to the compliance rate, see Fig. 4.13. The recommended route has the compliant rate c as the probability value assigned. The short route has $1 - c$ assigned, so that non-compliant agents continue to their default destination. Non-recommended alternative routes have a probability of 0 assigned. A compliance of $0\,\%$ means that an agent would reject the route recommendation and continues to take the short route. $100\,\%$ compliance means that an agent follows the route recommendation, regardless of which route is recommended. Inter-dependencies regarding compliance or herding phenomena are not captured by the model. The model has two advantages over conventional route choice models. Firstly, the computational effort is lower, as no state variables such as visibility, distances etc. need to be determined. Secondly, the model can be applied to arbitrary scenarios.
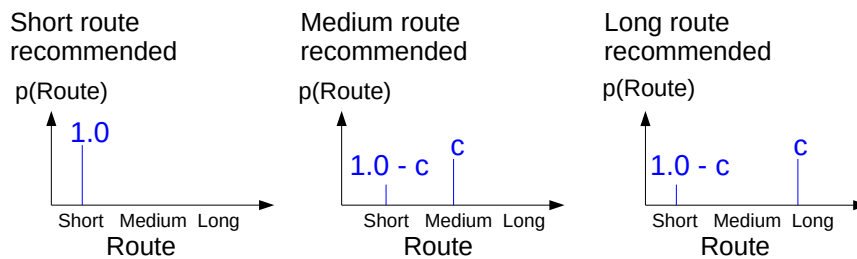
Figure 4.13.: Discrete probability distributions for different route recommendations. The default behavior is to take the short route which is why the probability assigned is 1.0 (left). The discrete probability distribution is controlled through the parameter compliance rate c (middle, right).

The novel cognition model is implemented as *ProbabilisticCognitionModel* class in *Vadere*. At each time step, the update method is called which checks whether an agent has already received the route recommendation. If the information is new, the agents makes a route choice which is realized by choosing one of the intermediate targets placed at each route. The target is randomly drawn from a discrete probability distribution, see Listing 4.1.

```
public LinkedList<Integer> getNewTarget(...) {
    ...
    EnumeratedIntegerDistribution dist = new EnumeratedIntegerDistribution
     (..., targetIds, probabilities);
    Integer newTargetId = dist.sample();
    LinkedList<Integer> newTarget = new LinkedList<>();
    return newTarget.add(newTargetId);
}
```

Listing 4.1: getNewTarget() method of the novel ProbabilisticCognitionModel. Route choice is realized by drawing from a random distribution. The discrete distributions in dependency of the compliance rate are depicted in Fig. 4.13.

4.2.3. Design of the simulation study

Parameter selection and quantities of interest

In the absence of more information, I assume the parameter *compliance rate* to be uniformly distributed. The lower bound is 0 (agents always ignore instructions), and the upper bound is 1 (agents always comply with route recommendations). A compliance rate $c = 0$ is similar to a setting without guidance. For each route recommendation algorithm the compliance rate is varied from 0% to 100%. The design of the simulation study is depicted in Fig. 4.14.
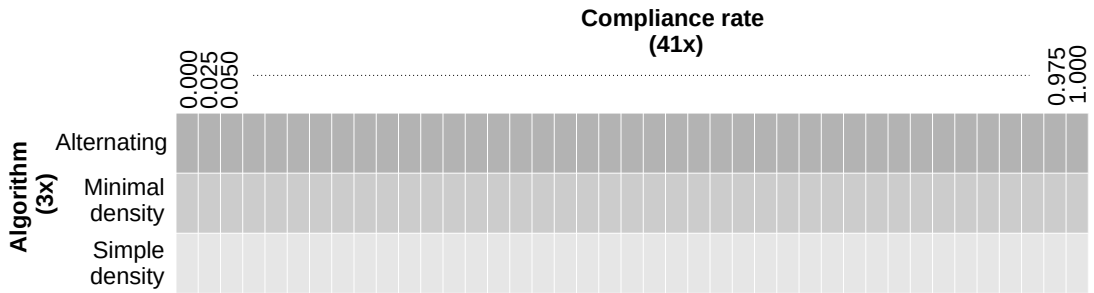
Figure 4.14.: Design of the simulation study. For each route recommendation algorithm, the compliance rate is varied between 0 and 1. 41 discrete compliance rates are tested.

To compare the performance of the algorithms the densities are measured in each of the three corridors, see Fig. 4.12. To detect congestion, velocities are measured. To understand the effect on the travel comfort, the individual travel time, that is the time an agent needs to walk from the source to a target, is recorded. See Tab. 4.6 for an overview of quantities of interest. The quantities of interest are measured every $0.4\,\mathrm{s}$, that is, the default time step size of the *Vadere* simulator. For estimating the density, the classical method is used where the density is the number of agents divided by the measurement area: $d = N/A$. As velocity, the spatial mean velocity is used which is defined as the average of the pedestrians' individual velocities within a measurement area.

Quantities of interest	Unit
Individual travel time	s
Density (for each corridor)	ped/m^2
Velocity (for each corridor)	m/s

Table 4.6.: Quantities of interest of the route recommendation algorithm study. The densities and velocities are measured in the three measurement areas (red areas in Fig. 4.12) every $0.4\,\mathrm{s}$. For each agent, the individual travel time is recorded.

The simulation study is carried out using the *SUQ-controller* module of *CrowNet*. To reduce the total simulation time, 10 simulations are run in parallel. The study is run on a server that is described in Appendix B.

Expected minimum compliance rate from flow equations

I estimate the minimum compliance rate that is necessary to resolve congestion to check the plausibility of the simulation results. For this purpose I use the jamming criterion from Section 2.1.2. As a reminder, the criterion says that there is congestion when the inflow exceeds the capacity of a bottleneck. The capacity J_C is the maximal pedestrian flow through the bottleneck. If J_C is the upper limit, the lower limit l of the percentage of people that need to be redirected, is:

$$l = \frac{J_{in} - J_C}{J_{in}} \tag{4.1}$$

The inflow is $J_{in} = 1.60\,\mathrm{ped}/(\mathrm{ms})$ in the scenario. As capacity value I use $J_C = 1.30\,\mathrm{ped}/(\mathrm{ms})$. This value is in line with handbooks [50, 51] and also with the fundamental diagram that I derived from the simulation of the unguided setting, see Appendix E.

The alternating algorithm recommends the shortest corridor to $1/3$ of the people. Thus, a proportion of $q = 2/3$ of the people are directed away from it. With a compliance rate c, the flow in the short corridor is:

$$J_{short} = J_{in}(1 - cq(c)) \tag{4.2}$$

To avoid a jam, it is required that $J_{short} \leq J_C$. Thus, the minimum compliance rate is

$$c = \frac{1}{q(c)}\left(1 - \frac{J_C}{J_{in}}\right) \tag{4.3}$$

Plugging $q = 2/3$ into Eq. 4.3 yields a compliance rate of at least 28% to avoid congestion. For two density-based route recommendation algorithms, $q(c)$ depends on the flow situation. Therefore, one cannot estimate a minimum compliance rate in advance.

4.2.4. Effect of guiding strategies

Congested routes cause high travel times

I first look at the case without guidance where all agents take the short route, see Fig. 4.15. For the performance evaluation, only the steady state of the system is considered, which is reached when densities and velocities stagnate. A steady-state flow

has been reached after about 250 s, see Fig. 4.16. Measurements taken before 250 s are excluded from the following analysis.
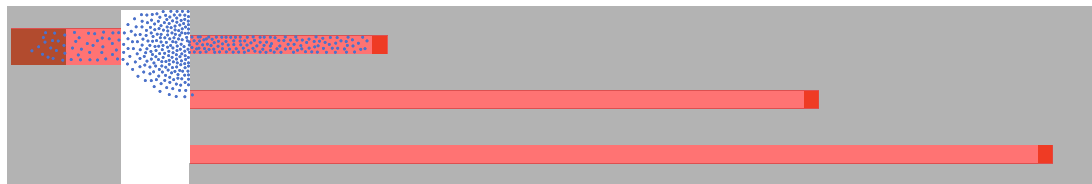

Figure 4.15.: Setting without guidance. All agents take the short route (simulation time $t = 250$ s). Congestion occurs because the inflow is larger than the capacity of the short corridor.
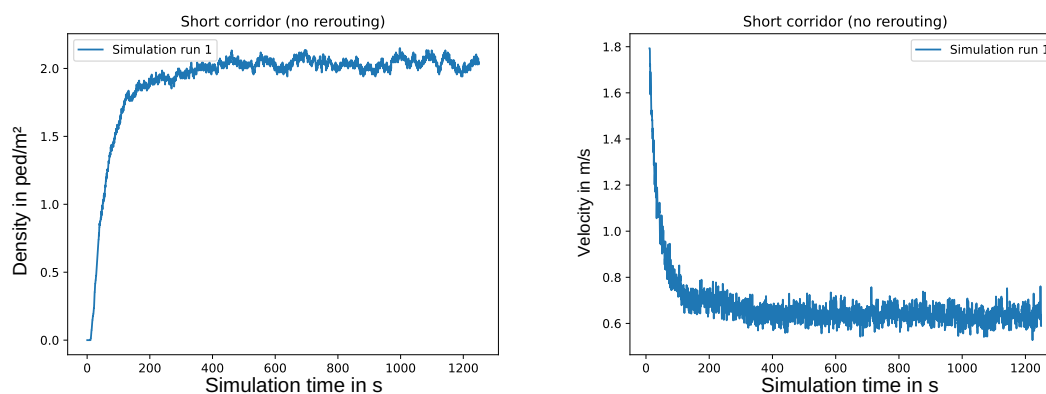

Figure 4.16.: Velocity and density in the short corridor over the simulation time in the setting without guidance. A steady-state flow is reached at approximately 250 s. Densities and velocities before 250 s are not included in the evaluations.

Tab. 4.7 lists the measured values on the short route: For the density and the velocity, the median and the mean values are identical. However, the mean travel time is considerably lower than the median travel time. Since this indicates that the travel time does not follow a normal distribution the median travel time is used for further evaluations.

The short route is approximately 50 m long. With a mean free-flow velocity of 1.34 m/s, the default parameter of the locomotion model, one would expect an average travel time of around 37 s, if agents walked freely. Instead, one observes congestion in the short corridor at a mean density of 2.0 ped/m² and with a walking speed reduced to 0.65 m/s on average. The density value of 2.0 ped/m² corresponds to E as a level of service according to Fruin. The movement temporally stops and no passing maneuvers are possible. Moreover, the median travel time of 178 s is almost five times longer than expected in a free flow regime, see Tab. 4.7.

	sample size	mean	std	min	25%	50%	75%	max
Density ped/m^2	2500	2.03	0.05	1.85	2.00	2.03	2.06	2.15
Velocity m/s	2500	0.64	0.03	0.53	0.62	0.64	0.66	0.76
Travel time [s]	2783	191.32	80.76	66.80	142.80	176.80	210.00	843.60

Table 4.7.: Statistics of the short corridor for the setting without guidance. Mean values, standard deviation, median, and further statistical quantities for the density and the velocity are computed using 2500 samples. The median and the mean value are similar for the density and the velocity. 2783 individual travel are used to compute the statistics of the travel time. The mean travel time is considerably lower than the median travel time, indicating that it is not normally distributed.

Any route recommendation algorithm reduces the travel time

Without guidance there is congestion resulting in high travel times. Fig. 4.17 shows the 25th, 50th and 75th percentiles of the travel time in dependency of the compliance rates. The black lines refer to the setting without guidance ($c = 0$) and represent an upper limit. The graphs are always below the black line, thus, any algorithm reduces the travel times in the scenario.

The minimum travel time is not achieved at the highest compliance rates, but at intermediate compliance rates. In Fig. 4.17 (middle), one can observe that the median travel time is around $100\,\text{s}$ for the *minimal density algorithm* at full compliance ($c = 1.0$). The minimal median travel time ($60\,\text{s}$) is achieved when the compliance rate is $c \in [0.2, 0.5]$ for the *minimal density algorithm* and $c \in [0.3, 0.8]$ for the *alternating algorithm*. For the *simple density algorithm*, the median values become minimum for $c \in [0.2, 1.0]$. The reason for this behavior is that full compliance evens out densities, but it does not optimize travel times. Imagine a compliance rate of $c = 0.3$. In this case, at least $70\,\%$ ($70\% = 1 - c$) of the agents take the short corridor. Thanks to sufficiently many other agents taking longer routes, there may be elevated density on the short path, but no congestion. This means short travel times for most. On the other hand, if 100% of the agents reject the route recommendation, the setting is equivalent to the scenario without guidance where the travel time peaks, see Fig. 4.17. This explains the rapid fall of the 25th, 50th and 75th percentiles of the travel time when $c \in [0.0, 0.2]$.

As soon as congestion is resolved, agents who take a longer route will reach the destination later. In the scenario, more than $75\,\%$ of the agents still reach their target faster than in the default setting with congestion, see Fig 4.17 (right): the 75th percentile is always below the black line. Interestingly, the 75th percentiles differ strongly for the three algorithms if the compliance rate is $c > 0.2$. The 75th percentiles of the density-based algorithms are always higher than the 75th percentiles of the alternating algorithm, see again Fig 4.17 (right). This is caused by the fact that the two density-based algorithms

recommend the long route more often than the *alternating algorithm*. Therefore, more agents have longer walking paths.
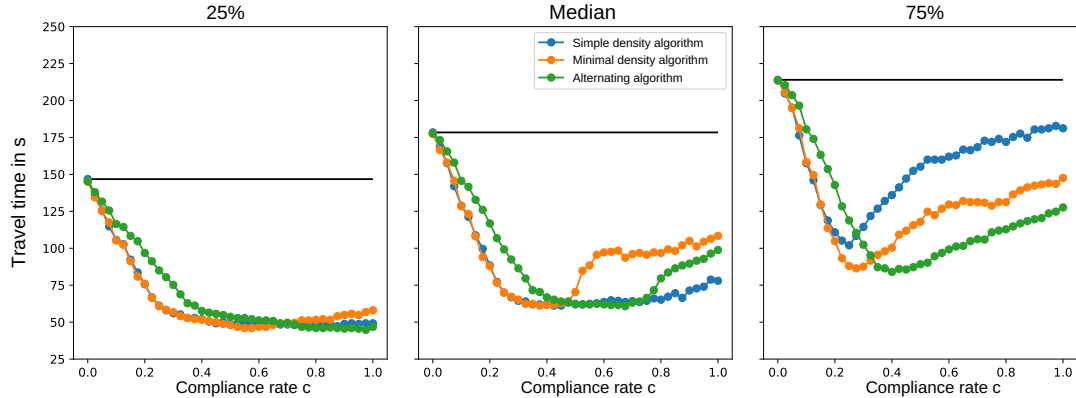

Figure 4.17.: 25th, 50th and 75th percentiles of the travel time in dependency on the compliance rate. The percentiles show at which compliance rate a certain percent of all agents experience travel times below the given value. The black line represents the setting without guidance, that is, a compliance rate of $c = 0$. The orange line represents the *minimal density algorithm*, the blue line the *simple density algorithm* and the green line the *alternating algorithm*. The three algorithms reduce the travel times for any compliance rate $c > 0$: The percentiles are always below the black line (without guidance).

A small proportion of compliant people suffice to avoid jamming

Whether or not an algorithms avoids congestion depends on the compliance rate. If no one listens to the route recommendation ($c = 0$), the situation is just as if there were no guidance. Based on the jamming criterion calculation, I estimated that congestion resolves for the *alternating algorithm* if the compliance rate is $c > 0.28$. The upper third of Fig. 4.18 depicts the density over the compliance rate in the short corridor. One can observe that the density increases when the compliance rate decreases.

With compliance rates below about $c = 0.3$ the density stagnates for the alternating algorithm. This is close to the estimate of $c = 0.28$. Note that the jamming criterion only evaluates congestion within the bottleneck. It does not consider agents of the bottleneck: One can observe this effect in Fig. 4.19. However, the number of waiting agents is much lower compared to the setting without guidance.

For the two density-based algorithms, one can observe that the densities stagnates when the compliance rate is $c \approx 0.2$, see Fig. 4.18 (top left and top center).
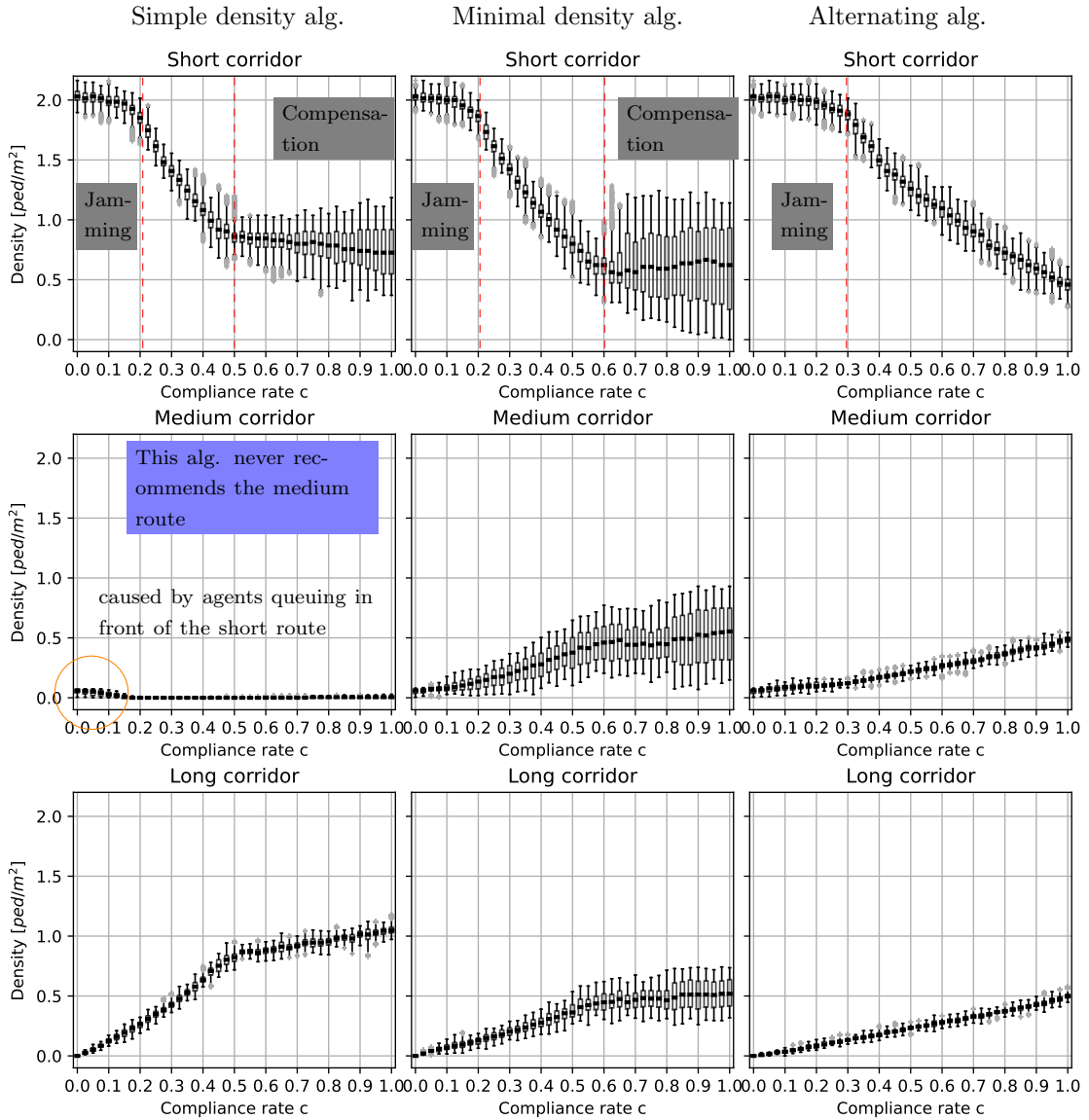
Figure 4.18.: Boxplots of the densities per route. The boxes represent the 25th and 75th percentiles. The thick black line in a box indicates the median value. Whiskers do not extend up and down from the box more than 1.5 times the interquartile range (75th percentile - 25th percentile). Values outside the whiskers are considered as outliers (gray dots). Different states can be observed: If the compliance rate is small, jamming occurs. With better compliance, the density decreases. Both density-based algorithms can compensate the lack of compliance: the density hardly changes for $c > 0.5$ and $c > 0.6$ respectively (top: left and center). The *alternating algorithm* cannot fully compensate a lack of compliance (top: right).

Compliance rate c=0.0:
no effect of the algorithm

Compliance rate

c=0.3

c=0.4

c=0.5

Simple density alg.   Minimal density alg.   Alternating alg.
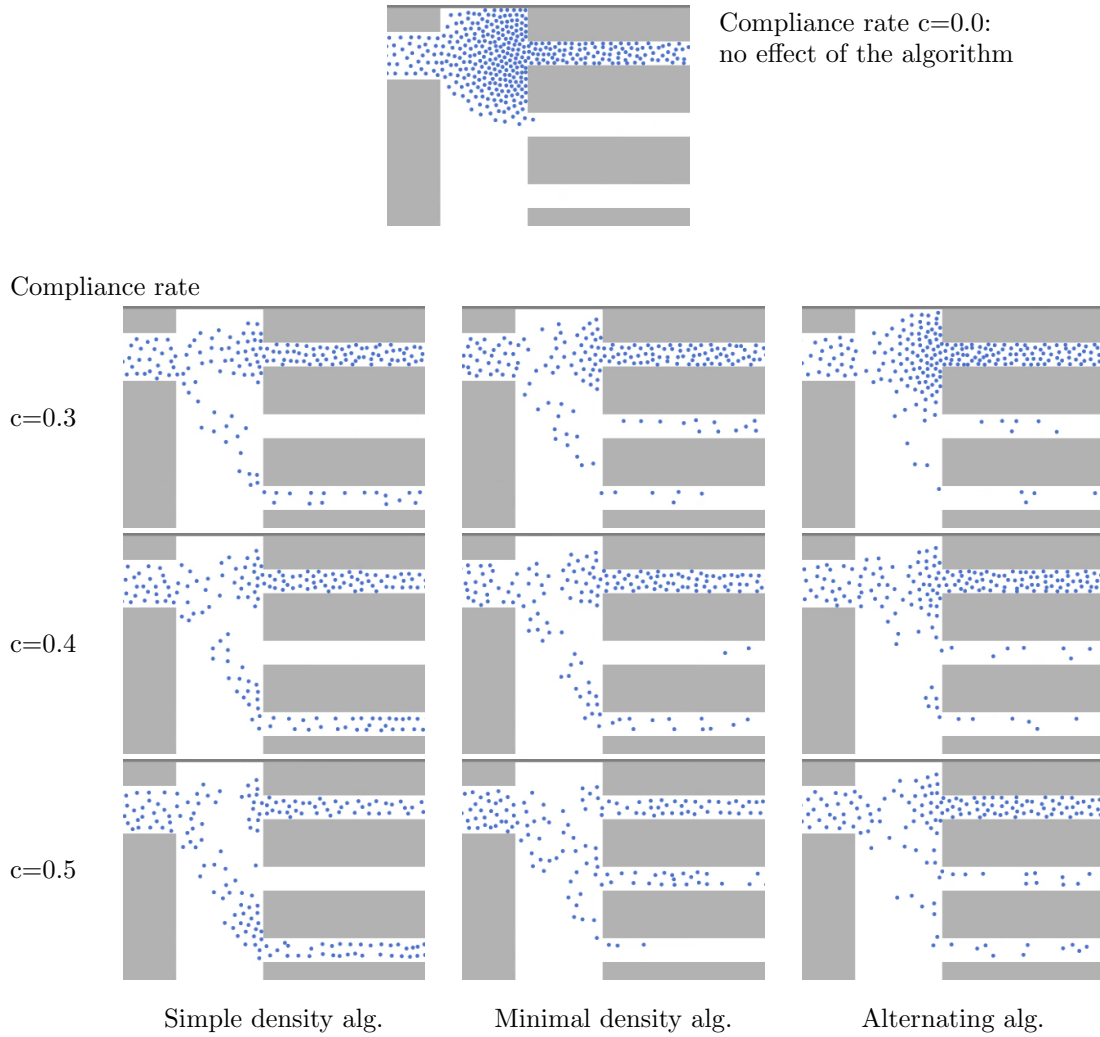
Figure 4.19.: Simulation snapshots at simulation time $t = 300\,\mathrm{s}$. The congestion in front of the bottleneck dissolves with increasing compliance rate. With the density-based algorithms, the congestion is almost completely resolved at a compliance rate $c = 0.3$. With the *alternating algorithm*, the congestion is resolved at a higher compliance rate ($c = 0.4$) because this algorithm cannot respond to the current density situation.

Next I analyze which routes are recommended at certain compliance rates. The *minimal density algorithm* never recommends the short route when the compliance rate is low: $c < 0.575$. The *simple density algorithm* recommends the long route all the time when the compliance is $c < 0.425$. Therefore, the proportion of people directed away from the short route is in both cases $q = 1$. Now that $q = 1$ is known, the minimum compliance rates can be estimated according to Eq. 4.3 for the two density-based algorithms. The

resulting minimum compliance rate is $c = 0.19$. This is line with the density situation, see Fig. 4.18 (top left and top center). The density behavior changes for both algorithms at $c \approx 0.2$. Interestingly, the median travel time becomes minimal as soon as congestion is resolved, see Fig. 4.17. This is true for any scenario where the waiting time caused by congestion is longer than the time required to take a longer alternative route. Importantly, it shows that resolving congestion not only improves safety but can also improve travel service by reducing travel times.

An important finding is that the two density-based algorithms can compensate missing compliance better than the *alternating algorithm*, see again Fig. 4.18 (top left and top center). The lower the compliance rate, the more non-compliant agents congregate in the short corridor so that the density-based route recommendation algorithms favor alternative routes. Thus, congestion is resolved at lower compliance rates, see again Fig. 4.19. Note that for any algorithm, the non-compliant agents remain in the short corridor where the density is high. Only compliant agents enjoy a better level of service along the medium or long route. Fig. 4.18 shows that the *minimal density algorithm* fully compensates the lack of compliance for compliance rates $c > 0.6$: the median density is at a fixed value. For the *simple density algorithm* the threshold is similar: $c = 0.5$.

Similar level of services for the route recommendation algorithms

To evaluate the safety, the maximum densities are analyzed. For all of the algorithms, the maximal density values occur on the short route, see again Fig. 4.18. The density values in the short route (top row) are higher than the density values in the medium and long route (second and bottom row) for any compliance rate.

For each algorithm, 41 simulation runs are available that refer to the 41 discrete compliance rates. For each simulation run, the maximum density is extracted. The resulting distributions are depicted as boxplots in Fig. 4.20. One can observe that the *alternating algorithm* achieves services levels according to Fruin that range from C ($< 0.72\,\mathrm{ped/m^2}$) to E ($< 2.17\,\mathrm{ped/m^2}$). The density-based algorithms achieve densities that correspond to the service levels D ($< 1.08\,\mathrm{ped/m^2}$) and E ($< 2.17\,\mathrm{ped/m^2}$). The spread is large because the compliance rate has a strong influence, see again Fig. 4.18.

In order to test whether one of the algorithms leads to lower maximum densities and, thus, to better safety, the maximum densities are compared statistically. First, the distributions are tested for normality. KS-tests reveal that the three distributions are not normally distributed (*Alternating algorithm*: $D = 0.985$, p-value $= 0.000$, *Minimal density algorithm*: $D = 0.986$, p-value $= 0.000$, *Simple density algorithm*: $D = 0.985$, p-value $= 0.000$).

Because the distributions are not normally distributed a Kruskal Wallis test is used to test whether the median values of the maximum densities differ. Since the p-value is larger than the significance level of 0.05 ($H = 1.010$, p-value $= 0.603$), I cannot reject the null hypothesis that the median density is the same for all three algorithms. Thus, I do
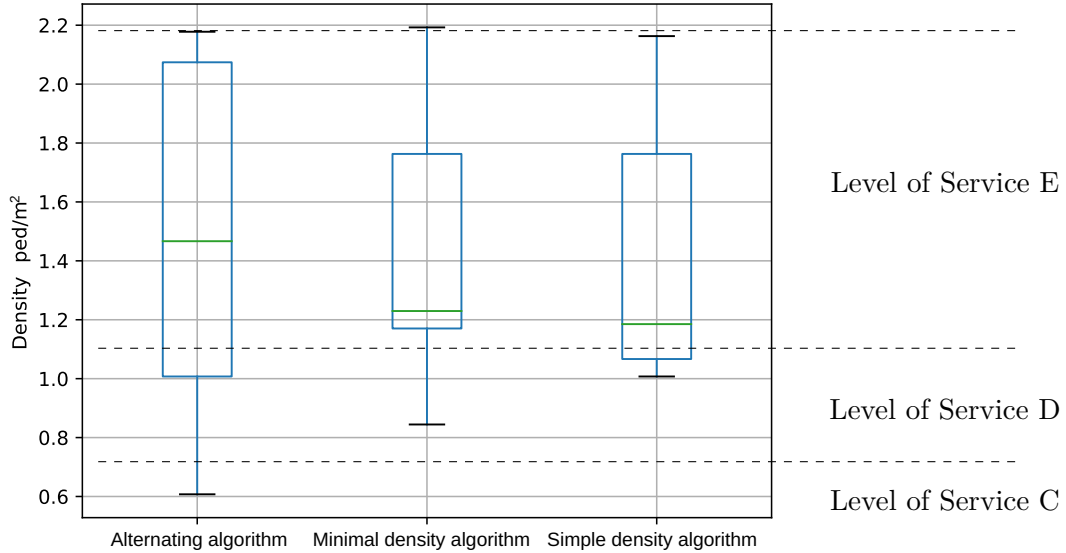
Figure 4.20.: Boxplots of the maximum densities. For each route recommendation, I extracted the maximal density from each of 41 simulation runs that differ in their compliance rate. A Kruskal Wallis test reveals that the medians (green line) do not differ statistically.

not have sufficient proof to claim that different route recommendation algorithms lead to statistically significant density differences. I conclude that there is no superior route recommendation algorithm that outperforms the others. All three route recommendation reduce congestion similarly successful.

The simple density algorithm requires few redirection measures

Finally, I look a the number of route recommendations generated by the three algorithms. I assume that an algorithm is more likely to be implemented in practice when it requires only a low number of interventions in the system. Also, I expect that the acceptance of the crowd guidance system is higher when people do not receive redirection instruction permanently. According to their definition, the *alternating algorithm* and the *minimal density algorithm* generate a route recommendation every 2 s. The *simple density algorithm* only generates a route recommendation when the short route is more occupied than the long route. In the scenario, the *simple density algorithm* intervenes the system less often than the other algorithms, see Fig 4.21. One could argue that the number of recommendations is similarly low for the *alternating algorithm* if the short

default route was not recommended. This is true, but the algorithm cannot resolve congestion at a low compliance rate. I conclude that the *simple density algorithm* has the best cost-benefit ratio: It resolves congestion successfully, while the crowd is only redirected if necessary.
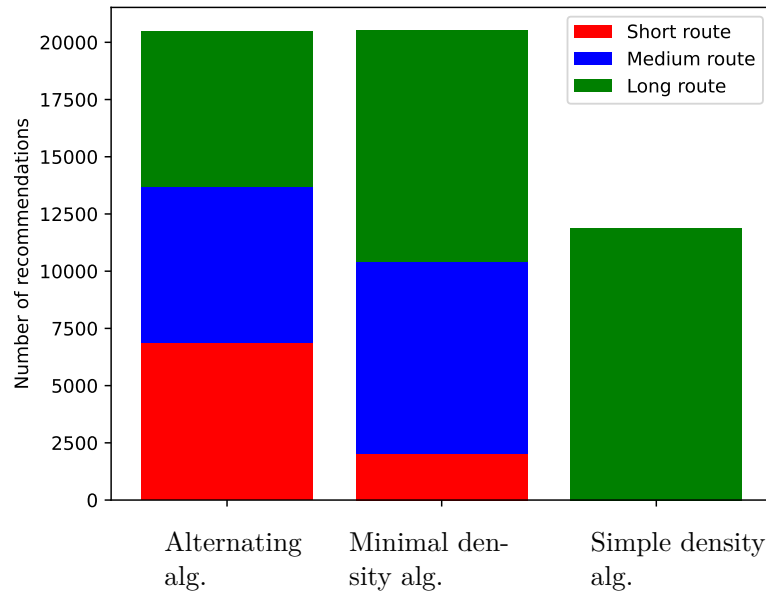

Figure 4.21.: Number of route recommendations. Each algorithm is evaluated 20500 times (= 500 route recommendations x 41 compliance rates) The *alternating algorithm* recommends every route equally often according to its definition (left). The *minimal density algorithm* recommends the short route less often to resolve congestion (middle). The *simple density algorithm* only recommends the long route when necessary.

4.2.5. Summary, limitations, lessons learned

Summary of the study

In this section I tested how suitable route recommendation algorithms are for resolving congestion when the proportion of pedestrians complying with route recommendations is uncertain. Since I could not find suitable algorithms in my literature review I proposed three algorithms myself: The first algorithm sequentially directs pedestrians to different routes. The second algorithm redirects pedestrians the route with the lowest density. In the third algorithm, pedestrians are directed away from the congested route if the density here is higher than on the longest alternative route. To test the algorithms, I

modeled a simple topography with three routes of different lengths and similar corridor widths. In a parameter study the compliance rate, that is the proportion of pedestrians that comply with route recommendations, was varied between $0\,\%$ and $100\,\%$. The results showed that density based algorithms resolve congestion at lower compliance rates than the alternating algorithm that does not consider the current density situation. I found that all of the algorithms lead to a reduction of the travel time in my scenario. Interestingly, the travel time was not minimal at high compliance rates since the algorithms consider densities but do not optimize travel times. To evaluate safety, I analyzed the density situation. I found that the maximum densities do not differ statistically: All algorithms successfully resolved congestion. For this reason, I looked at the number of redirection measures. The simple density-based algorithm, which only recommends the long alternative route but not the medium route, required fewer redirection measures than the other algorithms while resolving congestion at a low compliance rate. Therefore, I concluded that this density-based algorithm is most suitable for redirecting pedestrians.

> **Which algorithms are suitable for generating route recommendations for crowds, given not all people follow instructions? (RQ-2)**
>
> A density-based route recommendation algorithm can sufficiently avoid or resolve congestion by recommending the longest available alternative route if and only if the pedestrian density on this route is lower than on the congested route.

Limitations and future directions

I would like to point out some limitations of the study.

The proposed algorithms are based on simple heuristics. They cannot guarantee that pedestrians will be distributed along the routes in the most effective way.

In the study, a fixed inflow rate was assumed. Therefore, the minimum compliance rate, that is, the proportion of people who need to comply with the route recommendation to resolve congestion, is specific to this scenario.

The study assumed that people are evenly distributed within a route. In a real scenario, twists and turns create bottlenecks which lead to density differences along the route. Strategies must be found to ensure that local safety-critical congestion is not overlooked. I will propose a procedure in Section 4.4.

In the study, it was assumed that the density information and the route recommendations are immediately available and perfectly precise. If the crowd is sensed and informed using direct communication technologies, delays, and measurement errors may affect the capability of an algorithm to distribute pedestrians. This requires further investigation. I will investigate this in Section 4.4.

Lessons learned for my further investigations

The study demonstrated that heuristic route recommendation algorithms can indeed resolve safety-critical congestion. Heuristics that redirect people to less frequented routes based on crowd density measurements proved to be particularly effective. They resolved congestion even if only a small proportion of people followed the route recommendation by directing them away from the congested route sufficiently often.

One algorithm proved to be particularly suitable because it required only a few redirection measures, while it was capable of resolving congestion even at a low compliance rate. The algorithm recommends the longest available route if and only if the pedestrian density on this route is lower than on the congested route. Therefore I suggest to use this algorithm in a crowd guidance system.

4.3. Designing mobile messages for redirecting crowds

In this section, I investigate my third research sub-question (RQ-3), that is, how mobile messages should be designed to improve the compliance of crowd members to follow route recommendations. The study is inspired by a real-life use case in Munich (Germany) where football fans need to be guided from the bus or tram to the metro. Before and after football matches, the short direct route is usually congested although there are alternative routes available. To alleviate the congestion, I suggest to use a mobile application which works according to the route recommendation algorithm that I found was most suitable (see the previous section): The long alternative is recommended if and only if the short route is congested.

The more people follow the route recommendation, the better congestion is avoided or resolved. This is why communication plays a crucial role in a crowd guidance system. A crowd guidance system will not have any effect, when the information does not appeal to crowd members. From the social identity approach, it is known, that peoples' response is influenced by their social identity. Therefore, it is likely that the route choice of football fans at a train station indeed depends on their social identity. While conventional navigation applications are designed to fulfill the needs of individuals only, the focus of this study is on the message design in a navigation app for crowd members who share a social identity. In particular, I ask:

- How do appeals to the social identity improve crowd members' compliance to route instructions?

- Which other elements are necessary in a mobile message to convince crowd members?

Due to its interdisciplinary character, the study was developed in collaboration with a crowd psychologist, see Appendix A. The results of the study were published in the journal article *Designing mobile application messages to impact route choice* [307]. Therefore, this section contains textual overlaps with the publication.

4.3.1. Application scenario at metro station Münchner Freiheit

At the metro station Münchner Freiheit in Munich travelers typically take the shortest route to get from the bus station to the trains. Especially near escalators and elevators this leads to jamming and poses a safety risk, so that traffic managers are looking for ways to nudge the fans towards alternative, but longer routes. In total, there are three routes of different lengths to get from the bus or tram to the metro, see Fig. 4.22.
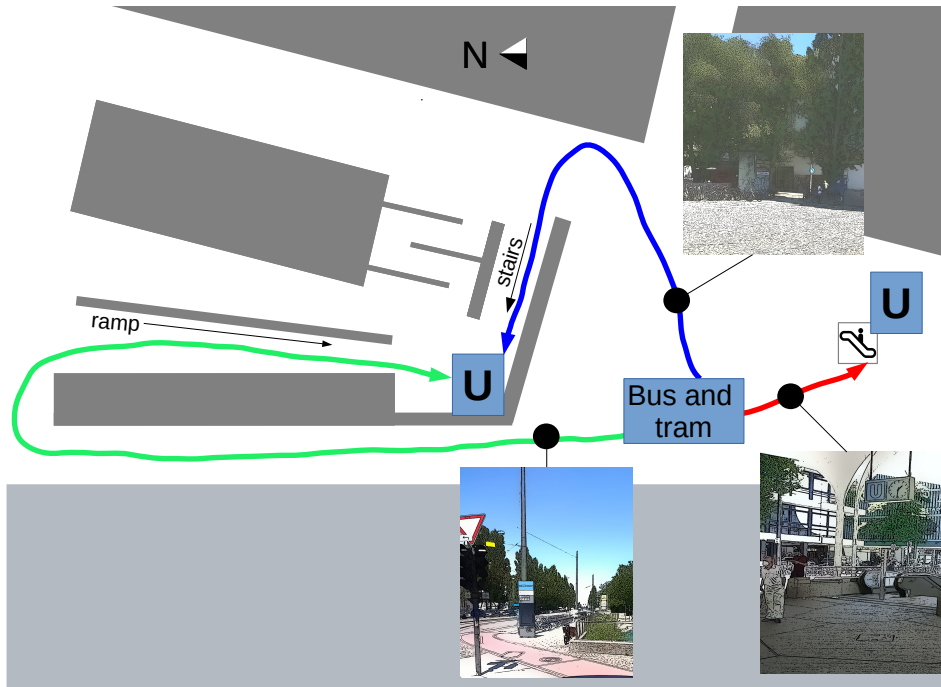
Figure 4.22.: Application scenario of the online survey. At the train station Münchner Freiheit in Munich (Germany), three routes lead to the underground trains. Note that the map is rotated counter-clockwise (North points to the left). Footballs fans walk from the bus or tram to the underground train. The pictures depict football fans' field of vision at the different routes. The sign to the underground (big blue U) is only visible for the short route (red).

The big blue letter **U** represents an entrance to the intermediate level where all pedestrian streams come together and from which pedestrians access the underground level. To avoid crowding, fans are redirected to the long alternative route (left route marked

in green) using a mobile application. Fans getting of the bus or tram can only see the U-sign pointing to the short route (picture at the bottom right). Only fans familiar with the environment are aware about the other entrance.

4.3.2. Survey design: groups, information state and message design

To test the hypothesis that football fans react differently to information than individuals due to their shared social identity, two online surveys are conducted: one with students and faculty associates at Hochschule München (Munich, Germany), and one with fans of the football team FC Bayern München. For each study, the same type of messages are used, see the study design in Fig. 4.23.
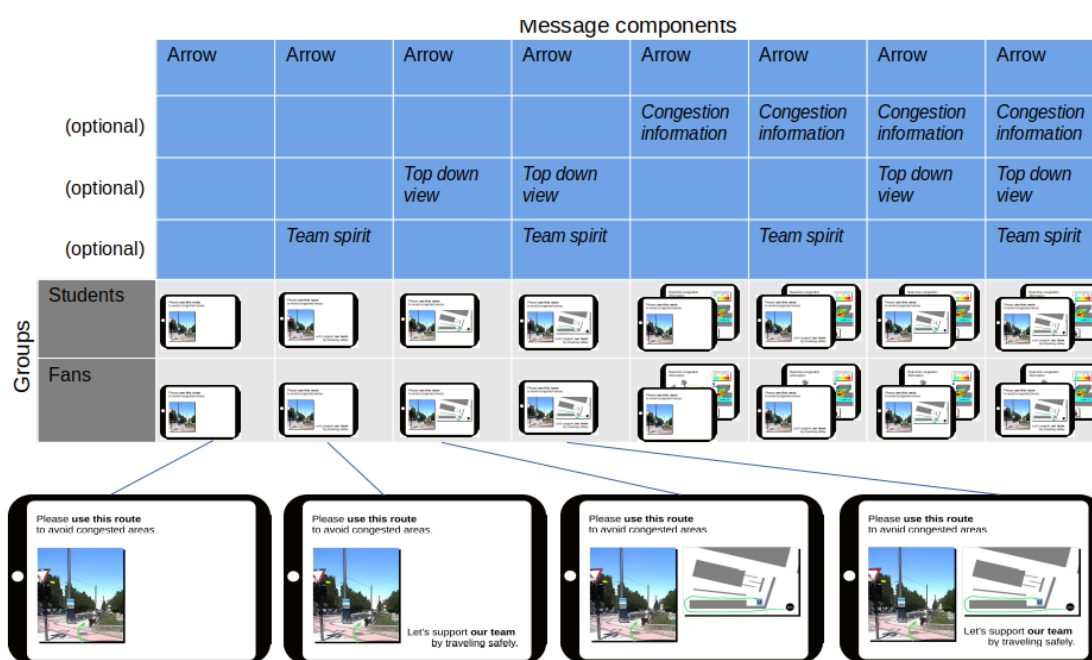
Figure 4.23.: Experimental between-subject design. The eight message levels result from the combination of three optional message components: *congestion information*, *top down view*, and *team spirit*. For a detailed view of the message components see Fig. 4.24. Figure from my publication [307].

The between-subjects design has one factor, message design, with eight levels. The levels stem from the combination of components. The first component is real-time *congestion information* where congestion is highlighted through a color scheme. The second is a *top down view* that depicts the underground train station from a top down view and with the recommended route marked in green. The third component is labeled *team spirit* and refers to the motivational phrase "Let's support our team by traveling safely" (german original: "Lass uns unser Team unterstützen, indem wir sicher reisen"). The combination

of these three components leads to $2^3 = 8$ message designs in total. Regardless of the combination, every message design contains a picture with an *arrow* that points in the direction of the long route and a route recommendation that is phrased as follows: 'Please use this route to avoid congestion.' (german original: "Bitte benutze diese Route, um Gedränge zu vermeiden."). Fig. 4.24 depicts the design with all optional components.
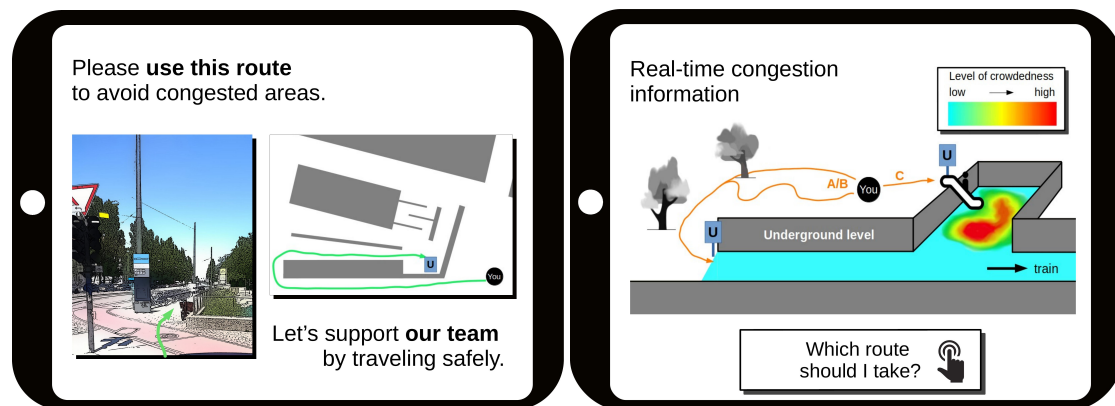

Figure 4.24.: Message design with all possible message components. Left: each message design contains the sentence 'Please use this route [...]' and a picture with an *arrow* pointing to the entrance of the recommended route. The components *top down view* and *team spirit* ('Let's support [...]') are optional. For message designs without *congestion information*, only the screen on the right is displayed to the participant. For designs with *congestion information* included, both screens are displayed. Imagine that in a real-life application, the left screen is what the app user sees first, followed by the second screen that appears when clicking on "Which route should I take?" (right). In the survey, the screens appeared next to each other. Figure from my publication [307].

The within-subjects design has one factor, information provision, with two levels: *prior to information*, and *after receiving information*. To collect quantitative self-report data, a questionnaire was used. The questionnaire is published under the doi: https://doi.org/10.1371/journal.pone.0284540.s001.

Maps of the scenario were depicted to participants to ensure they have similar knowledge about the surroundings, see the questionnaire. The goal was to create controlled conditions for locals and people not familiar with the scenario.

The dependent variables are: route attractiveness ("How likely is it that you take the following routes?"), and social identification ("I can imagine being part of the fan community."). For each, responses are scored on a 5-point Likert scale (1 = Very unlikely, 2 = Unlikely, 3 = Neutral, 4 = Likely, 5 = Very likely). The Likert-scaled estimate measures the attractiveness of each route.

I assumed that route choice behavior changes if at least one of the three route attractiveness differs statistically when recommending the long route. I also required that not all levels increase or decrease at the same time which ensures that there is an actual change in the route choice behavior. It was always fulfilled in the study.

4.3.3. Online survey with football fans and control group

I strictly follow ethical codes for investigations with human participants. Before the online survey was launched, ethical approval was obtained from the Hochschule München University of Applied Sciences (Munich, Germany), see Appendix F. Participation was voluntary and there were no monetary incentives. At the beginning of the survey, participants were informed about the research and told that their anonymised answers would be made publicly available. I also informed them that they were free to withdraw from the survey at any time. To participate in the survey, they had to confirm their consent and that they were at least 16 years old at the time of the participation.

To conduct the survey, I used the free and open source online survey tool *LimeSurvey*. See www.limesurvey.org for more information. Conditions were randomly assigned to participants. *Students and faculty associates* were recruited through Facebook, the *Vadere* research simulator webpage (www.vadere.org) and a general email to students and faculty associates at Hochschule München. Also, I sent personal appeals to the students of the faculty of Informatics and Mathematics at Hochschule München. Football fans are recruited using the official FC Bayern fan group on Facebook (www.facebook.com/FCBFanbetreuung) and Twitter (www.twitter.com/FCBayern_FB).

The survey for the *students and faculty associates* was conducted from 16.11.2021 to 23.02.2022. The survey for the *fans* ran from 24.02.2022 to 01.03.2022.

In total, 1365 out of 3067 (44.51 %) participants completed the survey, while 49.12 % dropped out in the introductory section, before or immediately after the declaration of consent. 444 out of these 1365 participants were *students or faculty associates* (43 % female, 54 % male). 921 participants were fans (18 % female, 80 % male). Many *fans* (72 %) were between 26 and 50 years old, see Tab. 4.8. *Students and faculty associates* were younger: only 36 % fell in the range from 26 to 50, half of them were younger than 26, and 14 % were older than 50.

Group	<18	18-25	26-35	36-50	51-65	>65	n
Fans	0 %	17 %	38 %	34 %	10 %	1 %	921
Students and faculty associates	7 %	43 %	28 %	8 %	11 %	3 %	444

Table 4.8.: Age distribution of the two groups: *football fans* (1), and *students and faculty associates* (2). *Students and faculty associates* were younger than fans which might be explained by the fact that most of them were students.

4.3.4. Route recommendations make the long route more attractive

Prior to information, both *fans* and *students and faculty associates* select their routes according to path lengths. They favor the short route overall, see Fig. 4.25. Please also find the statistical proof in Tab. G.2 in Appendix G. The route choice changes when the long route is recommended: The long route becomes more attractive, and the short route less attractive (compare the top and bottom rows in Fig. 4.25). The exact values are listed in Tab. G.7-G.8 in Appendix G.

As expected, Kruskal-Wallis tests reveal that the message design has a stronger effect on *fans* than on *students and faculty associates*. For the *fans*, the message design has a significant effect on the attractiveness of the short ($p = 0.0000$, $H = 95.76$, $df = 7$), medium ($p = 0.0000$, $H = 33.04$, $df = 7$) and long ($p = 0.0022$, $H = 22.42$, $df = 7$) route, while for the *students and faculty associates*, the design only makes a significant difference for the short ($p = 0.0000$, $H = 45.35$, $df = 7$) and medium ($p = 0.0000$, $H = 32.38$, $df = 7$) route. The attractiveness of the long route is not affected ($p = 0.8010$, $H = 3.8113$, $df = 7$). An overview of the statistical results is given in Tab. G.3 in Appendix G. From this, I conclude that *fans* are more responsive to message design than *students and faculty associates*.

Interestingly, *students and faculty associates* seem to be more willing to follow route recommendations in principle because the mean attractiveness of the long route was 3.84 for the *students and faculty associates* and only 3.69 for the *fans*, see Tab. 4.9.

Message design	Attractiveness of long route: mean value	
	Students and faculty associates	Fans
Congestion info + arrow	3.86 (n=58)	3.50 (n=103)
Congestion info + arrow + top down view	**4.02** (n=45)	3.90 (n=116)
Congestion info + arrow + team spirit	3.70 (n=50)	**3.98** (n=102)
Congestion info + arrow + top down view + team spirit	3.88 (n=61)	3.88 (n=107)
Arrow	3.79 (n=47)	3.43 (n=143)
Arrow + top down view	3.94 (n=65)	3.55 (n=111)
Arrow + team spirit	3.70 (n=56)	3.55 (n=103)
Arrow + top down view + team spirit	3.82 (n=62)	3.74 (n=136)
Average of means	3.84	3.69

Table 4.9.: Attractiveness of long route expressed through scores on a 5 point Likert scale. I asked the participants how likely it is that they take the long route (1: very unlikely, 2: unlikely, ..., 5: very likely). The higher the mean value, the more attractive the long route. The message design that contains all optional components (row 4) does not achieve the highest values: 4.02, 3.98. Note that n is the sample size.
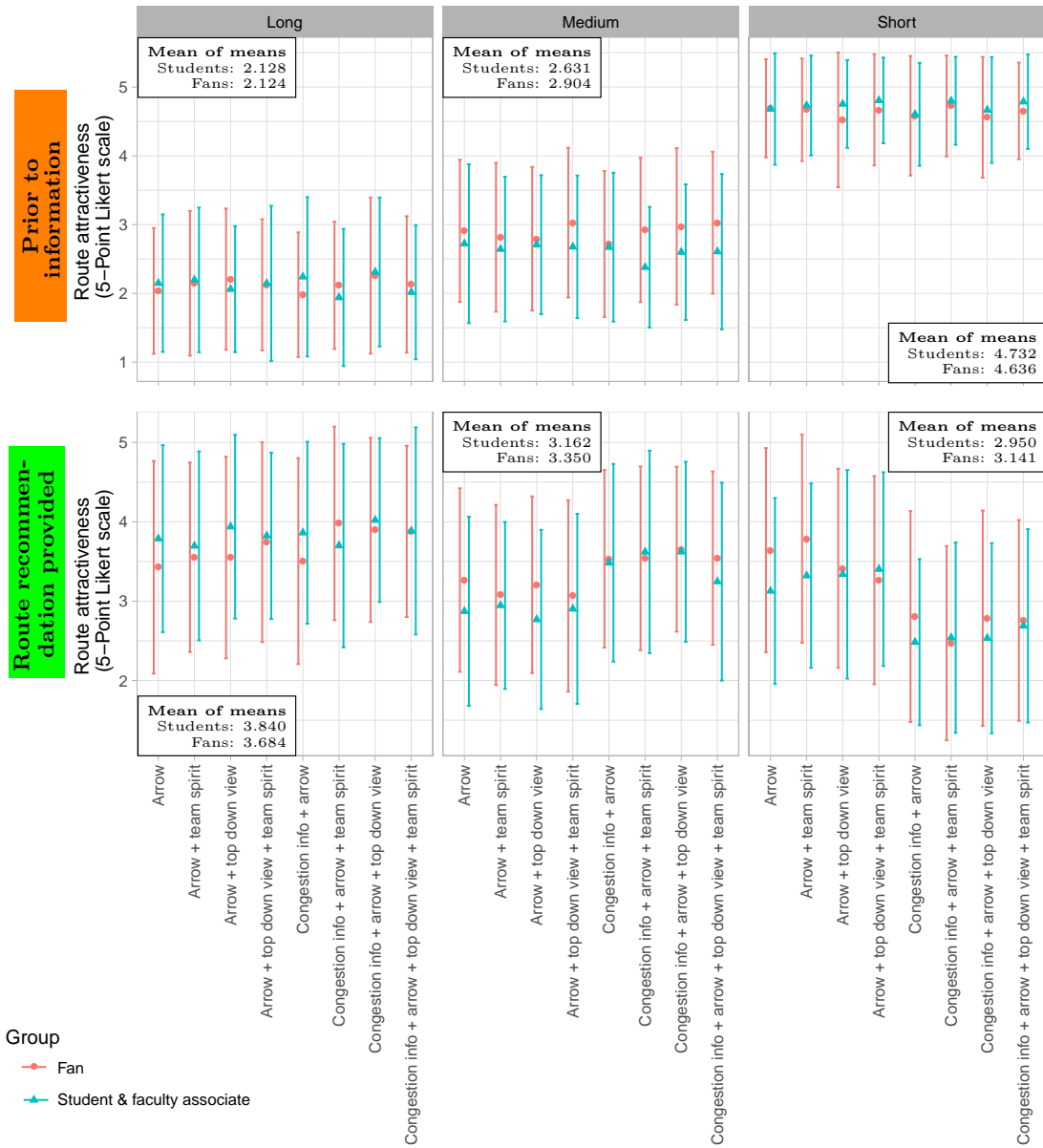
Figure 4.25.: Attractiveness of routes expressed through the Likert scale value for each message design. The error bars around the mean values (dots) represent the standard deviations. Prior to information, *students and faculty associates* and *fans* have a clear route preference: They favor the short route (top right) over the medium route (top center) followed by the long route (top left). With information provision, the short route becomes less attractive (bottom right) compared to the setting without information (top right), while the long route becomes more attractive (left). Figure from my publication [307].

4.3.5. Effect of message design on route choice

I then conduct Mann-Whitney U tests to assess the effect of adding the message components *congestion information*, *top down view*, and *team spirit* on the attractiveness of each route. I add one message component at each step, thus mitigating, to an extent, the inter-dependencies between designs. One can observe that adding components does not always help direct people away from the short congested route, but it never had a negative effect: through adding components, the short route becomes less attractive (see Tab. 4.10, 4.11, 4.12) or remains equally attractive which means that the difference is statistically not significant. The non-significant results are listed in Appendix G in Tab. G.5 and G.6.

At the same time, the medium and long routes become more attractive or remain equally attractive, see again Tab. 4.10, 4.11, 4.12 and Tab. G.5-G.6 in Appendix G. Crucially, adding all components together does not lead to an increased attractiveness of the long route: the mean was only 3.88 for both groups, see again Tab. 4.9. Hence, there is evidence for an interaction between the combinations.

There is strong evidence that the influence of the message components differs. The congestion information has always an effect on *fans* and *students and faculty associates*: The attractiveness of at least one route changes (see Tab. 4.10 and 4.11). The *top down view* and *team spirit* only has an effect in certain combinations with other message components. They also have a lower effect on route choice: either the short route becomes less attractive or the long route becomes more attractive. Both at the same time is never the case, see Tab. 4.12.

Message design	Route attractiveness		
	Short route	Medium route	Long route
Arrow	$p = 0.0019$, $W = 1804.0 \Downarrow$	$p = 0.0117$, $W = 989.0 \Uparrow$	*not sign.* *(p > 0.05)*
Arrow + top down view	$p = 0.0009$, $W = 1974.5 \Downarrow$	$p = 0.0003$, $W = 884.5 \Uparrow$	*not sign.* *(p > 0.05)*
Arrow + team spirit	$p = 0.0010$, $W = 1895.5 \Downarrow$	$p = 0.0031$, $W = 949.5 \Uparrow$	*not sign.* *(p > 0.05)*
Arrow + top down view + team spirit	$p = 0.0011$, $W = 2514.5 \Downarrow$	*not sign.* *(p > 0.05)*	*not sign.* *(p > 0.05)*

Table 4.10.: Effect of adding congestion information to message designs on the route attractiveness for *students and faculty associates*. For each message design, I use a Mann-Whitney U test to assess whether adding *congestion information* has an effect ($p < 0.05$) or not. Adding *congestion information* makes the short route always significantly less attractive ($\Downarrow$). The medium route becomes significantly more attractive ($\Uparrow$) except in the case where all other message components are already present (bottom row). The attractiveness of the long route is not affected.

Message design	Route attractiveness		
	Short route	Medium route	Long route
Arrow	$p = 0.0000$, $W = 9855 \Downarrow$	*not sign.* *(p > 0.05)*	*not sign.* *(p > 0.05)*
Arrow + top down view	$p = 0.0003$, $W = 8143 \Downarrow$	$p = 0.0021$, $W = 5006.5 \Uparrow$	$p = 0.0331$, $W = 5433 \Uparrow$
Arrow + team spirit	$p = 0.0000$, $W = 7940 \Downarrow$	$p = 0.0038$, $W = 4087.5 \Uparrow$	$p = 0.0047$, $W = 4107 \Uparrow$
Arrow + top down view + team spirit	$p = 0.0023$, $W = 8857.5 \Downarrow$	$p = 0.0022$, $W = 5678 \Uparrow$	*not sign.* *(p > 0.05)*

Table 4.11.: Effect of adding congestion information to message designs on the route attractiveness for *football fans*. For each message design, I use a Mann-Whitney U test to assess whether adding *congestion information* has an effect ($p < 0.05$) or not. Adding *congestion information* makes the short route always significantly less attractive ($\Downarrow$). For some message designs, the medium and long route become significantly more attractive ($\Uparrow$).

Added component	Message design	Route attractiveness		
		Short route	Medium route	Long route
Top down view	Arrow + team spirit	$p = 0.0024$, $W = 8555.5 \Downarrow$	*not sign.* *(p > 0.05)*	*not sign.* *(p > 0.05)*
Top down view	Arrow + congestion info.	*not sign.* *(p > 0.05)*	*not sign.* *(p > 0.05)*	$p = 0.0305$, $W = 5006 \Uparrow$
Team spirit	Arrow + congestion info.	*not sign.* *(p > 0.05)*	*not sign.* *(p > 0.05)*	$p = 0.0069$, $W = 4157 \Uparrow$

Table 4.12.: Effect of adding a *top down view* or *team spirit* to message designs for football fans. For each message design, I use a Mann-Whitney U test to assess the effect of adding a *top down view* or *team spirit*, that is present, if $p < 0.05$. When adding a *top down view* to the message design with *arrow* and *team spirit* (first row), the short route becomes significantly less attractive ($\Downarrow$). For the message design with *congestion information* (second row), the long route becomes significantly more attractive ($\Uparrow$). The *team spirit* makes the short route less attractive only in combination with the *arrow* and the *congestion information* (bottom row).

An important finding is that social identities can have an effect on *fans'* intended route choice behavior when appealing to their *team spirit*. Adding *team spirit* has an effect on *fans* for one message design (see Tab. 4.12), but never on students and faculty associates (see Appendix G: Tab. G.5). One explanation is that the spirit manipulation only works when there is an existing team identity to manipulate. I therefore investigate how strongly the two groups could imagine being part of the group by asking "I can imagine being part of the fan community" (german original: "Ich kann mir vorstellen, ein Teil der Fangemeinde zu sein."). In fact, there is a significant difference between *fans* (mean = 3.859) and *students and faculty associate*s (mean = 3.252), ($W = 270588$, $p < 0.05$).

However, evidence for the influence of social identities is given for one message design only. I do not believe that this is a statistical coincidence: There are other factors that may mask the effect like, for example, adding increasing amounts of information may lead to information overload. Also the reaction may already have been strong before *team spirit* was added, since the addition of the *team spirit* and the *top down view* together do not make the long route more attractive. A Kruskal-Wallis test reveals that the long route is equally attractive for designs that contain – in addition to *congestion information* – a *top down view* ($n = 116$, mean= 3.897) or *team spirit* ($n = 102$, mean = 3.980) or both ($n = 107$, mean = 3.879) ($p = 0.39$, $H = 1.884$, $df = 2$). From this, I conclude that the social identity can indeed be influential, but it is very sensitive to environmental conditions and depends on a pre-existing social identity to prime.

4.3.6. Summary, limitations, lessons learned

Summary

I investigated my third research sub-question for a real-life use case at the metro station Münchner Freiheit (Munich, Germany) where football fans need to be guided away from a congested route. Motivated by findings from social psychology on social identities, I developed eight message designs suitable for social groups, such as football fans, and investigated which message components have an effect on the route choice. In an online survey the message designs were tested for two groups: football fans, who were assumed to share a social identity, and a control group consisting of students and faculty associates. The participants rated the attractiveness of the three main routes from the bus to the underground, once without and once with detour information provided as a mobile message. Each participant was assigned one message design.

The results indicated that people are willing, in principle, to follow route recommendations provided by a mobile application. However, there was no ideal message design to convince any kind of social group to change their intended behavior. Messages that appeal to the team spirit increased the compliance of football fans when combined with other message components, but had no effect on students and faculty associates where social identification was lower.

The answer to the sub-question (RQ-3) is:

> **How should mobile messages be designed to improve the compliance of crowd members to follow route recommendations? (RQ-3)**
>
> There is no ideal design for mobile messages. Real-time congestion information proved most effective in fostering compliance. Appealing to the social identity of football fans increased the compliance under certain conditions.

I conclude that social identities indeed play a role in message design and suggest further research to assess how message design and social identity interplay. Combining instructions with explanations such as real-time information on congestion, and thus a reason why one should follow the instruction, proved most effective in fostering compliance.

Limitations

In the online survey, the participants completed the experiment individually. In reality, in-group members might base their decision on others. The inter-dependency of route choice was not assessed due to the study setup. However, I believe that group-based decisions rather help than hinder the success of the redirection, because in reality, sub-groups are likely to form within the crowd. Hence, people who do not use an application could still be redirected by following their in-group members.

Another assumption of this study was that football fans know the environment and consciously accept the long route. To ensure that participants were familiar with the surroundings, a map was displayed at the beginning of the survey. In reality, there would also be people unfamiliar with the environment. Hence, their orientation skills will influence whether they can take their chosen route. A non-representative survey, that I conducted on-site, indicates that people unfamiliar with the environment may have problems finding the recommended route, see Appendix H. Thus, additional measures such as static signs may be necessary to transfer the system into practice.

The analysis of the data indicated that there are most likely additional confounding factors, such as demographic characteristics. However, confounding factors were not anticipated and therefore made no provisions in the study design. Since the demographics of the overall populations are well represented in the survey with its high number of participants, I believe that the demographic characteristics do not affect the findings in a major way.

As the research design was based on online surveys, only the intended behavior could be assessed, not the actual behavior. It would be interesting to conduct field studies to observe people's behavior in real-life situations.

Lessons learned for my further investigations

In most message designs, the provision of real-time congestion information had an impact on the willingness to take a detour. Therefore, the traffic situation should be measured and visualized to the crowd.

When route recommendations were provided, people tended to avoid the congested route and favored the alternatives. This strongly indicates that people are indeed responding to information about pedestrian traffic, which is a prerequisite for the success of a crowd guidance system.

4.4. Synthesizing a crowd guidance system: a proof of concept

In this section I come back to my overall research question (RQ): How can crowds be redirected through mobile applications based on direct communication technologies? This section differs from the previous sections, where I investigated isolated components and sub-systems of a crowd guidance system. Here, I propose and test a concept for the metro station Münchner Freiheit (Munich, Germany) from Section 4.3.1, comprising all components and procedures necessary to implement a crowd guidance system in practice. I will simulate a real-life use case where football fans are guided. I pick up and extend findings from Sections 4.1-4.3, see Tab. 4.13.

Finding from previous section	Extension of this study
Section 4.1: As long as there is no shadowing, information is disseminated within a few seconds using WLAN 802.11p communication.	I extend my findings by testing the suitability of LTE sidelink communication to disseminate route recommendations.
Section 4.2: Crowds should be redirected depending on the current density situation. It is sufficient to recommend the longest alternative route if and only if the direct route is more congested than the alternative.	I apply this algorithm to a realistic topography with twists and turns and varying corridor widths. Since this algorithm requires density measurements, I employ the decentralized pedestrian density map application.
Section 4.3: Compliance is improved if mobile messages display current congestion information, a directional arrow, and appeals to team spirit.	I use such a message design to redirect football fans. I show how to model and simulate the effect of a message design.

Table 4.13.: Findings from Sections 4.1-4.3 and extensions of this study. I synthesize a novel crowd guidance system from components that I analyzed in the previous sections.

The section is structured as follows. First, the application scenario Münchner Freiheit is briefly recalled. Next, a concept for a crowd guidance system is proposed which includes the description of the model and its implementation. The concept is tested in a simulation study where the effect of the crowd guidance system on the congestion situation is analyzed. Also, the computational effort for the coupled simulation is discussed. Finally, I answer my research question and discuss how to deal with the problem that no empirical data is available for model validation.

4.4.1. Application scenario at metro station Münchner Freiheit

The application scenario is the same as the one used in the online survey (see Section 4.3.1). Therefore, I describe only basic properties and the scenario's relevance for application. The scenario was developed in collaboration with Stadtwerke München, the provider of public transportation in Munich to better understand congestion occurring before and after football games at the Münchner Freiheit metro station in Munich (Germany). While large stations such as the Munich Central station or Marienplatz (Munich, Germany) deploy staff to safely guide individuals to their trains, medium-sized stations like the Münchner Feiheit lack such personnel. For such scenarios, an app-based crowd guidance system could prove beneficial. The Münchner Freiheit station is particularly interesting to study, given that it features multiple entrances to accessing the underground from the surface level, see Fig. 4.26. The short route spans approximately 40 m, the medium route about 100 m, and the long route is approximately 300 m in length. The three routes are predominantly used by transfer passengers switching from bus or tram to the underground train. Results from a on-site count study indicate that 3 out of 4 passengers opt for the shortest route (see Appendix I).
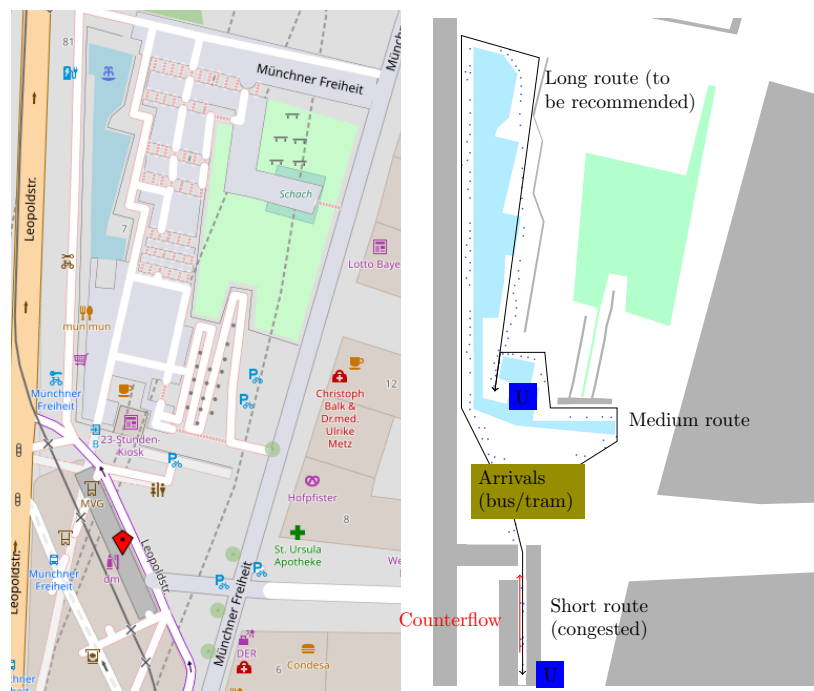
Figure 4.26.: Real-life scenario: guiding football at the metro station Münchner Freiheit (Munich, Germany). Left: Bird's-eye view of the surface level from © OpenStreetMap openstreetmap.org/copyright. Right: Topography model in the *Vadere* GUI. Obstacles are displayed in grey. The blue and green areas in Open Street Maps (left) are not accessible and are, therefore, modeled as (colored) obstacles.

4.4.2. Crowd guidance system: concept, model and implementation

Fig. 4.27 depicts my proposal for a crowd guidance system applied to the use case Münchner Freiheit.
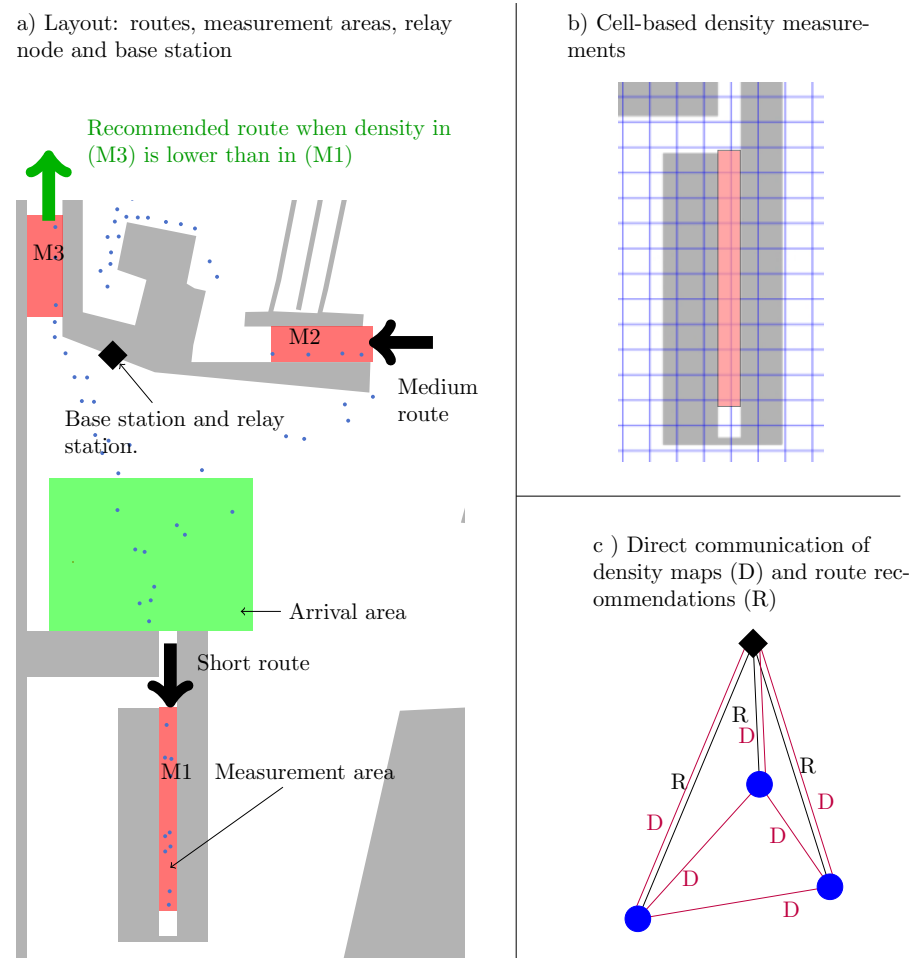
Figure 4.27.: Proposed crowd guidance system with controller station. Football fans (small blue dots) get off the bus or tram in the arrival area where an app provides route recommendations. The route recommendation algorithm, implemented in the controller station, compares the densities along the short (M1) and the long route (M3) every 2s (a). The long route is recommended if and only if the density in (M1) is higher than in (M3). The density values are average values derived from the density map application that counts pedestrian for each cell (b). Density information and route recommendations are disseminated in separate mobile apps using LTE sidelink communication (c) in the controlled mode.

133

The components of the crowd guidance system, depicted in Fig. 4.27, fulfill the following functions. The controller station, that is a computer including a network interface card for sidelink communication, evaluates the density-based route recommendation algorithm each time a fixed time interval has elapsed and disseminates the route recommendation over a mobile application in the crowd. The centralized computation of the recommendation ensures that pedestrians do not have contradictory recommendations displayed on their apps. The algorithm uses density measurements provided by the decentralized pedestrian density map application as input. For an overview of the data flow between apps and the route recommendation algorithm, see again Fig. 3.1 in Section 3. Both apps, the route recommendation application and the pedestrian density map application, use LTE sidelink communication in the controlled mode. To prevent individuals from detouring, route recommendations are only displayed in the arrival area (25 m×15 m) where football fans leave the bus or tram. The controller station is placed next to the arrival area to ensure coverage in the area where pedestrians arrive and need redirection information.

To test the concept through simulations, a system model is composed and implemented in *CrowNet*, see Tab. 4.14. The explicit method described in Section 3.2.2 is used as update scheme.

Component	Sub-component	Model	Implementation (Simulator)
Crowd	Locomotion model	Optimal Steps Model	OptimalStepsModel (Vadere)
	Perception model	Route choice probabilities	MultiPerceptionModel (Vadere)
	Cognition model	Probabilistic Cognition Model	ProbabilisticCognitionModel (Vadere)
Controller	Time stepping alg.	Fixed interval (2s)	FixedTimeStepper (flowcontrol)
	Algorithm	Simple density algorighm	AvoidShort (flowcontrol)
Network	Application layer	Detour application	CrownetUdpApp (CrowNet/app.)
		Density map application	DensityMapAppSimple (CrowNet/app.)
	Transport layer	Udp	Udp (inet)
	Internet layer	Ipv4	Ipv4NetworkLayer (inet)
	Air interface (PDCP)	3GPP TS 36.323	LtePdcpRrc(Ue/Enb)D2D (simu5G)
	Radio link layer (RLC)	3GPP TS 36.322	LteRlc (simu5G)
	Medium Access layer (MAC)	TS 36.321	LteMac(Ue/Enb)D2D (simu5G)
	Physical layer (PHY)	3GPP TS 36.201, 3GPP TS 36.211,-213	LtePhy(Ue/Enb)D2D (simu5G)
	→ Channel model	3GPP TR 36.873 (Microcell)	LteRealisticChannelModel (simu5G)

Table 4.14.: Composite model of the crowd guidance system.

The *Vadere* simulator provides mobility data and crowd behavior models, *OMNeT++* simulates the mobile network, and *flowcontrol* provides the route recommendation algorithm.

Modeling the topography and the arrival process in *Vadere*

To model the topography of the metro station Münchner Freiheit at the surface level I use bird's-eye views of the station that are publicly accessible via Open Street Maps. The topography spans an area of $164\,\text{m} \times 215\,\text{m}$. Areas, pedestrians cannot or should not step on, like buildings and major streets are modeled as obstacles. Agents are spawned in source A which is the area where football fans leave the bus or the tram, see Fig. 4.28.

According to the public transport provider of Munich, an average of 1200 people per hour change from bus or tram to the underground train at Münchner Freiheit before a football game. Between the arrivals of two trams or buses, the value might be lower, but shortly after the arrival of a tram, it is considerably higher. To capture the worst case I use 3600 ped/h (= 60 ped/min). In the simulation this flow is realized by spawning 2 agents every $2\,\text{s}$ in source A ($\text{x} = 66\,\text{m}$, $\text{y} = 77\,\text{m}$, $4.4\,\text{m} \times 3.6\,\text{m}$). A counterflow is formed by pedestrians walking from the underground to the bus. The counterflow is spawned in source B ($\text{x} = 69\,\text{m}$, $\text{y} = 41\,\text{m}$, $1.75\,\text{m} \times 1.5\,\text{m}$). Agents spawned in source B are assigned to target B, which represents a bus stop where people board the bus. To achieve a qualitatively similar flow situation as observed on site (see Appendix I), two agents are spawned every $3\,\text{s}$. The controller station and the base station are positioned close to the arrival area ($\text{x} = 62\,\text{m}$, $\text{y} = 95\,\text{m}$) to ensure coverage.

Route recommendation algorithm in *flowcontrol*

The density-based route recommendation algorithm computes a route recommendation every $2\,\text{s}$, ensuring the responsiveness of the crowd guidance system. A smaller time interval would not have an effect since the density measurements are updated every $2\,\text{s}$. The basic idea of the algorithm is to recommend the long route if the density on the short route surpasses the density on the long route. This recommendation is then disseminated as mobile message. Otherwise, no route recommendation is sent. At each moment, the route recommendation is identical for all individuals, preventing group separations due to conflicting information. To make a decision, the route recommendation algorithm requires a density value for the routes. I expect the maximum density at the bottleneck of each route where the corridor width is the smallest. Therefore, measurement areas are placed at the beginning of each route, see Fig. 4.28. The dimensions and positions are listed in Tab. 4.15. Their identical area size ensures equal smoothing of the density over time. The pedestrians arrive in the measurement areas at about the same time because the distance to the arrival area is similar. Therefore the effect of a redirection
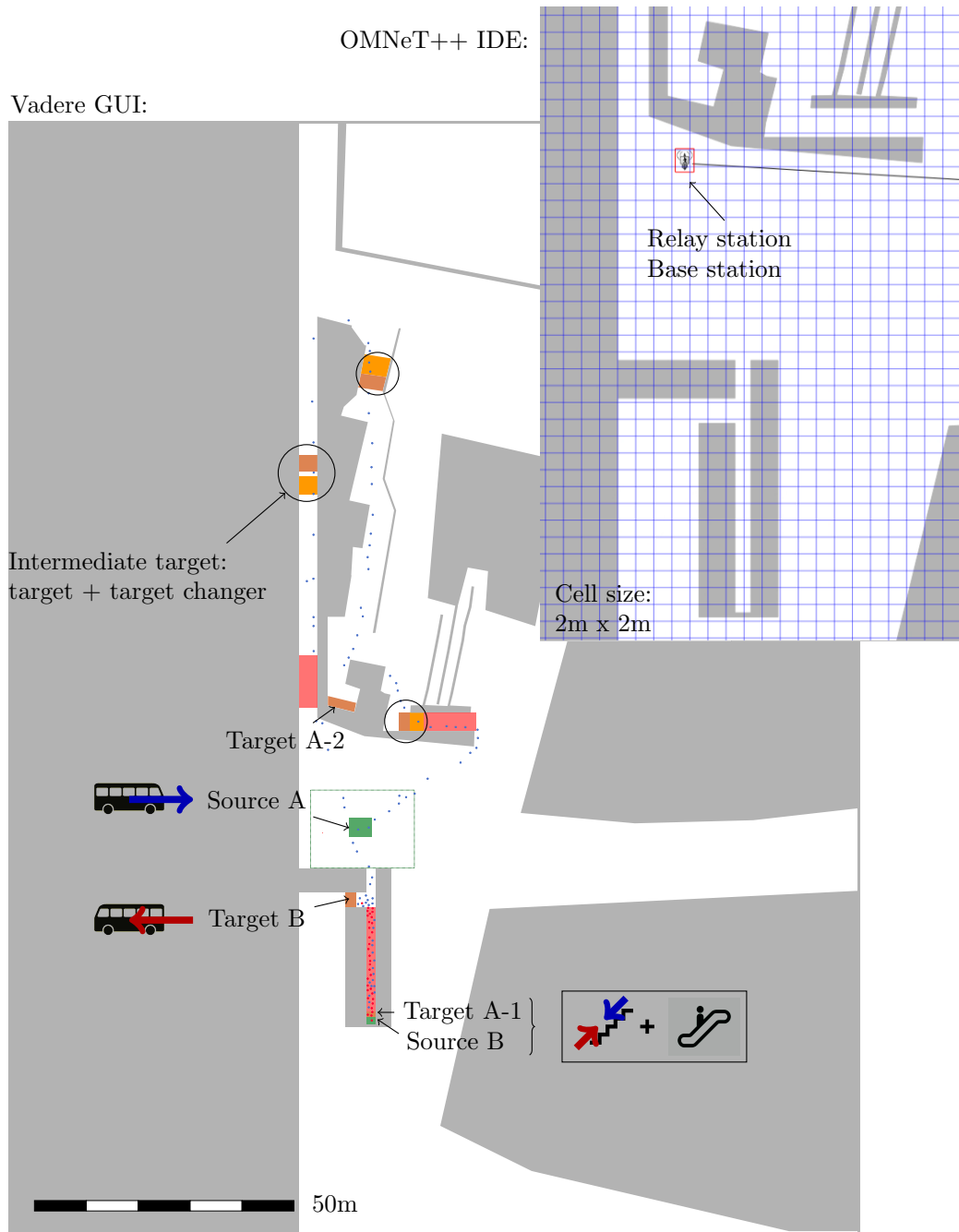
Figure 4.28.: Model of the topography (width: 164 m, height: 215 m). The topography and mobility behavior is modeled in the *Vadere* simulator (GUI on the left) using sources (green), targets (orange) and target changers (yellow). Blue agents are spawned in source (A) and walk along the three routes to get to the train (targets A-1 and A-2). Red agents walk from the train (source B) to the bus (target B) and form a counterflow. Red areas are measurement areas. In the *OMNeT++* simulator, the cell size, the position of the stationary relay station and the base station is defined.

Route	Coordinates (x,y) in m (Origin: bottom left corner)	Width in m	Height in m	Area in m^2
Short	(69.0,43.5)	1.75	20.0	35.0
Medium	(80.0,97.5)	10.0	3.5	35.0
Long	(56.0,102.0)	3.5	10.0	35.0

Table 4.15.: Dimensioning of the measurement ares. The identical area avoids artificial density discrepancies.

is measured with a similar delay, and, the effect of delay on the route recommendation algorithm can be neglected.

The *SimpleDensityAlgorithm* is implemented in the *flowcontrol* simulator. Every 2 s, the algorithm receives a pedestrian density map from the controller station that is simulated in the mobile networks simulator *OMNeT++*. The densities are mapped using the *DensityMapper* provided by the *flowcontrol* simulator.

Mobile application for disseminating route recommendations in *OMNeT++*

Route recommendations are disseminated through a mobile application based on LTE sidelink. The controller station sends route recommendations to pedestrians in proximity. Received recommendations need not be forwarded (single hop) because there are no obstacles between the controller station and the arrival area. The single hop ensures that the network load is low.

The mobile application is implemented as *BroadcastControlApp* in the *OMNeT++* module (Section 3.4). Every two seconds, the mobile application of the controller station can receive a route recommendation from the simulator *flowcontrol*. The recommendation is sent as single packet via broadcast to agents using UDP.

Mobile application for measuring densities: density map application in *OMNeT++*

The route recommendation algorithm requires density measurements. For this purpose the density map application [4] is employed (see again Section 3.4). In a real system the application would be implemented in the controller station. The application estimates the density for each cell based on the number of received beacons and based on density estimates shared by other devices. The parameter values can be found in Tab. 4.16.

The mobile application is composed of two services: One for disseminating beacons and one for dissemination density information (see again Section 3.4). The services are im-

plemented as separate applications in the *OMNeT++* module: as *DensityMapAppSimple* application and as *BeaconDynamic* application.

Application	Parameter	Value	Comment
DensityMapAppSimple	generationInterval	2 s	controls how often densities are updated
	mapTypeLog		Algorithm "Youngest measurement plus distance": Consideration of the actuality of the density value and the distance between the cell and the position of the device providing an estimate for the corresponding cell
	cellAgeTTL	15 s	exclude measurements/estimates older 15 s
	alpha	0.9	Parameters of the aggregation algorithm: weight the normalized actuality with the factor 0.9 and the normalized distance with 0.1.
	stepDist	60 m	The distance between the cell and the position of the device providing a density estimate for the respective cell has no influence on the aggregation of the density values if it is below the stepDist $= 60$ m; assumption: if the cell is close enough < 60 m, density estimates are similarly accurate.
	cell size	2 m	controls the spatial discretization
BeaconDynamic	generationInterval	0.5 s	controls how often beacons are sent
	maxAge	5 s	exclude beacons older 5 s

Table 4.16.: Parameter specifications for the density map application and the beacon application. I use default values except for the cell size that I reduce from 5 m (default) to 2 m since the smallest corridor width is 2 m in the scenario.

Modeling the protocol stack for LTE sidelink communication in *OMNeT++*

The LTE protocol stack is modeled as four layers: the Packet Data Convergence Protocol (PDCP) layer, the Radio Link Control (RLC) layer, the Medium Access Control (MAC) layer and the physical (PHY) layer, see again Fig. 2.11 in Section 2.3.

Different modes exist for the Radio Link Control (RLC) Protocol. I choose the unacknowledged mode which is recommended for direct communication [308]. Data packets are segmented, sorted and assembled. Data received more than once is discarded.

At the Medium Access Layer (MAC), the scheduling of resources is managed: every 1 ms resources are distributed over 25 bands at a carrier frequency of 2.6 GHz.

At the physical (PHY) layer, the signal propagation and interference is modeled. As channel model I choose the standardized microcell LTE 3D channel model provided by the 3GPP. The channel model computes the power of a received signal which is then used to estimate the signal to noise interference ratio. This values is used together with Block Error Rate (BLER) curves to decide whether data is successfully received or not.

The protocols and models of the four layers including the Packet Data Convergence Protocol (PDCP) layer are provided by the *simu5G* framework, see the technical specifications ('TS') in Tab. 4.14.

To enable direct communication in the simulation, the respective network interface card (NIC) in the mobile networks simulator's config file (*.ini) is selected: *LteNicUeD2D*. To model the Adaptive Modulation and Coding (AMC) scheme, the *amcMode* is set to "D2D" (device-to-device). For the communication over broadcast, the internet layer is specified as follows: all agents are assigned to a multicast group that is addressed using the IPv4 protocol and a suitable network mask.

Modeling participation: proportion of people using apps in $OMNeT++$

In reality, only a part of the crowd would use the two mobile applications described before. Based on the findings from [309], I choose a value of 40% which reflects the regular usage of navigation applications for individuals which, I think, can be transferred to a crowd guidance system. I do not model pedestrians' awareness, but assume that whoever installed the route recommendation application perceives the recommendation through a notification. Accordingly, 40 % of the individuals are equipped with smart phones in the mobile networks simulation.

Modeling crowd behavior in *Vadere*

The crowd model is composed of the locomotion model and the route choice model. For the locomotion behavior I use the Optimal Steps Model. To model how pedestrians distribute over the three routes when no guidance is provided, I use a discrete probability distribution that I derive from the survey data (Section 4.3). Agents spawned at source A are assigned to the three routes using a so-called target changer that is placed at source A in the topography model. Route changes after receiving route recommendations are modeled in the *ProbabilisticCognitionModel* which also uses probabilities values from the survey data.

In the survey, participants were asked to rate the attractiveness of the three routes on a 5-point scale (1-5). To estimate the route probabilities from the survey data, a model is needed that translates the Likert scaled route attractiveness into route probabilities. I suggest a simple model to interpret the scores from the survey as route frequencies. I published the model in [307]. The idea of the model is that a survey participant who rated the shortest route at 4, the medium route at 3, and the long route at 3 would pick the shortest route 4 out of $10(= 4 + 3 + 3)$ times, the medium 3 out of 10 times and the short 3 out of 10 times. I argue that this portrays the likely behavior in the long run.

With this in mind, I define the individual route probability $p_{j,m}$ that a participant m selects route j as

$$p_{j,m} = \frac{l_{j,m}}{\sum_j l_{j,m}} \tag{4.4}$$

where l is the route attractiveness that I take from the survey. Note that the division through $\sum_j l_{j,m}$ achieves that the sum of probabilities equals 1: $\sum_j p_{j,m} = 1$. This normalization solves the problem that participants might interpret the scale differently, e.g., the route distribution is $[1/3, 1/3, 1/3]$ no matter whether a participant rated the routes with $[2, 2, 2]$ or $[3, 3, 3]$. I argue that the average of the single behaviors is a good characterization of the behavior of the population as a whole. Therefore, I define the route probability p_j for the whole population as

$$p_j = \frac{1}{m} \sum_m p_{j,m} \tag{4.5}$$

With Eqs. (4.4) and (4.5), route choice probabilities are inferred from the survey data. Tab. 4.17 lists the estimated route probabilities for the group of footballs fans from the online survey (Section 4.3) with and without guidance. For the case with guidance, the route probabilities are computed for each message design separately because the survey showed that the message design has an effect.

Message design	Route probabilities		
	p_1 (short)	p_2 (medium)	p_3 (long)
Arrow	35	31	33
Arrow + top down view	34	31	35
Arrow + team spirit	36	29	35
Arrow + top down view + team spirit	32	30	38
Congestion info + arrow	29	36	35
Congestion info + arrow + top down view	27	35	38
Congestion info + arrow + top down view + team spirit	27	34	38
Congestion info + arrow + team spirit *(parameters of the cognition model)*	**25**	**35**	**40**
Prior to information *(parameters of the target changer)*	49	30	22

Table 4.17.: Route probabilities for the football fans. The route probabilities were estimated from survey data (Section 4.3) using the proposed model: Eqs. (4.4) and (4.5). The idea of the model is to translate the Likert scores by which survey participants rated the attractiveness of a route into percentages of passengers p_i that would take that route. The most convincing design is the one with the lowest probability of taking the congested short route (bold).

Without route recommendations, the short route is chosen with a probability of 49 %. This value deviates from the probability estimate from a manual counting experiment that is between 0.75...0.82 (see Appendix I). I attribute this large difference to the coarse nature of the counting study and to the assumption of a steady flow. Also, there were uncontrollable factors such as age and gender. The route choice probabilities in Tab. 4.17

indicate that with the message design *Congestion info + arrow + team spirit* more pedestrians are willing to take a detour than with the other message designs because the route probability of the short route is the lowest. Therefore, I use this message design in my investigations. The associated route probabilities serve as parameters of the *ProbabilisticCognitionModel*. The route choice probabilities corresponding to the case without route recommendation serve as parameters for the target changer.

4.4.3. Design of the simulation study

To assess whether safety and comfort can be improved with a crowd guidance system, I run simulations of the scenario with and without crowd guidance system. Due to the long duration of a single simulation run (see Section 4.4.5), I refrain from repetitions. I argue that the measurement areas are large enough ($35\,\mathrm{m}^2$) to smooth out stochastic effects so that running the simulation with different seeds would not change the qualitative behavior that I am interested in.

Key performance indicators

I choose Fruin's Level of Service [69] and the Pedestrian Comfort Level [79] (see Section 2.1.2) as key performance indicators. The density d is measured every $0.4\,\mathrm{s}$ in each measurement area (M1, M2, M3) by dividing the number of agents N by the area size A (classical method): $d = N/A$. To evaluate the Pedestrian Comfort Level [79], the specific flow is derived from the measured densities according to the Kladec formula, see Eq. (2.1). Therefore, flow and density are averaged values. The large size of the measurement area smooths out stochastic differences between different simulation runs (seeds). In addition, the number of lost packets, the packet life time, the number of generated route recommendations, the density measurements provided by the density map application, and the ratio of simulation time and real time are measured, see Tab. 4.18. Importantly, I expect the true density to be underestimated since only $40\,\%$ of the agents use the density map application in the scenario.

Inflow and steady state flow

Two system states are to be assessed within one simulation run: the inflow and a steady state flow. One can expect that the inflow takes the first $220\,\mathrm{s}$, that is the time an agent with average speed ($1.34\,\mathrm{m/s}$) needs to pass the longest route ($300\,\mathrm{m}$). After, the flow settles into a steady state. To observe both states, I choose a simulation duration of $10\,\mathrm{min}$ ($600\,\mathrm{s}$).

Quantity	Serves to evaluate ...
Density (ground truth)	Fruin's Level of Service [69] (safety) and the Pedestrian Comfort Level [79] (comfort); measured every 0.4 s.
Estimated density from app	Suitability of the density map application for providing density data. Measured every 2 s.
Number of route recommendations	Verification of the route recommendation algorithm. I expect that, due to the counter-flow, the long route is continuously recommended. Measured every 2 s.
Ratio of simulation time and real time, RAM usage, CPU usage	Real-time capability of the simulation setup.
Number of lost packets, packet lifetime	Reliability of the information provision through direct communication.
Total number of agents	Separation of inflow phase and steady flow.

Table 4.18.: Overview of key performance indicators and quantities of interest. Several quantities of interest are measured to evaluate the performance and reliability of the proposed crowd guidance system. Also, the computational effort is monitored: the RAM and CPU usage of the three Docker containers (*OMNeT++, flowcontrol, Vadere*).

4.4.4. Proof of concept through simulations

In this sub-section I present the results of the simulation study. I check whether expectations regarding the system behavior are met and look at the key performance to evaluate how the crowd guidance system affects the traffic situation.

Dynamic change: inflow and steady-state flow

Fig. 4.29 depicts the total number of agents over time. As expected, the number of agents increases until 220 s which indicates the inflow state. At around 220 s, the number of agents stagnates, thus, a steady state flow is reached. The maximum number of agents is 136 agents in the setting with guidance. Therefore, a maximum of 54 agents (40 %) communicate over the mobile network at the same time. In the three measurement areas the steady state is reached even earlier because they are located at the beginning of the routes. It is reached after 120 s at the short route, see Fig. 4.30 (top). On the medium and long route, the density keeps oscillating beyond 50 s due to the low number of agents.
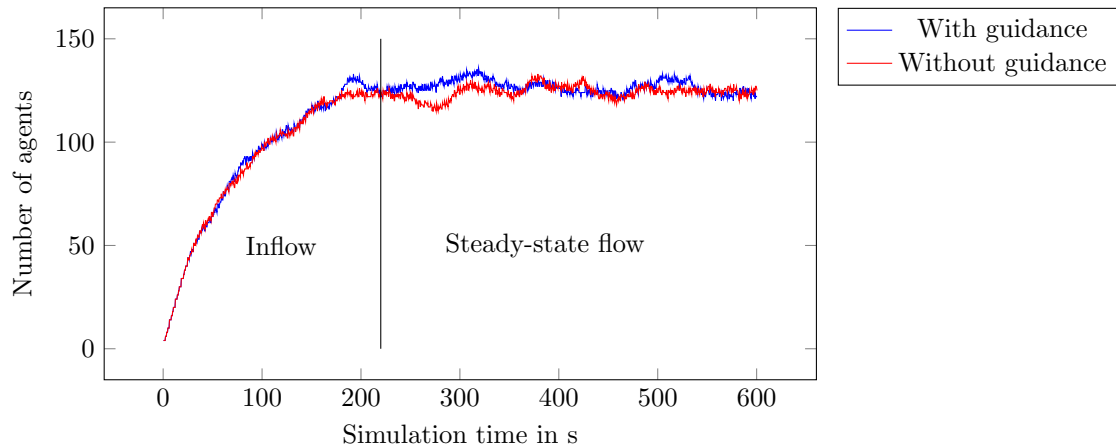
Figure 4.29.: Total number of agents in the topography over time. After 220 s, the number of agents does not change which indicates a steady state flow. The maximum number of agents is 133 (no guidance) and 136 agents (with guidance).

The crowd guidance system improves the level of service and safety

Fig. 4.30 depicts the density without and with crowd guidance for each route. The densities on the short route are always higher than on the medium and long route. At the longer routes, a level of service A is achieved regardless of whether crowd guidance is present or not: In reality pedestrians would walk with their desired speeds. For the short route, a level of service E persists for most of the time indicating that the crowd movement temporarily stops and passing maneuvers are impossible. This constitutes a safety risk. With guidance, the better level D is achieved most of the time with less restricted movement which allows some pedestrians to pass each other. Note that this is the best achievable level in the scenario due to the counter flow of 70 ped/m/min, which corresponds to a level D all by itself. Therefore, I conclude that the proposed guidance system improves the service quality and safety.

Redirecting only a few agents resolves congestion on the critical route

Redirecting only a few pedestrians has a large effect on the congestion situation. With guidance the density on the short route is $0.3\,\mathrm{ped/m^2}$ lower than without guidance, see time slots (1)-(3) in Fig. 4.30. According to the Kladec Eq. (2.1) at least 24 ped/min needed to be redirected to achieve such a decrease of the density.

The simulation, on the other hand, demonstrates that it suffices to redirect only 6 ped/min. This can be explained by the change in the dynamics of the movement behavior as a result of the redirection: The redirection prevents the opposing flows from blocking each

other. Fig. 4.31 depicts how congestion evolves for the system without guidance and how congestion is resolved for the system with guidance.
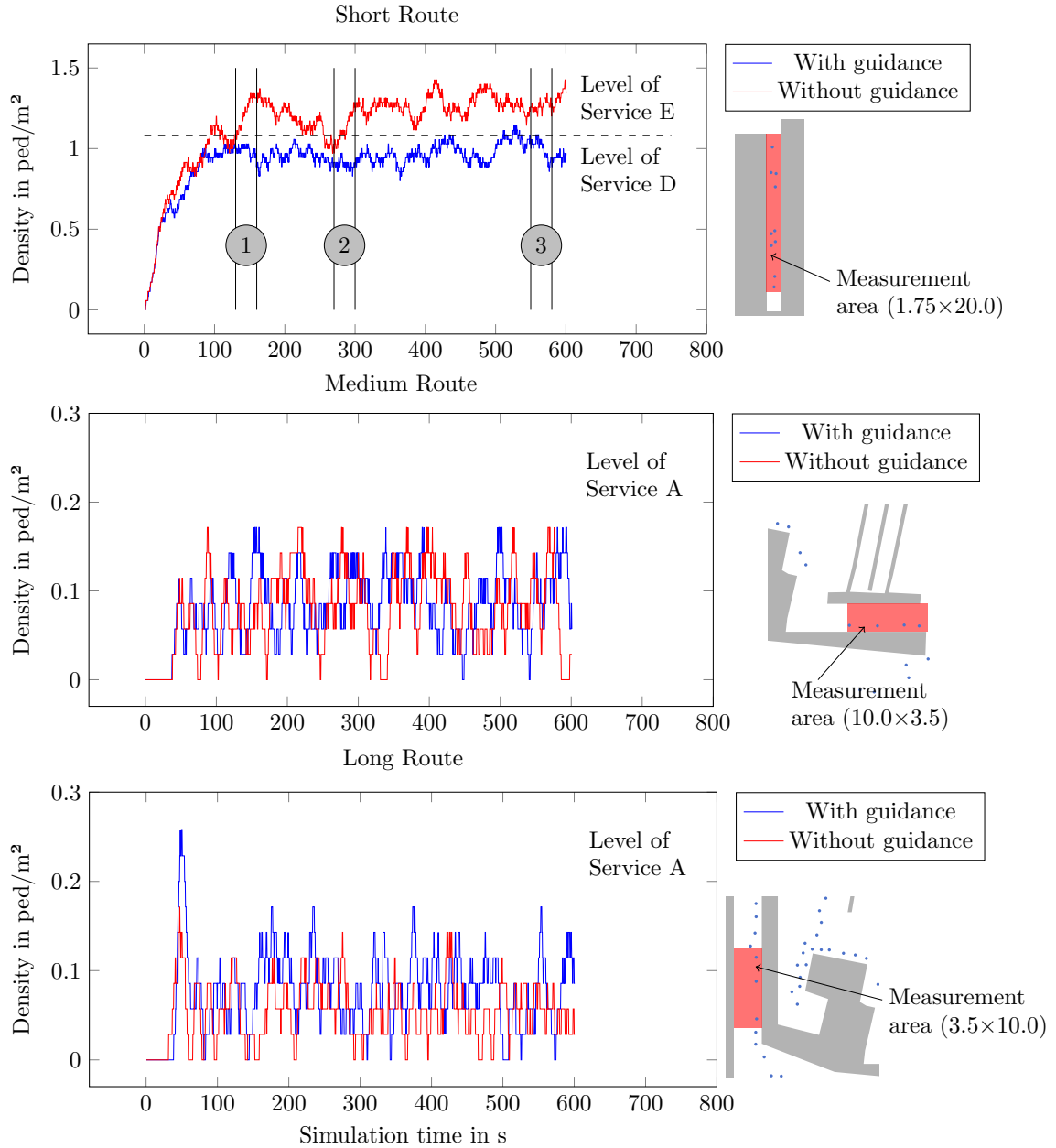
Figure 4.30.: Densities for safety evaluation with and without guidance. Providing route recommendations reduces the density on the short route (top): most of the time a level of service D can be achieved. Congestion is prevented only in the setting with guidance: the density does not increase within the time slots (1),(2),(3). Fig 4.31 depicts the congestion situation for time slot (1).
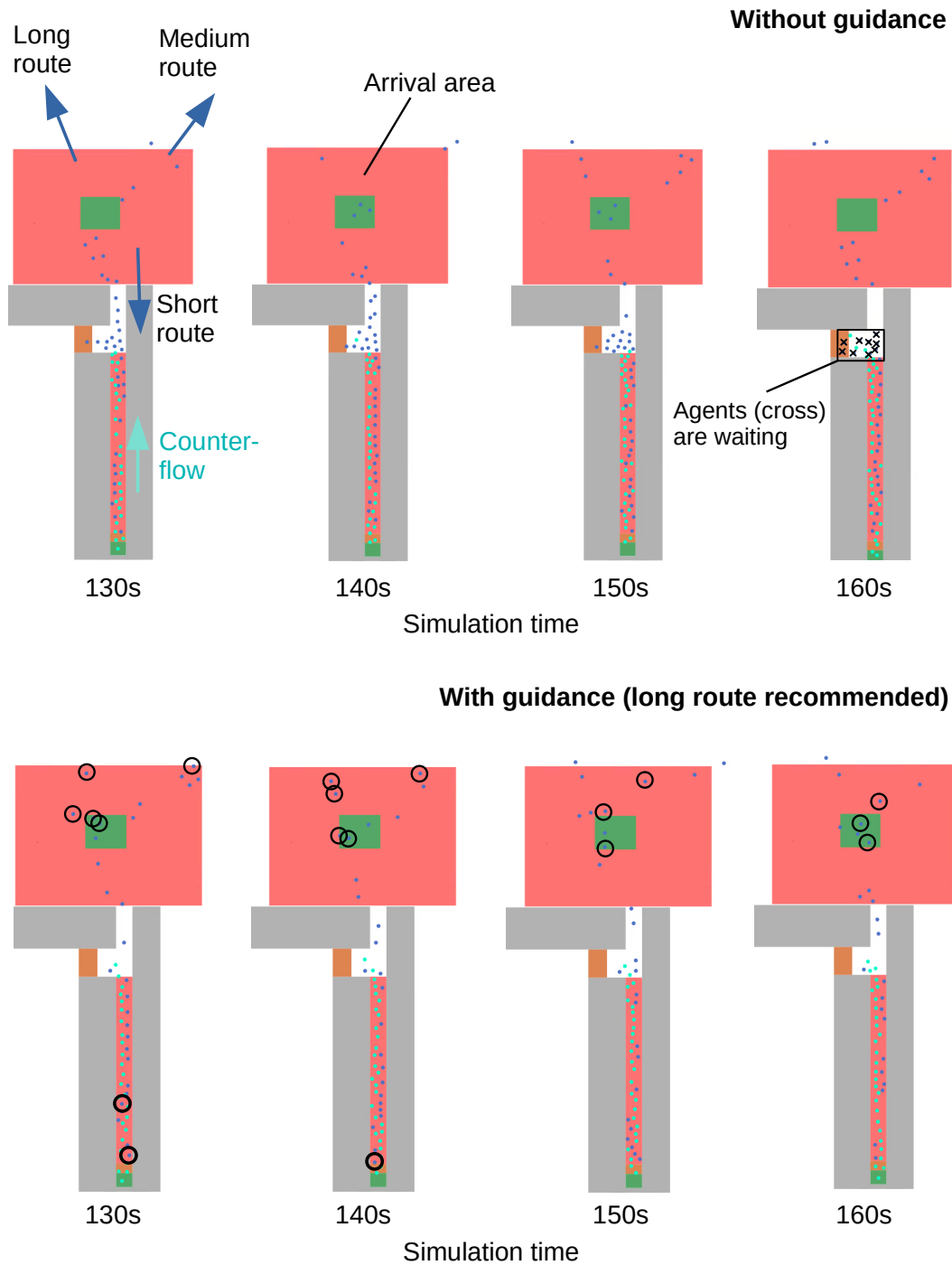
Figure 4.31.: Resolving congestion through redirecting pedestrians. Without guidance, the short route is congested due to the counter-flow: several agents wait in front of the route entrance (see top row). With guidance, the route is less congested (bottom row). Route recommendations are only displayed within the arrival area. Informed agents are marked with a circle: 40 % of them take the recommended long route, 35 % take the medium and 25 % take the short route.

No improvement of comfort due to the counter-flow

To evaluate comfort, the Pedestrian Comfort Level is analyzed. Converting the densities into flows using Eq. (2.1), the flow values are far higher than the recommended threshold of 17 ped/m/min considered as comfortable for such scenarios, see Fig. 4.32. Note, that the threshold value does not take into account that soccer fans, who share a social identity, may perceive proximity and congestion as less uncomfortable than individuals. Nevertheless, the threshold value is exceeded by a factor of four and I expect them to feel uncomfortable. I conclude that the pedestrian guidance system, while it improves safety, does not suffice to ensure travel comfort.
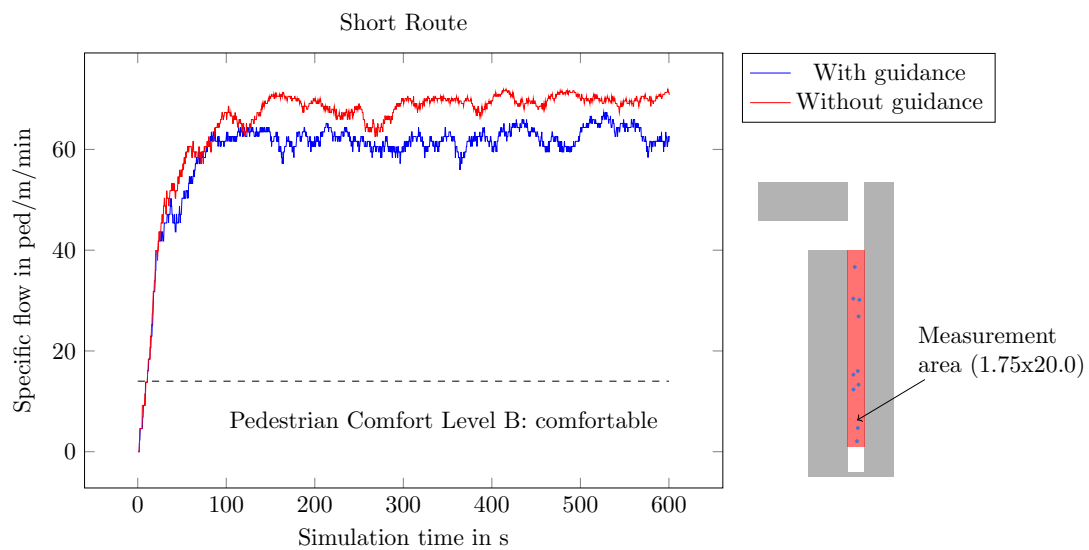
Figure 4.32.: Specific flows at the short route for comfort evaluation. The flows are derived from the density measurements using Eq. (2.1). Due to the counterflow, the pedestrian comfort level B is never reached.

Verification of the route recommendation algorithm

Fig. 4.33 shows the number of route recommendations over time. As expected, the route recommendation algorithm recommends the long route every 2 s once the route recommendation is active because the density on the short route is always higher than on the long route, that is, the route recommendation algorithm works correctly.
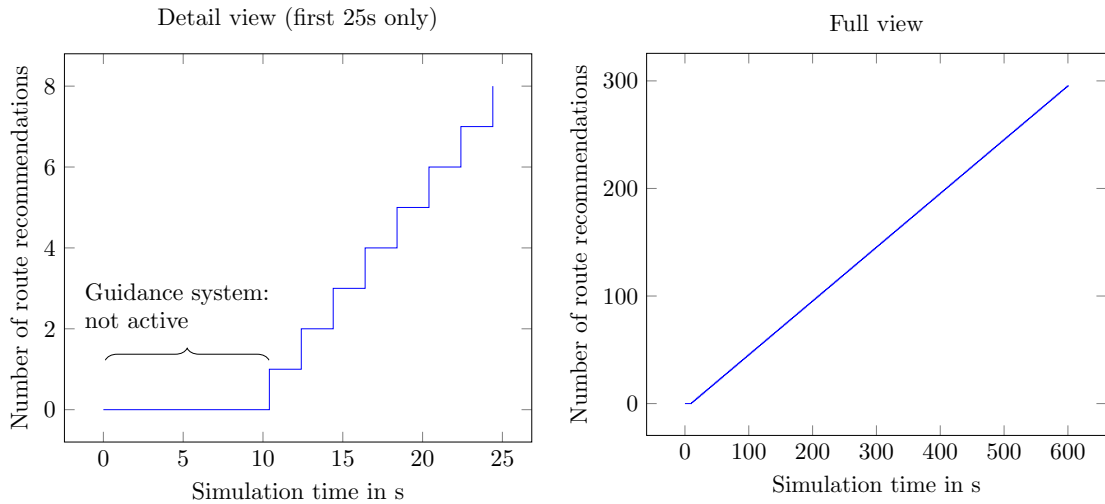
Figure 4.33.: Number of route recommendations. During the first 10s, the guidance system is not active (left). As soon as it starts working, the route recommendation algorithm continuously and correctly recommends the long route due to congestion along the short route (right).

The density map application provides suitable density measurements

In the following, it is evaluated how suitable the decentralized density map application [4] is for providing density measurements. Since the route recommendation is computed in the controller station, one needs to look at the density values of the application of the controller station. As expected, the density is underestimated because only 40 % of the agents share their location, see Fig. 4.34.

The scaled densities, that are, the densities that would be achieved if 100 % of the agents shared their location, are similar to the ground truth: compare the green and black curves in Fig. 4.34. Therefore, relative density differences between routes are measured correctly given that position-sharing agents are evenly distributed over routes. This is promising for future applications, though it would be desirable to capture absolute values more accurately.

Since density differences, not absolute values, drive the algorithm I conclude that the density map application is suitable as data provider for the suggested density-based route recommendation algorithm.
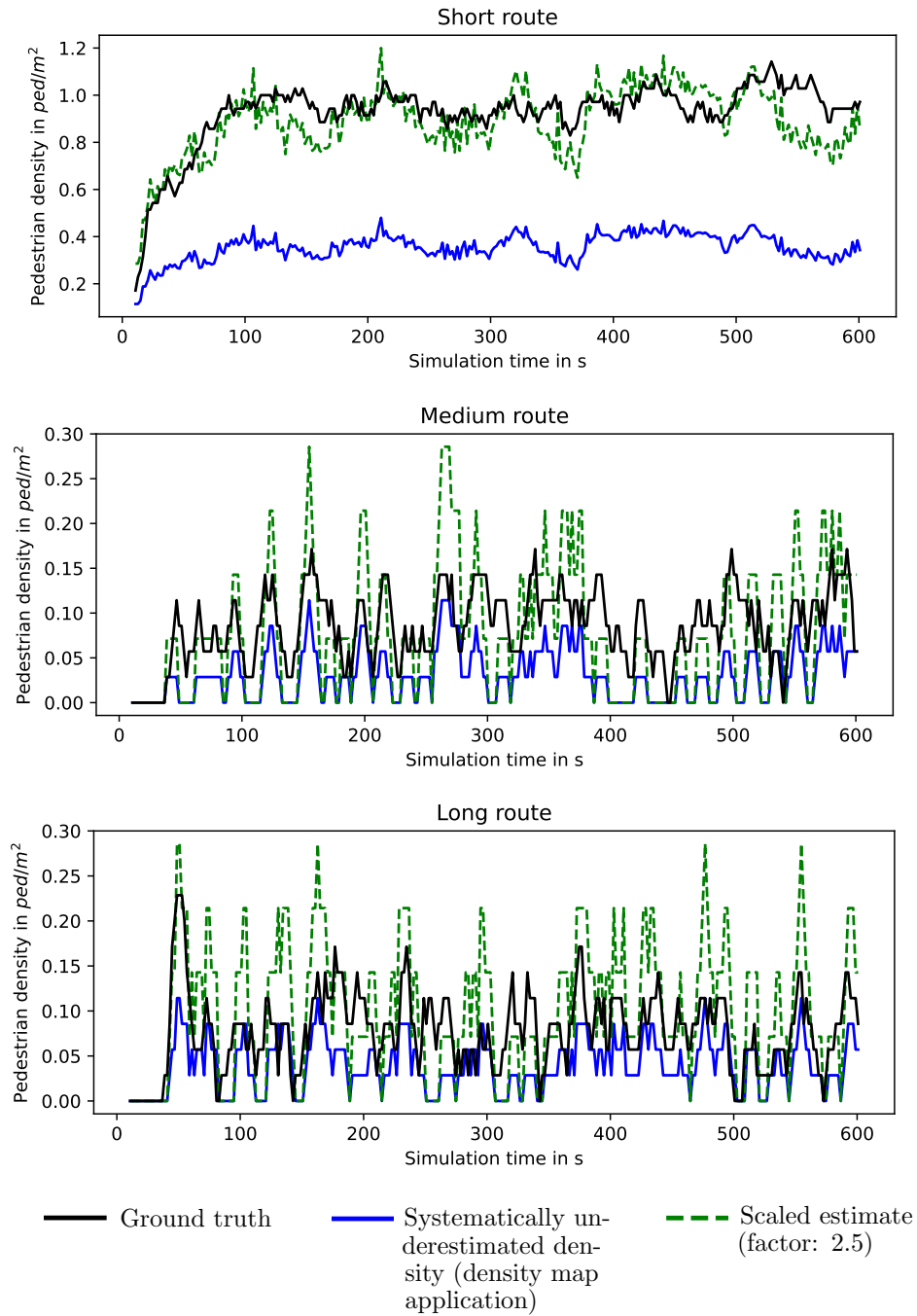
Figure 4.34.: Density estimates from the decentralized density map application (setting with guidance). The blue curve corresponds to the density estimates derived at the controller station. Since only 40 % of the agents share their position and density data, the density is systematically underestimated. The green curve is the same as the blue curve but scaled with a factor 2.5 which represents the case that 100 % of the agents would share their position.

The time between two route recommendations suffices to disseminate information and density data

The chosen value of 2 s for the route recommendation interval requires that information is transmitted in less than 2 s. Otherwise, decisions would be based on outdated data. Therefore it is necessary to look at the duration of the information dissemination. I use the packet lifetime, that is, the time a packet travels from its sender to a receiver over the mobile network. The maximum lifetime of a packet disseminated over the route recommendation application is 27 ms, see Tab. 4.19. There is no packet loss because all of the 13995 packets are received successfully. As the maximum packet lifetime is much smaller than 2 s, I conclude that route recommendations are always received in time. To ensure the the received density information is not obsolete, I look at the packet lifetime and the packet loss of the density map application. The packet loss is 50 % (only 861,419 of 1,751,011 packets arrive) which is high given that only 54 people communicate at the same time. Nevertheless, the maximum packet lifetime is only 203 ms which is again smaller than 2 s. Since both application provide information in less than 2 s, I conclude, that the chosen two seconds interval is suitable.

Packet lifetime in seconds

Application	mean	std	min	max
Route recommendation application	0.021	0.001	0.011	0.028
Density map application	0.026	0.008	0.016	0.265

Table 4.19.: Packet lifetimes. The controller station sends route recommendations over the route recommendation application to agents. Agents and the controller station share density measurements over the density map application. In total, 13,529 packets lifetimes are collected for the route recommendation app, and 861,419 for the density map application from which the above statistics are derived.

4.4.5. Performance evaluation

The performance evaluation is based on the simulation runs executed on a virtual machine with 80 CPUs and 94GB RAM, see Appendix B. One can observe that adding a crowd guidance system to a crowd simulation drastically increases the simulation effort, see Tab. 4.20. With guidance, the simulation took approximately 184 times longer. The reason for this is the extensive mobile communication simulation, which becomes slower the more communication takes place. This can be seen from the increase in the ratio of simulation time to real time during the inflow phase where the number of agents increases, see Fig. 4.35. After simulating 220 s, the ratio of simulation time and real time varies strongly. The lower limit is approximately 25, which means that in the best case the simulation is only 25 times slower than real time. In the worst case, however, the

ratio is 150. It is therefore not possible to simulate events in real time with the given hardware.

Simulation	Simulators	Start	End	Duration
Crowd without guidance	*Vadere*	19:15:54	19:17:14	1.33 min
Crowd guidance system	*Vadere, OMNeT++, flowcontro*l	19:17:14	23:21:33	244.3 min

Table 4.20.: Simulation runtimes measured on 16 Dec 2023. The simulation of the crowd guidance system requires coupling the simulators *Vadere, OMNeT++* and *flowcontrol.* Due to the extensive mobile communication simulation, the simulation with a crowd guidance system based on direct communication takes 184 times (= 224.3/1.33) longer than a crowd simulation without.

Simulating a guidance system also requires careful resource planning, as the RAM usage increases over the simulation time, see Fig. 4.35 (right). This makes the execution of simulation runs in parallel inefficient, as RAM must be reserved.
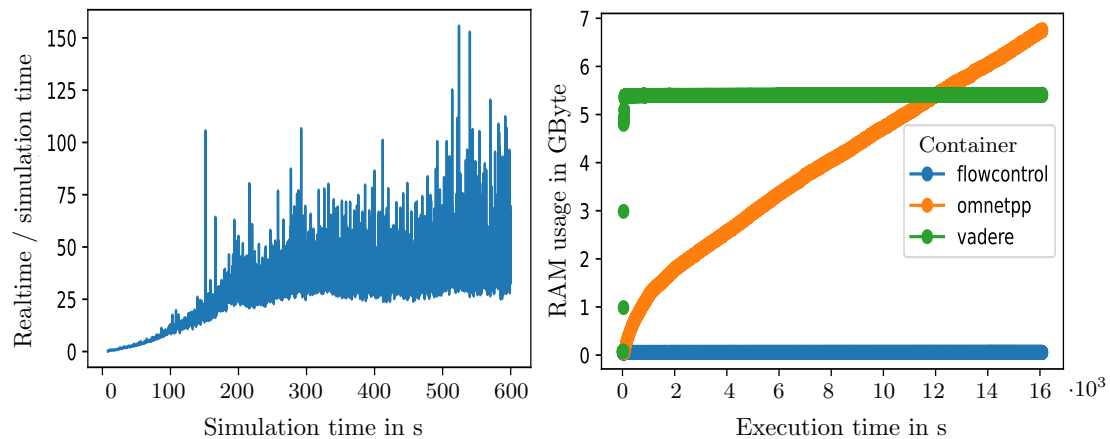
Figure 4.35.: Performance of the simulation with crowd guidance. The simulation is not real time capable: it is up to 150 times slower than the real time process (left). The RAM allocated by the mobile networks simulator *OMNeT++* increases over time (right).

4.4.6. Summary, model validation, limitations

Summary and lessons learned

In this section, I proposed a complete crowd guidance system and simulated it to investigate whether it can improve the congestion situation in a real-life use case. First, I modeled the crowd guidance system by combining previously tested component models into an overall model: a density-based route recommendation algorithm, a crowd model and two mobile applications based on LTE sidelink communication. The first mobile application senses the crowd and the second disseminates route recommendations. The route recommendation algorithm is evaluated every two seconds and recommends the longest route among three routes if and only if the pedestrian density on the congested short route is higher than on the long route.

To test the concept, a scenario before a football match at Münchner Freiheit metro station (Munich, Germany) was modeled where football fans change from the tram or bus to the underground which leads to congestion at the short route. The situation is aggravated by an anti-directional passenger flow. To find suitable parameters for the route choice model, route probabilities were estimated from survey data for which I suggested a novel model that converts the Likert-scaled survey data to route probabilities.

A simulation study was conducted with and without the crowd guidance system. The results demonstrated that the crowd guidance system can improve the safety. Without the system the level of service has a level E and with the redirection it reaches a level D which allows pedestrians to pass each other and thus prevents congestion at a critical level. On the other hand, another key performance indicator, the pedestrian comfort level, did not improve considerably.

From an engineering perspective, it was demonstrated that the system fulfills functional requirements: The results showed that the suggested time interval for the route recommendation algorithm of two seconds is feasible because the information is transmitted much faster over the mobile network.

Importantly, not all people need to use the mobile applications. Partial usage leads to a systematic underestimation of the density. However, this does not have any effect on the route decision because the route recommendation algorithm relies on the density difference between routes. Crucially, it suffices to redirect relatively few people to resolve congestion in the real-life use case. Therefore, I conclude that the proposed crowd guidance system can indeed improve the traffic situation.

Based on these findings I answer my research question:

> **How can crowds be redirected through mobile applications based on direct communication technologies? (RQ)**
>
> The crowd guidance system composed of the following components answers the question: Route recommendations are computed using a density-based algorithm. Densities are measured through a LTE sidelink-based mobile application. The recommendations are disseminated in the crowd via another LTE sidelink-based mobile application.

Model validation

The simulation model is made up of several sub-models that have already been validated against empirical data and simulation data, mainly the Optimal Steps Model for locomotion behavior and the 3GPP LTE channel model for mobile communication. The new route recommendation algorithm has a control function, but does not model a physical system. Thus, as an isolated component, it cannot be checked against measured data. The fact that the main component models have already been validated is a good prerequisite for the validity of the composed model. Nevertheless, the whole system can only be validated once empirical data is available that is when the system has been rolled out in reality. Thus, at this point, validation of the complete system must remain a task for the future.

Limitations and future directions

In the real-life use case, flows differed strongly between the possible routes. Therefore, small fluctuations of the densities had no effect on the route decision. If flows and densities of the routes are more similar, a smoothing procedure may be necessary. On the other hand, the use case is most interesting when a strong preference for one route causes congestion.

The density map application systematically underestimates the density if only a proportion of the virtual pedestrians used the measurement application which is a realistic assumption. In my scenario the underestimation had no effect on the decision of the route algorithm since the app users were evenly disseminated over the routes. If this is not the case, additional measurements and scaling procedures may be necessary.

The parameters of the route choice model were estimated using a simple model that interprets Likert-scaled route attractivenesses from the survey as route frequencies in the long run. The uncertainty of the estimated route choice proportions was not part of the study. It would be interesting to analyze how the uncertainties of the route choice proportions affect the key performance indicators.

Although the number of app users was low, the packet loss was high for the mobile application that measures densities using sidelinks. It would be interesting to analyze how scalable the mobile application is regarding the number of app users and how this affects the accuracy of the density estimates. Also, it would be interesting to test the system in the uncontrolled communication mode without base station.

The simulation of ten minutes real time took more than four hours due to the time-consuming mobile communication simulation. Consequently, parameter studies with many samples were infeasible. To accelerate the simulation, one could replace the mobile communication components by suitable surrogate models.

4.5. Summary of studies

In this chapter I assessed how crowds can be guided through direct communication technologies. Firstly, the suitability of direct communication technology for disseminating detour information in a crowd was investigated using a scenario where obstacles impaired wave propagation (RQ-1). The study was split into two parts: Finding the minimum number of pedestrians in the scenario to overcome shadowing and assessing the information dissemination in the absence of shadowing using forward propagation and global sensitivity analysis. The uncertain parameters 'number of pedestrians', the 'time interval' of the information broadcast, and the 'transmitter power', that served as non-influential control parameter, were propagated to construct polynomial chaos expansions. As expected, the control parameter had no influence. The results showed that the median of the packet lifetime depends only on the time interval of the broadcast. The crowd size slightly affected the minimum and maximum packet lifetimes. The time between two broadcasts was always more influential than the number of agents. The study strongly indicated that, despite interference, detour information was always provided within $10\,\mathrm{s}$. This is sufficiently fast enough to inform a crowd.

Secondly, I investigated the suitability of route recommendation algorithms (RQ-2). Since state-of-the-art algorithms were unsuitable, I proposed three heuristics: One that recommends routes sequentially, one that recommends the route with the lowest density, and an even simpler algorithm that always recommends the longest route among the available routes if the direct short route experienced higher densities. The algorithms were compared in a simulation study for an unknown level of compliance varied between $0\,\%$ and $100\,\%$. The two density based algorithms resolved congestion at lower compliance rates than the sequential procedure. Interestingly the maximum densities did not differ statistically and all algorithms successfully resolved congestion. However, fewer redirection measures were necessary with the simplest algorithm that provides redirection measures if and only if the short route is more congested than the long route.

Thirdly, I investigated the effect of message design on compliance behavior (RQ-3). I developed eight message designs and tested them on over 1400 test subjects in an online

survey that was based on a real-life scenario at the metro station Münchner Freiheit (Germany, Munich). The survey was conducted with two groups: Football fans for which I assumed a shared social identity and a control group of mainly students. The findings indicated that people are willing, in principle, to follow route recommendations provided by an app. However, there was no ideal message design that best fulfilled that task to change their behavior independently of the social group. Messages that appeal to the team spirit of football fans increased their compliance when combined with other message components, but had no effect on students and faculty associates where social identification was lower. Providing real-time information on congestion proved most effective in fostering compliance.

Finally I answered my research question of how crowds can be redirected using direct communication technology. I proposed a crowd guidance system, applied it to the real-life use case Münchner Freiheit, and tested it in a simulation study. The crowd guidance system is composed of a density-based route recommendation algorithm, the crowd model and two mobile applications based on LTE sidelink communication, one for sensing and another one for redirecting the crowd. To find suitable parameters for the route choice model, I estimated route probabilities from the survey data. The simulation study demonstrated that the crowd guidance system can lift the service level from E to D. At the same time comfort remained at a low level. The results also showed that the suggested time interval of two seconds between two evaluations of the route recommendation algorithm is feasible because information is transmitted much faster over the mobile network. Importantly, I could show that if only a proportion of the people uses the sensing application the underestimated densities have no negative effect because the route recommendation algorithm relies on density differences between routes. Crucially, I found that redirecting only relatively few people suffices to resolve congestion. On the whole, I conclude that my crowd guidance system can indeed improve the traffic situation.

5. Summary, future directions, and conclusion

In this chapter, I provide a summary of the thesis (Section 5.1), with a short review of each of the three main chapters. Section 5.2 contains future research directions. Section 5.3 concludes the thesis.

5.1. Summary

The goal of this work was to develop a crowd guidance system to redirect crowds through mobile applications based on direct communication technology. I asked how such a system can be designed, made a proposal, and investigated it using modeling and simulation. I started with analyzing the state of the art. I presented theories on crowd behavior and concluded that people share social identities that often dominate their behavior, which should be harnessed when redirecting crowds. My review of crowd-sensing procedures showed that while various procedures are available, none provide suitable crowd density data based on direct communication technology. Therefore, new procedures were needed. I also looked at algorithms that compute signals to guide crowds based on the current traffic situation. Algorithms were either customized to particular scenarios and, thus, not transferable or were tested under the assumption that people always comply with instructions, which is wrong. From this, I concluded that transferable algorithms were needed which should be tested for an unknown compliance.

Then, I presented models and simulators that can be employed to simulate direct communication in mobile networks based on the IEEE standards 802.11p/bd and the LTE and 5G standards of the 3rd Generation Partnership Project. Next, I looked at model and simulator coupling to build a crowd guidance model from the previously presented component models. Although there are suitable middlewares and interfaces, no simulation framework was found. I concluded that a new simulation framework had to be developed for my investigations. For this purpose, I created the simulation framework *CrowNet*. It was developed in collaboration with scientists from the field of mobile communication to cover the communication aspects. I developed the crucial software modules that enable the simulation of a crowd guidance system on my own; other modules were developed in collaboration, and some modules I only used. I followed software engineering principles and defined scope, requirements, and quality measures. Then, the individual modules were developed. The *crownetutils* module was created to enable the coupling of models and simulators using an explicit update scheme. The *SUQ-controller* module was introduced to conduct parameter studies of coupled simulation in parallel.

The *OMNeT++* module was introduced to provide mobile applications for crowd sensing and disseminating route recommendations based on direct communication technology. The *Vadere* crowd simulation module was integrated to provide models for the crowd behavior with the *flowcontrol* module as route recommendation provider. All modules are extensions of existing frameworks except for the novel framework *flowcontrol* that I created in this work.

In the last step, I used the newly created framework *CrowNet* to answer the main research question of how crowds can be redirected using direct communication technologies. Firstly, I assessed how reliably information is disseminated in a moving crowd. I analyzed how shadowing affects information dissemination, and I found that the information is disseminated within seconds when the crowd is sufficiently dense and well-distributed. Secondly, I compared route recommendation algorithms when the number of pedestrians complying with recommendations is uncertain. I found that algorithms that base recommendations on crowd density differences can compensate low compliance rates well. I suggested that the density-based algorithm that required the fewest redirection measure should be employed in a crowd guidance system. Since compliance depends on how information is presented, I developed message designs and tested them in an online survey with more than 1400 participants to find out which design is most convincing so that people follow a route recommendation. The statistical analyses showed that there is no ideal message design that fits all scenarios. However, real-time congestion information fosters compliance. I also found evidence that social identities affect route choice: Messages that appealed to the team spirit affected football fans' route choice but had no effect on the control group. Finally, I answered my research question by synthesizing the findings from the previous studies. I suggested a crowd guidance system in which the route recommendations computed by a density-based algorithm are disseminated in the crowd using LTE sidelink communication. Densities are also measured using a second sidelink-based mobile application. The concept was applied to a real-life use case at the metro station Münchner Freiheit (Munich, Germany) and tested in a simulation study. The results demonstrated that the system can improve pedestrian traffic by resolving congestion.

5.2. Future directions

One possible direction for future research is the optimization of the distribution of pedestrians. In this thesis, I used heuristic route recommendation algorithms. I proved that, despite their simplicity, they are able to prevent and resolve congestion. However, they cannot guarantee that people are optimally distributed over space, and thus, optimally prevent congestion. The novel simulation framework *flowcontrol* offers a flexible software structure which enables researches to implement and test algorithms for the optimization of the system. Promising candidates may be methods based on model predictive control,

state-space control or Koopman control given a suitable surrogate. Strategies must be found to deal with the uncontrollability of the crowd and the partial observability.

I further suggest investigating the scalability of the system for large-scale scenarios with many crowd members. The results have shown that information about LTE sidelink can be disseminated very quickly in the crowd. Density data was disseminated in approximately the time it takes a person to take a step. However, despite the small crowd size, the packet loss was high. It is unclear how the accuracy of the density estimates evolves when the number of pedestrians increases, and thus, the packet loss. Therefore, further investigations are necessary.

It may also be beneficial to accelerate the simulation to make parameter studies feasible: There are several uncertain parameters whose influence should be analyzed. This is currently not possible because a simulation run takes several hours. To accelerate the simulation, surrogate models could be employed that replace the extensive mobile communication simulation. The challenge is to find a surrogate capturing the influence of the technology used for direct communication and topographical features.

Future research could also study the mobile communication system in the autonomous mode, that is, without resources managed by a base station. Then emergencies with a network infrastructure failure could be handled.

The results of this work are a strong encouragement to implement a crowd guidance system in practice. The simulation model should be validated against empirical data as soon as such a system becomes available.

5.3. Conclusion

Previous crowd management approaches have focused either on technological processes, such as crowd detection, or psychological aspects, such as communication with the crowd. In this work, a complex socio-technical system that combines technology and psychology was proposed and tested for the first time: The crowd is automatically detected and informed using mobile applications while considering the crowd's psychological behavior. The work thus contributes considerably to understanding how complex automated guidance systems for crowds can be built. For the investigations, the *CrowNet* simulator was created, which is a free and open software tool for future research.

The most important finding of this work is that it is indeed possible to redirect crowds using mobile applications based on direct communication technologies. This was demonstrated for a real-life use case in a simulation study under realistic conditions. The proposed system resolved safety-critical jams even when only a part of the crowd was using the technology. This strongly encourages building a real system, making crowd management accessible for places or happenings without infrastructure for conventional crowd management.

Bibliography

The bibliography contains 312 references. It is divided into four parts:

- Articles (total number: 222)
- Books (total number: 28)
- Theses (total number: 19)
- Other sources (total number: 43)

The references are sorted according to their position in the text. It is therefore possible that the number within a part is not consecutive.

Articles

[1]   Milad Haghani and Majid Sarvi. "Crowd behaviour and motion: Empirical methods". In: *Transportation Research Part B: Methodological* 107 (2018), pp. 253–294. ISSN: 0191-2615. DOI: `10.1016/j.trb.2017.06.017`.

[3]   Waqar Anwar et al. "On the Reliability of NR-V2X and IEEE 802.11bd". In: *Conference IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (Sept. 2019), pp. 1–7. DOI: `10.1109/PIMRC.2019.8904104`.

[4]   Stefan Schuhbäck, Lars Wischhof, and Jorg Ott. "Cellular Sidelink Enabled Decentralized Pedestrian Sensing". In: *IEEE Access* 11 (2023), pp. 13349–13369. DOI: `10.1109/access.2023.3242946`.

[9]   Sebastian Seriani and Rodrigo Fernandez. "Pedestrian traffic management of boarding and alighting in metro stations". In: *Transportation Research Part C: Emerging Technologies* 53 (2015), pp. 76–92. DOI: `10.1016/j.trc.2015.02.003`.

[10] Fateh Kaakai, Said Hayat, and Abdellah El Moudni. "A hybrid Petri nets-based simulation model for evaluating the design of railway transit stations". In: *Simulation Modelling Practice and Theory* 15.8 (2007), pp. 935–969. DOI: `10.1016/j.simpat.2007.05.003`.

[11] Anders Johansson and Dirk Helbing. "Pedestrian flow optimization with a genetic algorithm based on Boolean grids". In: *Conference Pedestrian and Evacuation Dynamics 2005* (Jan. 2007), pp. 267–272. DOI: `10.1007/978-3-540-47064-9\_23`.

Bibliography

[12]   Ahmed Abdelghany et al. "Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities". In: *European Journal of Operational Research* 237.3 (Sept. 2014), pp. 1105–1118. DOI: `10.1016/j.ejor.2014.02.054`.

[13]   Min Zhou et al. "Guided crowd evacuation: approaches and challenges". In: *IEEE/CAA Journal of Automatica Sinica* 6.5 (2019), pp. 1081–1094. DOI: `10.1109/JAS.2019.1911672`.

[14]   Guang Zeng et al. "Experimental study on the effect of background music on pedestrian movement at high density". In: *Physics Letters A* 383.10 (2019), pp. 1011–1018. DOI: `10.1016/j.physleta.2018.12.019`.

[15]   C. Feliciani et al. "Experimental Investigation on Information Provision Methods and Guidance Strategies for Crowd Control". In: *Springer Proceedings in Physics* 252 (2020), pp. 61–67. DOI: `10.1007/978-3-030-55973-1_8`.

[16]   C. Feliciani et al. "Efficiently informing crowds - Experiments and simulations on route choice and decision making in pedestrian crowds with wheelchair users". In: *Transportation Research Part C: Emerging Technologies* 114 (2020), pp. 484–503. DOI: `10.1016/j.trc.2020.02.019`.

[17]   Inaki Echeverria-Huarte and Alexandre Nicolas. "Body and mind: Decoding the dynamics of pedestrians and the effect of smartphone distraction by coupling mechanical and decisional processes". In: *Transportation Research Part C Emerging Technologies* 157 (Oct. 2023), p. 104365. DOI: `10.1016/j.trc.2023.104365`.

[18]   K. Jiang et al. "Effects of mobile phone distraction on pedestrians' crossing behavior and visual attention allocation at a signalized intersection: An outdoor experimental study". In: *Accid Anal Prev* (2008), pp. 170–177. DOI: `10.1016/j.aap.2018.03.019`.

[19]   Miguel Ribeiro et al. "Using passive Wi-Fi for community crowd sensing during the COVID-19 pandemic". In: *Journal of Big Data* 10.1 (2023). DOI: `10.1186/s40537-022-00675-3`.

[20]   Emad Felemban, Adil A. Sheikh, and Faisal Karim Shaikh. "MMaPFlow: A Crowd-Sourcing Based Approach for Mapping Mass Pedestrian Flow". In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. Brussels, BEL: ICST, 2014, pp. 311–317. ISBN: 9781631900396. DOI: `10.4108/icst.mobiquitous.2014.257985`.

[21]   Emad Felemban et al. "Inferring Crowd Trip Informatics Using Mobile App in Hajj". In: *International Journal of Computer Science and network security* 20 (Sept. 2020), pp. 76–83. DOI: `10.22937/IJCSNS.2020.20.09.10`.

[22]   Emad Felemban, Adil A. Sheikh, and Atif Naseer. "Improving Response Time for Crowd Management in Hajj". In: *Computers* 10.4 (2021), pp. 1–14. DOI: `10.3390/computers10040046`.

Bibliography

[23] Pegah Torkamandi, Ljubica Pajevic Kärkkäinen, and Jörg Ott. "Characterizing Wi-Fi Probing Behavior for Privacy-Preserving Crowdsensing". In: *Proceedings of the International Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*. ACM, Oct. 2022. DOI: 10.1145/3551659.3559039.

[24] M. Grossglauser and D. Tse. "Mobility increases the capacity of ad-hoc wireless networks". In: *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. 20th Annual Joint Conference of the IEEE Computer and Communications Society*. Vol. 3. 2001, pp. 1360–1369. DOI: 10.1109/INFCOM.2001.916631.

[25] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. "The IMPORTANT framework for analyzing the Impact of Mobility on Performance Of RouTing protocols for Adhoc NeTworks". In: *Ad Hoc Networks* 1 (2003), pp. 383–403. DOI: 10.1016/S1570-8705(03)00040-4.

[26] Ólafur Helgason, Sylvia T. Kouyoumdjieva, and Gunnar Karlsson. "Opportunistic Communication and Human Mobility". In: *IEEE Transactions on Mobile Computing* 13.7 (2014), pp. 1597–1610. DOI: 10.1109/TMC.2013.160.

[27] George R. MacCartney, Theodore S. Rappaport, and Sundeep Rangan. "Rapid Fading Due to Human Blockage in Pedestrian Crowds at 5G Millimeter-Wave Frequencies". In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. IEEE, Dec. 2017. DOI: 10.1109/glocom.2017.8254900.

[28] A. Chaintreau et al. "Impact of Human Mobility on Opportunistic Forwarding Algorithms". In: *IEEE Transactions on Mobile Computing* 6.6 (2007), pp. 606–620. DOI: 10.1109/TMC.2007.1060.

[29] L. Chancay-García et al. "Evaluating and enhancing information dissemination in urban areas of interest using opportunistic networks". In: *IEEE Access* 6 (2018), pp. 32514–32531. DOI: 10.1109/ACCESS.2018.2846201.

[30] Lars Wischhof et al. "On the Influence of Microscopic Mobility in Modelling Pedestrian Communication". In: *Proc. of 7th International Symposium on Ubiquitous Networking (UNet'22)*. Oct. 2022.

[31] Sheng Zhang et al. "Survey on Mobility Model of Opportunistic Networks". In: *Proceedings of the 3rd International Conference on Wireless Communication and Sensor Networks (WCSN 2016)*. Atlantis Press, 2017. DOI: 10.2991/icwcsn-16.2017.98.

[33] Apoorva Shende, Mahendra Singh, and Pushkin Kachroo. "Optimization-Based Feedback Control for Pedestrian Evacuation From an Exit Corridor". In: *IEEE Transactions on Intelligent Transportation Systems* 12 (Dec. 2011), pp. 1167–1176. DOI: 10.1109/TITS.2011.2146251.

[34] Apoorva Shende, Mahendra P. Singh, and Pushkin Kachroo. "Optimal Feedback Flow Rates for Pedestrian Evacuation in a Network of Corridors". In: *IEEE Transactions on Intelligent Transportation Systems* 14.3 (2013), pp. 1053–1066. DOI: 10.1109/TITS.2013.2250965.

Bibliography

[35] Zhe Zhang, Limin Jia, and Yong Qin. "Level-of-Service Based Hierarchical Feedback Control Method of Network-Wide Pedestrian Flow". In: *Mathematical Problems in Engineering* 2016 (Jan. 2016), pp. 1–14. DOI: 10.1155/2016/9617890.

[36] F. Gao et al. "Effect of density control in partially observable asymmetric-exit evacuation under guidance: Strategic suggestion under time delay". In: *Applied Mathematics and Computation* 418 (2022). DOI: 10.1016/j.amc.2021.126838.

[38] Parthasarathy Krishnamurthy, Patrick Carter, and Edward Blair. "Attribute Framing and Goal Framing Effects in Health Decisions". In: *Organizational Behavior and Human Decision Processes* 85.2 (2001), pp. 382–399. DOI: 10.1006/obhd.2001.2962.

[39] Kristel M Gallagher and John A Updegraff. "Health message framing effects on attitudes, intentions, and behavior: A meta-analytic review". In: *Ann. Behav. Med.* 43.1 (Feb. 2012), pp. 101–116. DOI: 10.1007/s12160-011-9308-7.

[40] Holly Carter et al. "Effective Responder Communication Improves Efficiency and Psychological Outcomes in a Mass Decontamination Field Experiment: Implications for Public Behaviour in the Event of a Chemical Incident". In: *PLoS ONE* 9.3 (2014), e89846. DOI: 10.1371/journal.pone.0089846.

[41] Holly Carter et al. "Effective responder communication, perceived responder legitimacy, and group identification predict public cooperation and compliance in a mass decontamination visualization experiment". In: *Journal of Applied Social Psychology* 45.3 (2015), pp. 173–189. DOI: 10.1111/jasp.12286.

[42] Zakary L. Tormala and Richard E. Petty. "Contextual contrast and perceived knowledge: Exploring the implications for persuasion". In: *Journal of Experimental Social Psychology* 43.1 (2007), pp. 17–30. DOI: 10.1016/j.jesp.2005.11.007.

[43] Amira Osman et al. "Information to improve public perceptions of the food and drug administration (FDA's) tobacco regulatory role". In: *International Journal of Environmental Research and Public Health* 15.4 (2018). DOI: 10.3390/ijerph15040753.

[44] Stephen D. Reicher et al. "Core disgust is attenuated by ingroup relations". In: *Proceedings of the National Academy of Sciences* 113.10 (2016), pp. 2631–2635. DOI: 10.1073/pnas.1517027113.

[45] Isabella von Sivers et al. "Humans do not always act selfishly: Social identity and helping in emergency evacuation simulation". In: *The Conference in Pedestrian and Evacuation Dynamics 2014*. Transportation Research Procedia. Delft, The Netherlands, 2014, pp. 585–593. DOI: 10.1016/j.trpro.2014.09.099.

[46] David Novelli et al. "Crowdedness Mediates the Effect of Social Identification on Positive Emotion in a Crowd: A Survey of Two Crowd Events". In: *PLOS ONE* 8.11 (Nov. 2013), pp. 1–7. DOI: 10.1371/journal.pone.0078983.

Bibliography

[47]   Lara Britt Zomer et al. "Managing Crowds: The Possibilities and Limitations of Crowd Information During Urban Mass Events". In: *Planning Support Systems and Smart Cities*. Ed. by Stan Geertman et al. Cham: Springer International Publishing, 2015, pp. 77–97. ISBN: 978-3-319-18368-8. DOI: 10.1007/978-3-319-18368-8_5.

[49]   Juliane Adrian et al. "A Glossary for Research on Human Crowd Dynamics". In: *Collective Dynamics* (2019). DOI: 10.17815/CD.2019.19.

[53]   S. J. Older. "Movement of Pedestrians on Footways in Shopping Streets". In: *Traffic Engineering and Control* 10 (1968), pp. 160–163.

[54]   Dirk Helbing, Anders Johansson, and Habib Zein Al-Abideen. "Dynamics of crowd disasters: An empirical study". In: *Physical Review E* 4.75 (2007), p. 046109. DOI: 10.1103/PhysRevE.75.046109.

[57]   Sebastian Burghardt, Armin Seyfried, and Wolfram Klingsch. "Fundamental diagram and topographical measurements". In: *Transportation Research Part C: Emerging Technologies* 37 (2013), pp. 268–278. ISSN: 0968-090X. DOI: 10.1016/j.trc.2013.05.002.

[58]   Claudio Feliciani and Katsuhiro Nishinari. "Empirical analysis of the lane formation process in bidirectional pedestrian flow". In: *Phys. Rev. E* 94 (3 Sept. 2016), p. 032304. DOI: 10.1103/PhysRevE.94.032304.

[59]   Shuchao Cao et al. "Pedestrian dynamics in single-file movement of crowd with different age compositions". In: *Physical Review E* 94 (July 2016), p. 012312. DOI: 10.1103/PhysRevE.94.012312.

[60]   Xiangxia Ren et al. "Experimental study on elderly pedestrians passing through bottlenecks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12 (2019), p. 123204. DOI: 10.1088/1742-5468/ab4fdf.

[61]   Gayathri Harihara Subramanian, P.S. Karthika, and Anish Verma. "Macroscopic fundamental flow diagrams of a spiritually motivated crowd". In: *Advances in Complex Systems* 24.01 (2021), p. 2150002. DOI: 10.1142/S0219525921500028.

[62]   Ujjal Chattaraj, Armin Seyfried, and Partha Chakroborty. "Comparison of Pedestrian Fundamental Diagram Across Cultures". In: *Advances in Complex Systems* 12.3 (2009), pp. 393–405. DOI: 10.1142/S0219525909002209.

[63]   Shuchao Cao et al. "Dynamic analysis of pedestrian movement in single-file experiment under limited visibility". In: *Communications in Nonlinear Science and Numerical Simulation* 69 (Oct. 2018), pp. 329–342. DOI: 10.1016/j.cnsns.2018.10.007.

[64]   Rui Ye et al. "Pedestrian single-file movement on stairs under different motivations". In: *Physica A: Statistical Mechanics and its Applications* 571 (2021), p. 125849. DOI: 10.1016/j.physa.2021.125849.

Bibliography

[65]   Young-Jae Kim, Chanam Lee, and Jun-Hyun Kim. "Sidewalk landscape structure and thermal conditions for child and adult pedestrians". In: *Int. J. Environ. Res. Public Health* 15.1 (2018), p. 148.

[66]   Sarah Paetzke, Maik Boltes, and Armin Seyfried. "Influence of Gender Composition in Pedestrian Single-File Experiments". In: *Applied Sciences* 13.9 (2023). ISSN: 2076-3417. DOI: 10.3390/app13095450.

[67]   Masamitsu Mori and Hiroshi Tsukaguchi. "A new method for evaluation of level of service in pedestrian facilities". In: *Transportation Research Part A: General* 21.3 (1987), pp. 223–234. DOI: 10.1016/0191-2607(87)90016-1.

[70]   W. H. K. Lam, J. F. Morrall, and H. Ho. "Pedestrian flow characteristics in Hong Kong". In: *Transportation Research Record* 1487 (1995), p. 56.

[71]   F. P. Navin and R. J. Wheeler. "Pedestrian flow characteristics". In: *Traffic Engineering* 19.7 (1969), pp. 30–33.

[74]   Andreas Schadschneider et al. "Evacuation Dynamics: Empirical Results, Modeling and Applications". English. In: *Encyclopedia of Complexity and Systems Science.* Ed. by Robert A. Meyers. New York: Springer, 2009, pp. 3142–3176. ISBN: 978-0-387-75888-6. DOI: 10.1007/978-0-387-30440-3_187.

[75]   Jun Zhang et al. "Transitions in pedestrian fundamental diagrams of straight corridors and T-junctions". In: *Journal of Statistical Mechanics: Theory and Experiment* 2011.06 (2011), P06004. DOI: 10.1088/1742-5468/2011/06/P06004.

[77]   Varsha Yadav and Rajat Rastogi. "Pedestrian Level of Service: A Review of Factors and Methodology". In: *Lecture Notes in Civil Engineering* 377 (2024), pp. 1–18. DOI: 10.1007/978-981-99-4464-4_1.

[81]   Charles V. Zegeer et al. "Index for assessing pedestrian safety at intersections". In: *Transportation Research Record* 1982 (2006), pp. 76–83. DOI: 10.3141/1982-11.

[82]   Alfonso Montella et al. "Systemic approach to improve safety of urban unsignalized intersections: Development and validation of a Safety Index". In: *Accident Analysis and Prevention* 141 (2020). DOI: 10.1016/j.aap.2020.105523.

[83]   A. Muthu Krishnan and Sankaran Marisamynathan. "Development of pedestrian crossing behavior and safety index models at signalized intersections under mixed traffic conditions". In: *Innovative Infrastructure Solutions* 8.8 (2023). DOI: 10.1007/s41062-023-01183-w.

[90]   Richard A. Berk. "A Gaming Approach to Crowd Behavior". In: *American Sociological Review* 39.3 (1974), pp. 355–373.

[93]   Ian Donald and David Canter. "Intentionality and fatality during the King's Cross underground fire". In: *European Journal of Social Psychology* 22.3 (1992), pp. 203–218. DOI: 10.1002/ejsp.2420220302.

[94]   John C. Turner et al. "Self and Collective: Cognition and Social Context". In: *Personality and Social Psychology Bulletin* 20 (1994), pp. 454–463.

[95]   Henri Tajfel. "Social identity and intergroup behaviour". In: *Social Science Information* 13.2 (1974), pp. 65–93. DOI: 10.1177/053901847401300204.

[96]   Henri Tajfel and John C. Turner. "An integrative theory of intergroup conflict". In: *The social psychology of intergroup relations* (1979). Ed. by W. G. Austin and S. Worchel, pp. 33–47.

[97]   Henri Tajfel. "Social Psychology of Intergroup Relations". In: *Annual Review of Psychology* 33 (1982), pp. 1–39. DOI: 10.1146/annurev.ps.33.020182.000245.

[99]   John C. Turner. "Towards a cognitive redefinition of the social group." In: *Cahiers de Psychologie Cognitive/Current Psychology of Cognition* 1.2 (1981), pp. 93–118. URL: http://psycnet.apa.org/psycinfo/1982-12053-001.

[100]   John C. Turner. "Towards a cognitive redefinition of the social group". In: *Social identity and intergroup relations*. Ed. by Henri Tajfel. Cambridge: Cambridge University Press, 1982, pp. 15–40.

[101]   John C. Turner. "Rediscovering the social group: A self-categorization theory". In: *Rediscovering the social group: A self-categorization theory* (1987). Ed. by John C. Turner, pp. 68–88.

[102]   Anne Templeton and Fergus Neville. "Modeling Collective Behaviour: Insights and Applications from Crowd Psychology". In: (2020), pp. 55–81. DOI: 10.1007/978-3-030-50450-2_4.

[103]   Mark Levine et al. "Identity and emergency intervention: how social group membership and inclusiveness of group boundaries shape helping behavior." In: *Personality and social psychology bulletin* 31.4 (2005), pp. 443–453. DOI: 10.1177/0146167204271651.

[104]   Anne Templeton, John Drury, and Andrew Philippides. "Walking together: behavioural signatures of psychological crowds". In: *Royal Society Open Science* (2018). DOI: 10.1098/rsos.180172.

[105]   Andrea Gorrini, Stefania Bandini, and Majid Sarvi. "Group Dynamics in Pedestrian Crowds: Estimating Proxemic Behavior". In: *Transportation Research Record* 2421.1 (2014), pp. 51–56. DOI: 10.3141/2421-06.

[106]   Gerta Köster et al. "On modelling the influence of group formations in a crowd". In: *Contemporary Social Science* 6.3 (2011), pp. 397–414. DOI: 10.1080/21582041.2011.619867.

[107]   Mehdi Moussaïd et al. "The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics". In: *PLoS ONE* 5.4 (2010), e10047. DOI: 10.1371/journal.pone.0010047.

[108]   Fasheng Qiu and Xiaolin Hu. "Modeling group structures in pedestrian crowd simulation". In: *Simulation Modelling Practice and Theory* 18.2 (2010), pp. 190–205. DOI: 10.1016/j.simpat.2009.10.005.

[109] Gayathri Harihara Subramanian, Nipun Choubey, and Ashish Verma. "Modelling and simulating serpentine group behaviour in crowds using modified social force model". In: *Physica A: Statistical Mechanics and its Applications* 604 (2022), p. 127674. DOI: 10.1016/j.physa.2022.127674.

[111] Barbara Reynolds and Matthew Seeger. "Crisis and Emergency Risk Communication as an Integrative Model". In: *Journal of Health Communication* 10.1 (2005), pp. 43–55. DOI: 10.1080/10810730590904571.

[112] J. Drury et al. "Cooperation versus competition in a mass emergency evacuation: A new laboratory simulation and a new theoretical model". In: *Behavior Research Methods* 41.3 (2009), pp. 957–970. DOI: 10.3758/BRM.41.3.957.

[113] S. Reicher et al. "An integrated approach to crowd psychology and public order policing". In: *Policing* 27.4 (2004), pp. 558–572. DOI: 10.1108/13639510410566271.

[114] A.C. Farr et al. "Wayfinding: A simple concept, a complex process". In: *Transport Reviews* 32.6 (2012), pp. 715–743. DOI: 10.1080/01441647.2012.712555.

[115] P. Palttala et al. "Communication Gaps in Disaster Management: Perceptions by Experts from Governmental and Non-Governmental Organizations". In: *Journal of Contingencies and Crisis Management* 20.1 (2012), pp. 2–12. DOI: 10.1111/j.1468-5973.2011.00656.x.

[116] C. Natalie van der Wal et al. "Examining Evacuee Response to Emergency Communications with Agent-Based Simulations". In: *Sustainability* 13.9 (2021). DOI: 10.3390/su13094623.

[117] Valentina Carfora, Maria Morandi, and Patrizia Catellani. "The Effect of Message Framing in Promoting the Mediterranean Diet: The Moderating Role of Eating Self-Efficacy". In: *Foods* 11.10 (2022). DOI: 10.3390/foods11101454.

[118] Karigan P. Capps et al. "Field Experiment of Signs Promoting Hand Hygiene During the COVID-19 Pandemic". In: *Health Psychology* (2022). DOI: 10.1037/hea0001211.

[119] N. Nobel. "Interplay between benefit appeal and valence framing in reducing smoking behavior: Evidence from a field experience". In: *Behavioral Decision Making* (2022). DOI: 10.1002/bdm.2301.

[120] Thiago Teixeira, Gershon Dublon, and Andreas Savvides. "A Survey of Human-Sensing: Methods for Detecting Presence, Count, Location, Track, and Identity". In: *ACM Computing Surveys* 5 (Jan. 2010), p. 59.

[121] Antonio Brunetti et al. "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey". In: *Neurocomputing* 300 (2018), pp. 17–33. DOI: 10.1016/j.neucom.2018.01.092.

[122] Sylvia T. Kouyoumdjieva, Peter Danielis, and Gunnar Karlsson. "Survey of Non-Image-Based Approaches for Counting People". In: *IEEE Communications Surveys & Tutorials* 22.2 (2020), pp. 1305–1336. DOI: 10.1109/COMST.2019.2902824.

[123]   Kaishun Wu et al. "FILA: Fine-grained indoor localization". In: *2012 Proceedings IEEE INFOCOM*. 2012, pp. 2210–2218. DOI: 10.1109/INFCOM.2012.6195606.

[124]   Jeong Woo Choi, Dae Hyeon Yim, and Sung Ho Cho. "People Counting Based on an IR-UWB Radar Sensor". In: *IEEE Sensors Journal* 17.17 (2017), pp. 5717–5727. DOI: 10.1109/JSEN.2017.2723766.

[125]   Alessandro Olivo, Giulio Maternini, and Benedetto Barabino. "Empirical Study on the Accuracy and Precision of Automatic Passenger Counting in European Bus Services". In: *The Open Transportation Journal* 13 (Dec. 2019), pp. 250–260. DOI: 10.2174/1874447801913010250.

[126]   Wenjun Lei et al. "Study on the relationship between the CO2 concentration and pedestrian flow in a building evacuation passageway". In: *Indoor and Built Environment* 30.9 (2021), pp. 1411–1428. DOI: 10.1177/1420326X20940368.

[127]   Alex Leykin and Riad Hammoud. "Pedestrian tracking by fusion of thermal-visible surveillance videos". In: *Machine Vision and Applications* 21 (2010), pp. 587–595. DOI: 10.1007/s00138-008-0176-5.

[128]   Tom Nicolai and Holger Kenn. "About the Relationship between People and Discoverable Bluetooth Devices in Urban Environments". In: *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*. Mobility '07. Singapore: Association for Computing Machinery, 2007, pp. 72–78. ISBN: 9781595938190. DOI: 10.1145/1378063.1378076.

[129]   Cahyani Wulan Hidayat et al. "Implementation of RFID as tracking for hajj pilgrims connected with GPS". In: *AIP Conference Proceedings* 2088.1 (Mar. 2019), p. 020032. ISSN: 0094-243X. DOI: 10.1063/1.5095284.

[130]   Jens Weppner et al. "Participatory Bluetooth Scans Serving as Urban Crowd Probes". In: *IEEE Sensors Journal* 14.12 (2014), pp. 4196–4206. DOI: 10.1109/JSEN.2014.2360123.

[131]   Mathias Versichele et al. "The Use of Bluetooth for Analysing Spatiotemporal Dynamics of Human Movement at Mass Events: A Case Study of the Ghent Festivities". In: *Applied Geography* 32 (Mar. 2012), pp. 208–220. DOI: 10.1016/j.apgeog.2011.05.011.

[132]   Jeroen van den Heuvel et al. "Using Bluetooth to Estimate the Impact of Congestion on Pedestrian Route Choice at Train Stations". In: *Traffic and Granular Flow '13*. Jan. 2015, pp. 73–82. ISBN: 978-3-319-10628-1. DOI: 10.1007/978-3-319-10629-8_9.

[133]   Michele Nitti et al. "iABACUS: A Wi-Fi-Based Automatic Bus Passenger Counting System". In: *Energies* 13.6 (2020). DOI: 10.3390/en13061446.

[134]   Mario Vega-Barbas et al. "AFOROS: A Low-Cost Wi-Fi-Based Monitoring System for Estimating Occupancy of Public Spaces". In: *Sensors* 21.11 (2021), p. 3863. DOI: 10.3390/s21113863.

[135] Kyosuke Shibata and Hiroshi Yamamoto. "People Crowd Density Estimation System using Deep Learning for Radio Wave Sensing of Cellular Communication". In: *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* (2019), pp. 143–148.

[136] Simone Di Domenico et al. "LTE-based passive device-free crowd density estimation". In: *2017 IEEE International Conference on Communications (ICC)*. 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7997194.

[137] Peter Danielis, Sylvia T. Kouyoumdjieva, and Gunnar Karlsson. "UrbanCount: Mobile crowd counting in urban environments". In: *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 2017, pp. 640–648. DOI: 10.1109/IEMCON.2017.8117189.

[138] Donatella Darsena et al. "Sensing Technologies for Crowd Management, Adaptation, and Information Dissemination in Public Transportation Systems: A Review". In: *IEEE Sensors Journal* 23.1 (2023), pp. 68–87. DOI: 10.1109/JSEN.2022.3223297.

[139] Tal Hassner, Yossi Itcher, and Orit Kliper-Gross. "Violent flows: Real-time detection of violent crowd behavior". In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2012, pp. 1–6. DOI: 10.1109/CVPRW.2012.6239348.

[140] Benjamin Sebastian Bergner. "The Measurement of Stress at Open-Air Events: Monitoring Emotion and Motion Utilizing Wearable Sensor Technology". In: *Traffic and Granular Flow 2019*. Ed. by Iker Zuriguel, Angel Garcimartin, and Raul Cruz. Cham: Springer International Publishing, 2020, pp. 11–19. DOI: 10.1007/978-3-030-55973-1_2.

[141] Mirza Waqar Baig et al. "Crowd Emotion Detection Using Dynamic Probabilistic Models". In: (2014). Ed. by Angel P. del Pobil et al., pp. 328–337. DOI: 10.1007/978-3-319-08864-8_32.

[142] Arnav Sinha, Shriya Padhi, and Sajeeda Shikalgar. "A Survey and Analysis of Crowd Anomaly Detection Techniques". In: *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. 2021, pp. 1500–1504. DOI: 10.1109/ICICV50876.2021.9388400.

[143] Sonu Lamba and Neeta Nain. "Crowd Monitoring and Classification: A Survey". In: *Advances in Computer and Computational Sciences*. Ed. by Sanjiv K. Bhatia et al. Springer Singapore, 2017, pp. 21–31. DOI: 10.1007/978-981-10-3770-2_3.

[144] Sadegh Mohammadi et al. "Physics-Inspired Models for Detecting Abnormal Behaviors in Crowded Scenes". In: *Group and Crowd Behavior for Computer Vision*. Ed. by Vittorio Murino et al. Academic Press, 2017, pp. 253–272. ISBN: 978-0-12-809276-7. DOI: 10.1016/B978-0-12-809276-7.00013-8.

[145] Yue Liu et al. "Abnormal Crowd Behavior Detection Based on Predictive Neural Network". In: *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (Mar. 2019), pp. 221–225. DOI: 10.1109/ICAICA.2019.8873488.

[146] Mariska van Essen et al. "From user equilibrium to system optimum: a literature review on the role of travel information, bounded rationality and non-selfish behaviour at the network and individual levels". In: *Transport Reviews* 36.4 (2016), pp. 527–548. DOI: 10.1080/01441647.2015.1125399.

[147] Toshiki Sato, Tomoko Izumi, and Yoshio Nakatani. "Tourist Evacuation Guidance Support System for Use in Disasters". In: *International Conference on Human-Computer Interaction* (June 2014), pp. 494–501. DOI: 10.1007/978-3-319-07227-2_47.

[148] H. Ren, Y. Yan, and F. Gao. "Variable guiding strategies in multi-exits evacuation: Pursuing balanced pedestrian densities". In: *Applied Mathematics and Computation* 397 (2021). DOI: 10.1016/j.amc.2021.125965.

[149] M. A. Lopez-Carmona and A. P. Garcia. "CellEVAC: An adaptive guidance system for crowd evacuation through behavioral optimization". In: *Safety Science* 139 (2021), p. 105215. DOI: 10.1016/j.ssci.2021.105215.

[150] Nicholas Molyneaux, Riccardo Scarinci, and Michel Bierlaire. "Design and analysis of control strategies for pedestrian flows". In: *Transportation* (May 2020). DOI: 10.1007/s11116-020-10111-1.

[151] Serge P. Hoogendoorn and Piet H. L. Bovy. "Pedestrian route-choice and activity scheduling theory and models". In: *Transportation Research Part B: Methodological* 38.2 (2004), pp. 169–190. DOI: 10.1016/S0191-2615(03)00007-9.

[153] Milad Haghani and Majid Sarvi. "Crowd model calibration at strategic, tactical, and operational levels: Full-spectrum sensitivity analyses show bottleneck parameters are most critical, followed by exit-choice-changing parameters". In: *Transportation Letters* 0.0 (2023), pp. 1–28. DOI: 10.1080/19427867.2023.2195729.

[157] André Borrmann et al. "Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models". In: *Safety Science* 50 (2012), pp. 1695–1703. DOI: 10.1016/j.ssci.2011.12.021.

[158] Peter G. Gipps and Bertil S. Marksjö. "A micro-simulation model for pedestrian flows". In: *Mathematics and Computers in Simulation* 27.2–3 (1985), pp. 95–105. DOI: 10.1016/0378-4754(85)90027-8.

[159] Ekaterina Kirik, Tat'yana Yurgel'yan, and Dmitriy Krouglov. "An intelligent floor field cellular automation model for pedestrian dynamics". In: *Proceedings of the 2007 Summer Computer Simulation Conference*. SCSC '07. San Diego, California, 2007, 21:1–21:6. ISBN: 1-56555-316-0. URL: http://dl.acm.org/citation.cfm?id=1357910.1358127.

[160] Andreas Schadschneider. "Cellular Automaton Approach to Pedestrian Dynamics - Theory". In: *Pedestrian and Evacuation Dynamics*. Ed. by Michael Schreckenberg and Som Deo Sharma. Springer, 2001, pp. 75–86.

[161] Michael J. Seitz and Gerta Köster. "Natural discretization of pedestrian movement in continuous space". In: *Physical Review E* 86.4 (2012), p. 046108. DOI: 10.1103/PhysRevE.86.046108.

[162] Michael J. Seitz, Nikolai W. F. Bode, and Gerta Köster. "How cognitive heuristics can explain social interactions in spatial movement". In: *Journal of the Royal Society Interface* 13.121 (2016), p. 20160439. DOI: 10.1098/rsif.2016.0439.

[165] Gerta Köster and Benedikt Zönnchen. "Queuing at Bottlenecks Using a Dynamic Floor Field for Navigation". In: *The Conference in Pedestrian and Evacuation Dynamics 2014*. Transportation Research Procedia. Delft, The Netherlands, 2014, pp. 344–352. DOI: 10.1016/j.trpro.2014.09.029.

[166] Isabella von Sivers and Gerta Köster. "Dynamic Stride Length Adaptation According to Utility And Personal Space". In: *Transportation Research Part B: Methodological* 74 (2015), pp. 104–117. DOI: 10.1016/j.trb.2015.01.009.

[167] Antoine Tordeux and Armin Seyfried. "Collision-free nonuniform dynamics within continuous optimal velocity models". In: *Physical Review E* 90 (4 2014), p. 042812. DOI: 10.1103/PhysRevE.90.042812.

[168] Ioannis Karamouzas et al. "A predictive collision avoidance model for pedestrian simulation". In: *Motion in Games: Second International Workshop, MIG 2009, Zeist, The Netherlands, November 21–24, 2009, Proceedings*. Springer, 2009, pp. 41–52. DOI: 10.1007/978-3-642-10347-6_4.

[169] K. Hirai and K. Tarui. "A simulation of the behavior of a crowd in panic". In: *Proc. of the 1975 International Conference on Cybernetics and Society*. 1975, p. 409.

[170] Dirk Helbing and Péter Molnár. "Social Force Model for pedestrian dynamics". In: *Physical Review E* 51.5 (1995), pp. 4282–4286. DOI: 10.1103/PhysRevE.51.4282.

[171] Juan Chen, Jian Ma, and S. M. Lo. "Geometric constraint based pedestrian movement model on stairways". In: *Statistical Mechanics and its Applications* 505 (2018), pp. 1212–1230. DOI: 10.1016/j.physa.2018.03.051.

[172] Gerta Köster, Franz Treml, and Marion Gödel. "Avoiding numerical pitfalls in social force models". In: *Physical Review E* 87.6 (2013), p. 063305. DOI: 10.1103/PhysRevE.87.063305.

[173] Felix Dietrich and Gerta Köster. "Gradient navigation model for pedestrian dynamics". In: *Physical Review E* 89.6 (2014), p. 062801. DOI: 10.1103/PhysRevE.89.062801.

[174] Gianluca Antonini, Michel Bierlaire, and Mats Weber. "Discrete choice models of pedestrian walking behavior". In: *Transportation Research Part B: Methodological* 40.8 (2006), pp. 667–687. DOI: 10.1016/j.trb.2005.09.006.

[175] Inhi Kim, Ronald Galiza, and Luis Ferreira. "Modeling pedestrian queuing using micro-simulation". In: *Transportation Research Part A: Policy and Practice* 49.0 (2013), pp. 232–240. DOI: 10.1016/j.tra.2013.01.018.

[177] Michael Seitz, Gerta Köster, and Dirk Hartmann. "On modeling the separation and reunion of social groups". In: *Proceedings of the International Conference on Emergency Evacuation of People from Buildings*. Warzaw, Poland, 2011.

[178] David Boyce. "Network Equilibrium Models for Urban Transport". In: *Handbook of Regional Science* (2021), pp. 247–275. DOI: 10.1007/978-3-662-60723-7_45.

[179] John Glen Wardrop. "Some theoretical Aspects of road traffic research". In: *Proceedings of the Institute of Civil Engineers, Part II* 1 (1952), pp. 325–378. DOI: 10.1680/ipeds.1952.11362.

[181] Alexandra Beaulieu and Bilal Farooq. "A dynamic mixed logit model with agent effect for pedestrian next location choice using ubiquitous Wi-Fi network data". In: *International Journal of Transportation Science and Technology* 8.3 (2019), pp. 280–289. DOI: 10.1016/j.ijtst.2019.02.003.

[182] André de Palma, Robin Lindsey, and Nathalie Picard. "Risk Aversion, the Value of Information, and Traffic Equilibrium". In: *Transportation Science* 46 (Feb. 2012). DOI: 10.2307/41432822.

[183] Luca Crociani et al. "Route choice in pedestrian simulation: Design and evaluation of a model based on empirical observations". In: *Intelligenza Artificiale* 10 (Dec. 2016), pp. 163–182. DOI: 10.3233/IA-160102.

[184] Max Kinateder, Brittany Comunale, and William H. Warren. "Exit choice in an emergency evacuation scenario is influenced by exit familiarity and neighbor behavior". In: *Safety Science* 106 (2018), pp. 170–175. DOI: https://doi.org/10.1016/j.ssci.2018.03.015.

[185] Zhuangbin Shi et al. "Understanding passenger route choice behavior under the influence of detailed route information based on smart card data". In: *Transportation* (2023). DOI: 10.1007/s11116-023-10432-x.

[187] Jie Xu et al. "Route Choice in Subway Station during Morning Peak Hours: A Case of Guangzhou Subway". In: *Discrete Dynamics in Nature and Society* 2015 (Mar. 2015), pp. 1–8. DOI: 10.1155/2015/151434.

[188] G. de Moraes Ramos et al. "Route choice behaviour and travel information in a congested network: Static and dynamic recursive models". In: *Transportation Research Part C: Emerging Technologies* 114 (2020), pp. 681–693. DOI: 10.1016/j.trc.2020.02.014.

[189] Yunhe Tong and Nikolai W.F. Bode. "The value pedestrians attribute to environmental information diminishes in route choice sequences". In: *Transportation Research Part C: Emerging Technologies* 124 (2021), p. 102909. DOI: 10.1016/j.trc.2020.102909.

[190] Sara Arganda, Alfonso Pérez-Escudero, and Gonzalo G. de Polavieja. "A common rule for decision making in animal collectives across species". In: *Proceedings of the National Academy of Sciences* 109.50 (2012), pp. 20508–20513. DOI: 10.1073/pnas.1210664109.

[191] Yunhe Tong and Nikolai W.F. Bode. "Higher investment levels into pre-planned routes increase the adherence of pedestrians to them". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 82 (2021), pp. 297–315. DOI: 10.1016/j.trf.2021.07.019.

[192] Mitko Aleksandrov et al. "People Choice Modelling for Evacuation of Tall Buildings". In: *Fire Technology* 54 (May 2018). DOI: 10.1007/s10694-018-0731-1.

[193] Ruggiero Lovreglio, Achille Fonzone, and Luigi dell'Olio. "A mixed logit model for predicting exit choice during building evacuations". In: *Transportation Research Part A: Policy and Practice* 92 (2016), pp. 59–75. ISSN: 0965-8564. DOI: 10.1016/j.tra.2016.06.018.

[194] Milad Haghani, Majid Sarvi, and Abbas Rajabifard. "Simulating Indoor Evacuation of Pedestrians: The Sensitivity of Predictions to Directional-Choice Calibration Parameters". In: *Transportation Research Record* (2018). DOI: 10.1177/0361198118796351.

[195] Milad Haghani and Majid Sarvi. "Simulating dynamics of adaptive exit-choice changing in crowd evacuations: Model implementation and behavioural interpretations". In: *Transportation Research Part C: Emerging Technologies* 103 (2019), pp. 56–82. ISSN: 0968-090X. DOI: doi.org/10.1016/j.trc.2019.04.009.

[196] Ryo Nishida, Masaki Onishi, and Koichi Hashimoto. "Crowd simulation incorporating a route choice model and similarity evaluation using real large-scale data". In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)* (2023), 2751–2753.

[197] Zhan Zhao and Yuebing Liang. "A deep inverse reinforcement learning approach to route choice modeling with context-dependent rewards". In: *Transportation Research Part C: Emerging Technologies* 149 (2023), p. 104079. DOI: 10.1016/j.trc.2023.104079.

[198] Anne Templeton et al. "Agent-based models of social behaviour and communication in evacuations: A systematic review". In: *arXiv e-prints (preprint)* (Oct. 2023). DOI: 10.48550/arXiv.2310.15761.

[199] Gabriele Bernardini et al. "An Agent-based Model for Earthquake. Pedestrians Evacuation Simulation in Urban Scenarios". In: *Transportation Research Procedia* 2 (2014). The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands, pp. 255–263. DOI: 10.1016/j.trpro.2014.09.050.

[200] Yan Yang, Liang Mao, and Sara S. Metcalf. "Diffusion of hurricane evacuation behavior through a home-workplace social network: A spatially explicit agent-based simulation model". In: *Computers, Environment and Urban Systems* 74 (2019), pp. 13–22. DOI: 10.1016/j.compenvurbsys.2018.11.010.

[201] Linjiang Zheng et al. "Simulation of pedestrian evacuation considering emergency spread and pedestrian panic". In: *Physica A: Statistical Mechanics and its Applications* 522 (2019), pp. 167–181. ISSN: 0378-4371. DOI: 10.1016/j.physa.2019.01.128.

[202] Alexandros Rigos, Erik Mohlin, and Enrico Ronchi. "The cry wolf effect in evacuation: A game-theoretic approach". In: *Physica A: Statistical Mechanics and its Applications* 526 (2019), p. 120890. DOI: 10.1016/j.physa.2019.04.126.

[203] Nuria Pelechano et al. "Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication". In: *International Workshop on Crowd Simulation (V-CROWDS 2005)* (Nov. 2005). URL: http://gamma.cs.unc.edu/LARGE/papers/psych_role_and_communication.pdf.

[204] Tibor Bosse et al. "Modelling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions". In: *Autonomous Agents and Multi-Agent Systems* 27.1 (July 2013), pp. 52–84. ISSN: 1573-7454. DOI: 10.1007/s10458-012-9201-1.

[207] Isabella von Sivers, Florian Künzner, and Gerta Köster. "Pedestrian Evacuation Simulation with Separated Families". In: *Proceedings of the 8th International Conference on Pedestrian and Evacuation Dynamics (PED2016)*. Hefei, China, Oct. 2016.

[208] Mohcine Chraibi and Jun Zhang. "JuPedSim: an open framework for simulating and analyzing the dynamics of pedestrians". In: *SUMO2016 - Traffic, Mobility, and Logistics, Proceedings*. Vol. 30. Berichte aus dem DLR-Institut für Verkehrssystemtechnik. SUMO Conference 2016. May 2016, pp. 127–134.

[209] Benedikt Kleinmeier et al. "Vadere: An Open-Source Simulation Framework to Promote Interdisciplinary Understanding". In: *Collective Dynamics* 4 (2019). DOI: 10.17815/CD.2019.21.

[210] Patrick Taillandier et al. "A BDI Agent Architecture for the GAMA Modeling and Simulation Platform". In: *Multi-Agent Based Simulation XVII*. Ed. by L. Nardin and L. Antunes. Vol. 10399. Lecture Notes in Computer Science. Springer, 2017. ISBN: 978-3-319-67476-6. DOI: 10.1007/978-3-319-67477-3_1.

[211] George Datseris, Ali R. Vahdati, and Timothy C. DuBois. "Agents.jl: a performant and feature-full agent-based modeling software of minimal code complexity". In: *SIMULATION* 0.0 (2022), p. 003754972110688. DOI: 10.1177/00375497211068820.

[214] David Masad and Jacqueline Kazil. "Mesa: An Agent-Based Modeling Framework". In: *Python in Science Conference 2015*. 2015, pp. 51–58. DOI: 10.25080/Majora-7b98e3ed-009.

Bibliography

[216] Sean Curtis, Andrew Best, and Dinesh Manocha. "Menge: A Modular Framework for Simulating Crowd Movement". In: *Collective Dynamics* (2016). DOI: doi: 10.17815/CD.2016.1.

[217] Timo Korhonen et al. "Integration of an agent based evacuation simulation and state-of-the-art fire simulation". In: *Proceedings of the 7th Asia-Oceania Symposium on Fire Science & Technology, Hong Kong.* 2007.

[222] Shanzhi Chen et al. "V2X Network Architecture and Standards System". In: *Cellular Vehicle-to-Everything (C-V2X).* Springer Nature Singapore, 2023, pp. 81–116. ISBN: 9789811951305. DOI: 10.1007/978-981-19-5130-5_3.

[223] Rakesh Shrestha et al. "Evolution of V2X Communication and Integration of Blockchain for Security Enhancements". In: *Electronics* 9.9 (2020). DOI: 10.3390/electronics9091338.

[227] A Costandoiu and Monica Leba. "Convergence of V2X communication systems and next generation networks". In: *IOP Conference Series: Materials Science and Engineering* 477 (2019), p. 012052. DOI: 10.1088/1757-899X/477/1/012052.

[229] Badreddine Yacine Yacheur, Toufik Ahmed, and Mohamed Mosbah. "Analysis and Comparison of IEEE 802.11p and IEEE 802.11bd". In: *Communication Technologies for Vehicles.* Ed. by Francine Krief et al. Cham: Springer International Publishing, 2020, pp. 55–65. DOI: 10.1007/978-3-030-66030-7_5.

[230] Alessandro Bazzi et al. "On the Design of Sidelink for Cellular V2X: A Literature Review and Outlook for Future". In: *IEEE Access* 9 (2021), pp. 97953–97980. DOI: 10.1109/access.2021.3094161.

[244] Rafael Molina-Masegosa, Javier Gozalvez, and Miguel Sepulcre. "Comparison of IEEE 802.11p and LTE-V2X: An Evaluation With Periodic and Aperiodic Messages of Constant and Variable Size". In: *IEEE Access* 8 (2020), pp. 121526–121548. DOI: 10.1109/ACCESS.2020.3007115.

[246] Caleb Phillips, Douglas Sicker, and Dirk Grunwald. "A Survey of Wireless Path Loss Prediction and Coverage Mapping Methods". In: *IEEE Communications Surveys & Tutorials* 15.1 (2013), pp. 255–270. DOI: 10.1109/SURV.2012.022412.00172.

[247] H.T. Friis. "A Note on a Simple Transmission Formula". In: *Proceedings of the IRE* 34.5 (1946), pp. 254–256. DOI: 10.1109/JRPROC.1946.234568.

[248] T. Kawano Y. Okumura E. Ohmori and K. Fukuda. "Field strengh and its variability in VHF and UHF land mobile radio service". In: *Rev. Electr. Commun. Lab.* 16 (1968), pp. 825–873.

[249] Basile K.I. Thierry F.S. Deussom Djomadji E.M. and T. Emanuel. "COST 231-Hata Propagation Model Optimization in 1800 MHz Band Based on Magnetic Optimization Algorithm: Application to the City of Limbe". In: *Journal of Computer and Communications* 11 (2023), pp. 57–74. DOI: 10.4236/jcc.2023.112005.

[250]   Bernard Sklar. "Rayleigh Fading Channels in Mobile Digital Communication Systems Part II: Mitigation". In: *Communications Magazine, IEEE* 35 (Oct. 1997), pp. 148–155. DOI: 10.1109/MCOM.1997.621038.

[251]   G. Durgin, T.S. Rappaport, and H. Xu. "5.85-GHz radio path loss and penetration loss measurements in and around homes and trees". In: *IEEE Communications Letters* 2.3 (1998), pp. 70–72. DOI: 10.1109/4234.662630.

[252]   Y.L.C. de Jong, M.H.J.L. Koelen, and M.H.A.J. Herben. "A building-transmission model for improved propagation prediction in urban microcells". In: *IEEE Transactions on Vehicular Technology* 53.2 (2004), pp. 490–502. DOI: 10.1109/TVT.2004.823491.

[253]   Saul Torrico, Henry Bertoni, and Roger Lang. "Modeling tree effects on path loss in a residential environment". In: *Antennas and Propagation, IEEE Transactions on* 46 (July 1998), pp. 872–880. DOI: 10.1109/8.686776.

[259]   Lingfeng Liu et al. "The COST 2100 MIMO channel model". In: *IEEE Wireless Communications* 19.6 (2012), pp. 92–99. DOI: 10.1109/MWC.2012.6393523.

[265]   Andras Varga. "A Practical Introduction to the OMNeT++ Simulation Framework". In: May 2019, pp. 3–51. ISBN: 978-3-030-12841-8. DOI: 10.1007/978-3-030-12842-5_1.

[266]   George Riley and Thomas Henderson. "The ns-3 Network Simulator". In: Jan. 2010, pp. 15–34. ISBN: 9783642123306. DOI: 10.1007/978-3-642-12331-3_2.

[267]   Giovanni Nardini et al. "Simu5G - An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks". In: *IEEE Access* 8 (2020), pp. 181176–181191. DOI: 10.1109/access.2020.3028550.

[268]   X. Zeng, R. Bagrodia, and M. Gerla. "GloMoSim: a library for parallel simulation of large-scale wireless networks". In: *Proceedings. Twelfth Workshop on Parallel and Distributed Simulation PADS '98 (Cat. No.98TB100233)*. 1998, pp. 154–161. DOI: 10.1109/PADS.1998.685281.

[269]   Giuseppe Piro et al. "Simulating LTE Cellular Systems: An Open-Source Framework". In: *IEEE Transactions on Vehicular Technology* 60.2 (2011), pp. 498–513. DOI: 10.1109/TVT.2010.2091660.

[271]   Atta R. Khan, Sardar M. Bilal, and Mazliza Othman. "A performance comparison of open source network simulators for wireless networks". In: *2012 IEEE International Conference on Control System, Computing and Engineering*. 2012, pp. 34–38. DOI: 10.1109/ICCSCE.2012.6487111.

[272]   Fan Bai et al. "Modeling Path Duration Distributions in MANETs and Their Impact on Reactive Routing Protocols". In: *Selected Areas in Communications, IEEE Journal on* 22 (Oct. 2004), pp. 1357–1373. DOI: 10.1109/JSAC.2004.829353.

[273] Axel Wegener et al. "TraCI: an interface for coupling road traffic and network simulators". In: *Proceedings of the 11th communications and networking simulation symposium on - CNS '08.* ACM Press, 2008, pp. 155–163. ISBN: 1-56555-318-7. DOI: 10.1145/1400713.1400740.

[275] Steffen Schütte, Stefan Scherfke, and Michael Sonnenschein. "mosaik - Smart Grid Simulation API". In: *International Conference on Smart Grids and Green IT Systems.* Apr. 2012, pp. 15–24.

[276] Florian Schloegl et al. "Towards a classification scheme for co-simulation approaches in energy systems". In: *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST).* 2015, pp. 516–521. DOI: 10.1109/SEDST.2015.7315262.

[277] C. Steinbrink et al. "CPES testing with MOSAIK: Co-Simulation planning, execution and analysis". In: *Applied Sciences* 9.5 (2019). DOI: 10.3390/app9050923.

[278] Luca Barbierato et al. "COMET: Co-simulation of Multi-Energy Systems for Energy Transition". In: *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC).* 2022, pp. 1343–1348. DOI: 10.1109/COMPSAC54236.2022.00212.

[279] Christoph Binder et al. "Applying model-based Co-Simulation on modular Production Units in Complex Automation Systems". In: *2021 IEEE International Symposium on Systems Engineering (ISSE).* 2021, pp. 1–6. DOI: 10.1109/ISSE51541.2021.9582531.

[281] C. Steinbrink et al. "Smart grid co-simulation with MOSAIK and HLA: a comparison study". In: *Computer Science - Research and Development* 33.1-2 (2018), pp. 135–143. DOI: 10.1007/s00450-017-0379-y.

[287] Dongbin Xiu and George Em Karniadakis. "The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations". In: *SIAM Journal on Scientific Computing* 24.2 (2002), pp. 61–644. DOI: 10.1137/S1064827501387826.

[288] Dongbin Xiu. "Fast numerical methods for stochastic computations: a review". In: *Communications in computational physics* 5.2-4 (2009), pp. 242–272.

[289] Jonathan Feinberg, Vinzenz Gregor Eck, and Hans Petter Langtangen. "Multivariate Polynomial Chaos Expansions with Dependent Variables". In: *SIAM Journal on Scientific Computing* 40.1 (2018), pp. 199–223. DOI: 10.1137/15M1020447.

[290] Bruno Sudret. "Global sensitivity analysis using polynomial chaos expansions". In: *Reliability Engineering & System Safety* 93.7 (2008). Bayesian Networks in Dependability, pp. 964–979. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2007.04.002.

[291] Andrea Saltelli et al. "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index". In: *Computer Physics Communications* 181.2 (2010), pp. 259–270. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2009.09.018.

[292]   Thierry A. Mara and William E. Becker. "Polynomial chaos expansion for sensitivity analysis of model output with dependent inputs". In: *Reliability Engineering & System Safety* 214 (2021), p. 107795. DOI: 10.1016/j.ress.2021.107795.

[293]   Matthias Rupp, Stefan Schuhbäck, and Lars Wischhof. "Coupling Microscopic Mobility and Mobile Network Emulation for Pedestrian Communication Applications". In: *CoRR* abs/2109.12018 (2021). arXiv: 2109.12018. URL: https://arxiv.org/abs/2109.12018.

[294]   Stefan Schuhbäck and Lars Wischhof. "Decentralized Pedestrian Density Maps Based On Sidelink Communication". In: *ICC 2021 - 2021 IEEE International Conference on Communications (ICC)* (June 17, 2021), pp. 1–6. DOI: 10.1109/ICCWorkshops50388.2021.9473545.

[295]   Christina Maria Mayr et al. "Analysis of information dissemination through direct communication in a moving crowd". In: *Safety Science* 142 (2021), p. 105386. DOI: 10.1016/j.ssci.2021.105386.

[297]   Stefan Schuhbäck et al. "Towards a Bidirectional Coupling of Pedestrian Dynamics and Mobile Communication Simulation". In: *Proceedings of the OMNeT++ Community Summit 2019*. 2019. DOI: 10.29007/nnfj.

[298]   Levente Mészáros, Andras Varga, and Michael Kirsche. "INET Framework". In: May 2019, pp. 55–106. ISBN: 978-3-030-12841-8. DOI: 10.1007/978-3-030-12842-5_2.

[299]   Antonio Virdis, Giovanni Stea, and Giovanni Nardini. "SimuLTE – A Modular System-level Simulator for LTE/LTE-A Networks based on OMNeT++". In: *Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*. SCITEPRESS - Science and Technology Publications, 2014. DOI: 10.5220/0005040000590070.

[300]   Christoph Sommer, Reinhard German, and Falco Dressler. "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis". In: *IEEE Transactions on Mobile Computing* 10.1 (Jan. 2011), pp. 3–15. DOI: 10.1109/tmc.2010.133.

[301]   Raphael Riebl et al. "Artery: Extending Veins for VANET applications". In: *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, June 2015. DOI: 10.1109/mtits.2015.7223293.

[302]   Raphael Riebl, Christina Obermaier, and Hendrik-Jörn Günther. "Artery: Large Scale Simulation Environment for ITS Applications". In: 2019, pp. 365–406. ISBN: 978-3-030-12841-8. DOI: 10.1007/978-3-030-12842-5_12.

[305]   Benedikt Kleinmeier, Gerta Köster, and John Drury. "Agent-Based Simulation of Collective Cooperation: From Experiment to Model". In: *Journal of the Royal Society Interface* 17 (171 2020), p. 20200396. ISSN: 1742-5662. DOI: 10.1098/rsif.2020.0396.

[306]   Christina Maria Mayr and Gerta Köster. "Guiding crowds when facing limited compliance: Simulating strategies". In: *PLOS ONE* 17.11 (Nov. 2022), pp. 1–24. DOI: 10.1371/journal.pone.0276229. URL: 10.1371/journal.pone.0276229.

[307]   Christina Maria Mayr, Anne Templeton, and Gerta Köster. "Designing mobile application messages to impact route choice: A survey and simulation study". In: *PLOS ONE* 18.4 (Apr. 2023), pp. 1–20. DOI: 10.1371/journal.pone.0284540.

[308]   Giovanni Nardini, Antonio Virdis, and Giovanni Stea. "Modeling Network-Controlled Device-to-Device Communications in SimuLTE". In: *Sensors* 18.10 (Oct. 2018), p. 3551. DOI: 10.3390/s18103551.

[309]   Fernando Fonseca et al. "Use and Perceptions of Pedestrian Navigation Apps: Findings from Bologna and Porto". In: *ISPRS International Journal of Geo-Information* 10.7 (2021). DOI: 10.3390/ijgi10070446.

[312]   Christina Maria Mayr and Gerta Köster. "Estimating pedestrian flows using route distributions and sparse counting data". In: *Collective Dynamics* (2024), pp. 1–8. DOI: 10.17815/CD.2024.149.

Books

[5]   W. O'Toole and T. Semmens. *Crowd Management: Risk, Security and Health.* Events management theory and methods series. Goodfellow Publishers, Limited, 2019. ISBN: 9781911396895.

[6]   J.R. Silvers and W. O'Toole. *Risk Management for Events.* Events Management. Taylor & Francis, 2020. ISBN: 9781000299236.

[7]   Claudio Feliciani, Kenichiro Shimura, and Katsuhiro Nishinari. *Introduction to Crowd Management. Managing Crowds in the Digital Era: Theory and Practice.* 1st ed. Springer Cham, 2021. ISBN: 9783030900120. DOI: 10.1007/978-3-030-90012-0.

[8]   G. K. Still. *Applied Crowd Science.* CRC PRESS, 2021. ISBN: 9781351053044.

[32]   Pushkin Kachroo et al. *Pedestrian Dynamics: Feedback Control of Crowd Evacuation.* Springer, 2010. DOI: 10.1007/978-3-540-75561-6.

[50]   Morgan J. Hurley et al. *SFPE Handbook of Fire Protection Engineering.* 5th ed. New York: Springer, 2016. ISBN: 9781493925643. DOI: 10.1007/978-1-4939-2565-0.

[55]   J. Zhang. *Pedestrian Fundamental Diagrams: Comparative Analysis of Experiments in Different Geometries.* Schriften des Forschungszentrums Jülich / IAS series: IAS series. Forschungszentrum, Zentralbibliothek, 2012. ISBN: 9783893368259.

[72]   Ulrich Weidmann. *Transporttechnik der Fußgänger.* 2nd ed. Vol. 90. Schriftenreihe des IVT. Zürich: Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau (IVT) ETH, 1993. DOI: 10.3929/ethz-b-000242008.

[78]   Transportation Research Board, Engineering National Academies of Sciences, and Medicine. *Highway Capacity Manual 7th Edition: A Guide for Multimodal Mobility Analysis.* Washington, DC: The National Academies Press, 2022. ISBN: 978-0-309-08766-7. DOI: 10.17226/26432.

[80]   D.J. Carter et al. *Pedestrian and Bicyclist Intersection Safety Indices: User guide.* Turner-Fairbank Highway Research Center, Federal Highway Administration, 2007.

[84]   Gustave Le Bon. *La Psychologie des Foules.* Alcan, 1895. URL: http://www.gutenberg.org/ebooks/24007.

[85]   Sigmund Freud. *Group psychology and the analysis of the ego.* International Psychoanalytic Publishing House, 1921.

[86]   Floyd Henry Allport. *Social Psychology.* Boston: Houghton Mifflin Company, 1924.

[88]   Muzafer Sherif. *The psychology of social norms.* Harper & Brothers, 1936. URL: https://archive.org/details/in.ernet.dli.2015.264611/.

[89]   R. H. Turner and L. Killian. *Collective Behaviour.* Englewood Cliffs, NJ: Prentice Hall, 1957.

[91]   R.A. Berk. *Collective Behavior.* Elements of sociology. W. C. Brown Company, 1974. ISBN: 9780697075253.

[98]   Ralph Herbert Turner and Lewis M. Killian. *Collective Behaviour.* Ed. by Series Editor Herbert Blumer. 2nd. Prentice Hall, 1972.

[110]   D.K. Berlo. *The Process of Communication: An Introduction to Theory and Practice.* The Process of Communication: An Introduction to Theory and Practice 10. Holt, Rinehart and Winston, 1960. ISBN: 9780030074905.

[213]   Bertrand Maury and Sylvain Faure. *Crowds in Equations: An Introduction to the Microscopic Modeling of Crowds.* WORLD SCIENTIFIC (EUROPE), 2018. DOI: 10.1142/q0163.

[220]   Theodore S. Rappaport. *Wireless Communications: Principles and Practice.* 2nd ed. Prentice Hall, 2001, p. 736. ISBN: 9780130422323.

[245]   Andreas F. Molisch. *Wireless communication.* 2nd ed. Wiley & Sons Ltd., 2011. ISBN: 9780470666692.

[254]   Antonio Virdis and Michael Kirsche, eds. *Recent Advances in Network Simulation.* Springer International Publishing, 2019. DOI: 10.1007/978-3-030-12842-5.

[283]   L. Zhang, B.P. Zeigler, and Y. LaiLi. *Model Engineering for Simulation.* Elsevier Science, 2019. ISBN: 9780128135440.

[285]   Ralph C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications.* Computational Science and Engineering. Society for Industrial and Applied Mathematics, 2014. ISBN: 978-1-611973-21-1. DOI: doi.org/10.1137/1.9781611973228.

[286] Dongbin Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton, NJ: Princeton University Press, 2010. DOI: 10.2307/j.ctv7h0skv.

[296] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. 1st ed. Prentice Hall, 2008. ISBN: 0132350882.

[310] J.L. Burbank, W. Kasch, and J. Ward. *An Introduction to Network Modeling and Simulation for the Practicing Engineer*. The ComSoc Guides to Communications Technologies. Wiley, 2011. ISBN: 9781118063644.

[311] John J. Fruin. *Pedestrian Planning and Design*. New York: Metropolitan Association of Urban Designers and Environmental Planners, 1964.

Theses

[51] Ulrich Weidmann. "Der Fahrgastwechsel im öffentlichen Personenverkehr". Dissertation ETH Nr.10630. PhD thesis. Eidgenössische Technische Hochschule Zürich, 1994.

[68] Ernst Bosina. "A New Generic Approach to the Pedestrian Fundamental Diagram". PhD thesis. Institut für Verkehrsplanung und Transportsysteme (IVT), ETH Zürich, 2018. DOI: 10.3929/ethz-b-000296226.

[152] Benedikt Sebastian Zönnchen. "Efficient parallel algorithms for large-scale pedestrian simulation". Dissertation. München: Technische Universität München, 2021.

[154] Michael J. Seitz. "Simulating pedestrian dynamics: Towards natural locomotion and psychological decision making". PhD thesis. Munich, Germany: Technische Universität München, 2016.

[155] Benedikt Kleinmeier. "Modeling of Behavioral Changes in Agent-Based Simulations". PhD thesis. Technische Universität München, 2021.

[156] Xiaoshan Pan. "Computational Modeling of Human and Social Behaviors for Emergency Egress Analysis". PhD thesis. Stanford University, 2006.

[163] Stefan Seer. "A unified framework for evaluating microscopic pedestrian simulation models". TU Wien Online Katalog (AC12656133). PhD thesis. Fakultät für Mathematik und Geoinformation, Institut für Analysis und Scientific Computing: Technische Universität Wien, 2015.

[164] Isabella Katharina Maximiliana von Sivers. "Modellierung sozialpsychologischer Faktoren in Personenstromsimulationen - Interpersonale Distanz und soziale Identitäten". PhD thesis. Technische Universität München, 2016.

[176] Benedikt Zönnchen. "Navigation around pedestrian groups and queueing using a dynamic adaption of traveling". Bachelor's thesis. Hochschule München, Sept. 2013.

[180]   Antonin Danalet. "Activity choice modeling for pedestrian facilities". PhD thesis. Lausanne: INTER, 2015, p. 188. DOI: 10.5075/epfl-thesis-6806.

[186]   Winnie Daamen. "Modelling passenger flows in public transport facilities". PhD thesis. Delft, The Netherlands: Delft University of Technology, 2004. ISBN: 90-407-2521-7.

[205]   Peter Michael Kielar. "Kognitive Modellierung und Computergestützte Simulation der Räumlich-Sequenziellen Zielauswahl von Fußgängern". Dissertation. München: Technische Universität München, 2017.

[206]   Nanda Wijermans. "Understanding Crowd Behaviour: Simulating Situated Individuals". PhD thesis. Rijksuniversiteit Groningen, 2011.

[215]   Christian Gloor. "Distributed intelligence in real-world mobility simulations". PhD thesis. ETH Zurich, 2005, pp. 1–212.

[270]   Rimon Barr. "An efficient, unifying approach to simulation using virtual machines". PhD thesis. USA, 2004. DOI: 10.5555/1037322.

[274]   Florian Lindner. "Data Transfer in Partitioned Multi-Physics Simulations: Interpolation & Communication". PhD thesis. 2019.

[280]   Bernhard Gatzhammer. "Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions". PhD thesis. 2014.

[282]   Cornelius Steinbrink. "Uncertainty Quantification for Modular Co-Simulation". PhD thesis. University of Oldenburg, Germany, 2017. ISBN: 978-3-95599-046-6.

[284]   Eric W. Weisel. "Models, Composability, and Validity". PhD thesis. 2004. DOI: 10.25777/43pv-gs24.

Other sources

[2]   Dorothy Stanley. *IEEE 802.11bd-2022*. Mar. 10, 2023.

[37]   Marcel Menner et al. *MPC-based Pedestrian Routing for Congestion Balancing*. Mitsubishi Electric Research Laboratories, 2023.

[48]   Grotenhuis J. Lankhuijzen R. Tertoolen G. *Human factors en dynamisch verkeersmanagement: Waar psychologie en techniek samenkomen*. Amsterdam, 2012.

[52]   V.M. Predtechenskii and A.I. Milinskii. *Planning for Foot Traffic Flow in Buildings*. 1968.

[56]   Manuel Kitzlinger and Angelika Kneidl. *Staubewertung mithilfe von Personenstromanalysen*. Jan. 2020.

[69]   John J. Fruin. *Designing for pedestrians: a level-of-service concept*. 1971.

[73]   RiMEA. *Guideline for Microscopic Evacuation Analysis*. 3.0.0. RiMEA e.V. 2016. URL: http://www.rimea.de/.

[76]   Dirk Oberhagemann. *Statische und dynamische Personendichten bei Großver-anstaltungen*. Tech. rep. Technisch-Wissenschaftlicher Beirat (TWB) der Vereinigung zur Förderung des Deutschen Brandschutzes e.V., 2012.

[79]   Transport for London. *Pedestrian Comfort Level Guidance*. 2010. URL: https://content.tfl.gov.uk/pedestrian-comfort-guidance-technical-guide.pdf.

[87]   Philip Zimbardo. *Stanford Prison Experiment*. Accessed: 5. August 2020 (via Internet Archive Wayback Machine). 1999. URL: https://web.archive.org/web/20000512020449/http://www.prisonexp.org/slide-4.htm.

[92]   Rose Challenger et al. *Understanding Crowd Behaviours: Supporting Evidence*. Tech. rep. University of Leeds, 2009.

[212]   A. Meinert et al. *Crowd simulation in urban areas with openstreetmap data*. 2019.

[218]   Peter Kielar, Daniel Biedermann, and Andre Borrmann. *MomenTUMv2: A Modular, Extensible, and Generic Agent-Based Pedestrian Behavior Simulation Framework*. Tech. rep. TUM, Aug. 2016. DOI: 10.13140/RG.2.2.21387.69929.

[219]   Thomas Richards. *A review of software for crowd simulation*. Leeds Institute for Data Science (LIDA), University of Leeds. Accessed: 4. November 2020. Mar. 10, 2020. URL: https://urban-analytics.github.io/dust/docs/ped_sim_review.pdf.

[221]   802.11 WG Wireless LAN Working Group. *IEEE 802.11p-2010*. Tech. rep. IEEE standards association, 2010.

[224]   IEEE. *IEEE Std 1609.4-2016 (Revision of IEEE Std 1609.4-2010), IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-Channel Operation*. Tech. rep. IEEE Vehicular Technology Society, 2016, pp. 1–94. DOI: 10.1109/IEEESTD.2016.7435228.

[225]   ETSI. *EN 302 663 V1.3.0 (2019-05), Intelligent Transport Systems (ITS);ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band*. Tech. rep. ETSI, May 2019. URL: https://www.etsi.org/deliver/etsi_en/302600_302699/302663/.

[226]   ITS. *ARIB STD-T109 700 MHz Band Intelligent Transport Systems*. Tech. rep. 2012.

[228]   802.11 WG Wireless LAN Working Group. *IEEE 802.11bd-2022 - IEEE Standard for Information Technology –Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks – Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Next Generation V2X*. Tech. rep. Mar. 10, 2023.

[231]   3GPP Technical Specification Group Services and System Aspects. *TR 22.885 Study on LTE Support for Vehicle to Everything (V2X) Services (Release 14)*. Tech. rep. 2015.

[232]   3GPP Technical Specification Group Services and System Aspects. *TR 36.885 Study on LTE-Based V2X Services (Release 14)*. Tech. rep. 2016.

[233]   3GPP Technical Specification Group Services and System Aspects. *3GPP TS 22.185 Service Requirements for V2X Services (Release 16)*. Tech. rep. 2020.

[234]   3GPP Technical Specification Group Services and System Aspects. *TS 23.285 Architecture Enhancements for V2X Services (Release 16)*. Tech. rep. 2020.

[235]   3GPP Technical Specification Group Radio Access Network. *TS 36.213 Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures (Release 16)*. Tech. rep. 2020.

[236]   Technical Specification Group Radio Access Network. *TS 36.321 Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) Protocol Specification (Release 16)*. Tech. rep. 2020.

[237]   3GPP Technical Specification Group Services and System Aspects. *TR 22.886 Study on Enhancement of 3GPP Support for 5G V2X Services (Release 16)*. Tech. rep. 2018.

[238]   3GPP Technical Specification Group Radio Access Network. *TR 38.885 Study on NR Vehicle-to-Everything (V2X) (Release 16)*. Tech. rep. 2019.

[239]   3GPP Technical Specification Group Services and System Aspects. *TS 23.287 Architecture Enhancements for 5G System (5GS) to Support Vehicle-to-Everything (V2X) Services (Release 16)*. Tech. rep. 2020.

[240]   3GPP Technical Specification Group Services and System Aspects. *TS 22.186 Enhance-ment of 3GPP support for V2X scenarios (Release 16)*. Tech. rep. 2019.

[241]   3GPP Technical Specification Group Radio Access Network. *TS 38.213 Physical Layer Procedures for Control (Release 16)*. Tech. rep. 2020.

[242]   3GPP Technical Specification Group Radio Access Network. *TS 38.321 Medium Access Control (MAC) Protocol Specification (Release 16)*. Tech. rep. 2020.

[243]   3GPP Technical Specification Group Radio Access Network. *TS 37.895 Overall Description of Radio Access Network (RAN) Aspects for Vehicle-to-Everything (V2X) Based on LTE and NR (Release 16)*. Tech. rep. 2020.

[255]   P. Kyritsi A. Molisch D. Baum A. Gorokhov et al. V. Erceg L. Schumacher. *IEEE 802.11 TGn Channel Models*. Tech. rep. 802.11-03/940r4. IEEE, 2004.

[256]   L. Hentila X. Zhao T. Jamsa C. Schneider P. Kyosti J. Meinila. *WINNER II channel models*. 2007.

[257]   Spatial Channel Model AHG (Combined ad-hoc from 3GPP and 3GPP2). *Spatial Channel Model SCM-134 text V6.0*. 2003.

[258]   *Study on 3D Channel Model for LTE*. 2017.

[260]   Radio communication sector of ITU. *Guidelines for evaluation of radio interface technologies for IMT-2020*. Tech. rep. ITU-R M.2412-0. International Telecommunication Union, 2017.

[261] 3GPP. *TR 38.901 Study on channel model for frequencies from 0.5 to 100 GHz V15.0.0.* 2018.

[262] *ICT-317669-METIS/D1.4 report. METIS channel model.* 2015.

[263] M. Peter et al. *Measurement results and final mmMAGIC channel models.* 2017.

[264] S. Jaeckel et al. *QuaDRiGa-quasi deterministic radio channel generator user manual and documentation.* 2019.

[303] Ulrich Möncke. *Rechtsfragen des roVer-Projekts und eines roVer-basierten Verkehrsleitdienstes.* Tech. rep. Fakultät für Informatik und Mathematik, 2023, p. 198.

[304] International Telecommunication Union (ITU). *Propagation data and prediction methods for the planning of short-range outdoor radio communication systems and radiolocal area networks in the frequency range 300 MHz to 100 GHz.* Tech. rep. International Telecommunication Union, Feb. 2012.

A. Software, data, collaborations

I follow the principles of Open Science as I am convinced they make research more transparent:

- I make my dissertation thesis and all scientific publications, to which I refer to, **open access**.

- I make the software *CrowNet* **free and open source**. I use this software in my studies.

- I make my simulation **data open**. This includes raw data, post-processing scripts, and the actual results.

- I foster **collaboration** with experts from different fields.

- I ensure the **reproducibility** of my data.

I guarantee that my data and source code is at least available until 01.01.2035. For storing these, I use *LRZ sync and share* and *github*.

Simulation data and software

The results of my simulation studies (Chapter 4) are publicly available via *LRZ sync and share*:

- Raw data (57 GB). Includes trajectories, database with mobile communication recordings, etc. https://syncandshare.lrz.de/getlink/fiTUk3Xso5YSpLezh5dXXR/rawdata-2023-12-16.zip.

- Output from analysis (0.06 GB). Includes figures, tables, text files, etc. https://syncandshare.lrz.de/getlink/fiDVxs4LRF2gaD4P4dHDQM/data-2023-12-16-11.zip

The data was created with the software *CrowNet* developed at Hochschule München University of Applied Sciences in Munich, Germany. The software is described in Chapter 3. The data was produced using the stable branch of the software (stable branch: 15 Dec 2023, git commit hash: 69af8b55bc9f40f21ad32070ee429397635b4d85). The software is freely available on github: https://github.com/roVer-HM/crownet.

The 15 Dec 2023 state is freely available

- as zip-file (version 69af8b55) via LRZ sync and share (6GB): Contains binaries, virtual environments. `https://syncandshare.lrz.de/getlink/fiC254uPj28rFMp2ke1QXG/crownet.zip`

- as repository on github (Version 69af8b55): Clean repository (requires less storage because it does not contain any binaries or virtual environments). `https://github.com/roVer-HM/crownet/commit/69af8b55bc9f40f21ad32070ee429397635b4d85`

Three Docker images are used in the simulation studies: one for each of the simulators *Vadere, flowcontrol and OMNeT++*. The Docker images are publicly available

- through the image registry of the *CrowNet* repository on github: `https://github.com/orgs/roVer-HM/packages?repo_name=crownet`. You can simply download the images by copying the following commands in a terminal: `docker pull ghcr.io/rover-hm/vadere:46c81df1 && docker pull ghcr.io/rover-hm/flowcontrol:496ff02c && docker pull ghcr.io/rover-hm/omnetpp:6.0.1`

- as zip-file on sync and share (3.3 GB): download the images from `https://syncandshare.lrz.de/getlink/fiY1WxE1p8VoQtHFMCk3K5/dockerimages.zip` and extract the zip-file and restore each image using `docker load`.

Reproducability of the simulation data

To reproduce the results of Chapter 4, please follow the next steps. First, make sure that your system meets the following requirements:

- Linux operating system with `git` and `Docker` installed

- ≥ 70 GB available memory

- ≥ 80 CPUs

- ≥ 170 GB free disk space for storing the simulation data

Second, run the simulation studies. This comprises cloning the *CrowNet* repository and executing the script `crownet/scripts/run_simulations_Diss_Mayr`. You can simply do this by pasting the following lines in your terminal:

```
1 $ git clone --recurse-submodules https://github.com/roVer-HM/crownet.git
2 $ cd crownet && git checkout 69af8b55bc9f40f21ad32070ee429397635b4d85
3 $ source setup -i
4 $ cd scripts && ./run_simulations_Diss_Mayr
```

Note that the script checks whether your system fulfills the requirements before the simulation studies are executed. Third, wait approximately 14 hours for the simulations to be finished. The script will tell you whether the simulation data has been reproduced successfully. This procedure was tested on the system described in Appendix B.

Survey data

The raw data from the online survey (see Section 4.3), as well as analysis scripts and results of statistical tests are freely available

- as zipped repository via LRZ sync and share (0.5MB): `https://syncandshare.lrz.de/getlink/fiWz7mx6v8ZG7mCu2UKvi6/survey-route-choice-master.zip`

- on github: `https://github.com/pedestrian-dynamics-HM/survey-route-choice` (github repository)

To reproduce the results, please clone or download the repository and run the scripts in the root directory. This requires R to be installed. I tested the scripts for R version 3.6.3 (2020-02-29) on a Linux operating system (Ubuntu 20.04).

Collaboration with other scientists

Some parts of this work have been developed in collaboration with other scientists. I would like to acknowledge their contributions in the following.

Contributions to the Software *CrowNet*

CrowNet was developed in the *roVer* research project (2018-2023) at Hochschule München Munich University of Applied Sciences (Munich, Germany).

Stefan Schuhbäck implemented the initial coupling of the simulators *Vadere* and *OM-NeT++* [297]. He also introduced the virtualization of the simulators using Docker images. Furthermore, he proposed the decentralized pedestrian density map application [4, 294].

Several others contributed to the source code of *CrowNet*: Prof. Dr. Lars Wischhof (project lead of *roVer*), Sebastian Benz (student), Matthias Rupp (student), Nico Dassler (student), Wolfgang Kallies (student), and Maximilian Kilian (student). I would like to mention them even if I do not use their work in my investigations.

My individual contributions for each *CrowNet* module are:

- *flowcontrol* module: I developed the core of the novel Python framework on my own. The core comprises the route recommendation algorithms, the timestepping procedures, and the density mapping. Stefan Schuhbäck supported me with the implementation of the Traffic Control Interface that enables the communication with other simulators. *flowcontrol* is my major contribution to the *CrowNet* software.

- *SUQ-controller* module: I extended the existing framework *SUQ-controller* (`https://gitlab.lrz.de/vadere/suq-controller`) that was developed by Daniel Lehmberg. I added new process calls to execute coupled simulations.

- *Vadere* module: I adjusted the existing framework *Vadere* (`https://gitlab.lrz.de/vadere/vadere`) so that dynamically generated stimuli can be processed by the psychology layer. The Traffic Control Interface was implemented by Stefan Schuhbäck in [297].

- *crownetutils* module: I contributed to the `dockerrunner` sub-module. In particular I defined the `dockerrunner` for the *flowcontrol* simulator and added methods to start the containers in the correct order.

- *OMNeT++* module: I did not contribute to this module. The module was developed by Stefan Schuhbäck, Lars Wischhof, and students.

Survey - collaboration with psychologist

In Section 4.3 I present an online survey. The survey design was developed in collaboration with the crowd psychologist Dr. Anne Templeton (University of Edinburgh, UK). I conducted the survey on my own. I was in charge of storing the data and I conducted the statistical analysis in R. We analyzed results and findings together.

Usage of the decentralized density map application

In Section 4.4 I use the decentralized density map application to measure the crowd density in certain areas. The application has been published in [4, 294] to which I refer to repetitively.

Nevertheless I would like to briefly describe how the decentralized density map application is related to my work: The application is one element of my crowd guidance system. It provides density estimates that are fed into the route recommendation algorithm. I consider the application as a black box because I am interested in the performance of the overall system and not in the performance of individual components. Importantly, my scientific contribution is the assessment of the overall system including the interplay of the density map application and the route recommendation algorithm.

The density map application is actively researched and further developed by Stefan Schuhbäck with whom I worked together in the *roVer* research project. At the time of writing this thesis, he was further improving the application. These recent developments have not been published yet. Therefore I used an older version of the density map application in my investigations that has been already published in [4, 294].

Python package omnetinireader

Stefan Schuhbäck and I developed the *omnetinireader* Python package that is publicly available under: https://test.pypi.org/project/omnetinireader/1.7.2/

It serves as external library in the *SUQ-controller* module of the *CrowNet* software.

B. Virtual machine: properties

I run all simulations on a virtual machine as described in Table B.1 and B.2.

Architektur	x86_64
CPU Operationsmodus	32-bit, 64-bit
Byte-Reihenfolge	Little Endian
CPU(s)	80
Liste der Online-CPU(s)	0-79
Thread(s) pro Kern	1
Kern(e) pro Socket	40
Sockel	2
NUMA-Knoten	1
Anbieterkennung	GenuineIntel
Prozessorfamilie	15
Modell	6
Modellname	Common KVM processor
Stepping	1
CPU MHz	2099.998
BogoMIPS	4199.99
Hypervisor-Anbieter	KVM
Virtualisierungstyp	voll
L1d Cach	32K
L1i Cache	32K
L2 Cache	4096K
L3 Cache	16384K
NUMA-Knoten0 CPU(s)	0-79
Markierungen	fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx lm constant_tsc nopl xtopology cpuid tsc_known_freq pni cx16 x2apic hypervisor lahf_lm cpuid_fault pti

Table B.1.: Virtual machine: processor properties. The following output was generated by calling the terminal command `lscpu`.

	Gesamt	belegt	frei	gemns.	Cache	verfügbar
Speicher	94G	5,0G	85G	145M	3,7G	88G
Auslagerungsspeicher	8,0G	0B	8,0G			

Table B.2.: Virtual machine: working memory. The following output was generated by calling the terminal command `free -h`.

C. Bounds of the control parameter (Section 4.1)

The transmitter power parameter should not have an influence in the scenario in Section 4.1 if it is in the range the range $[2\,\mathrm{mW}, 20\,\mathrm{mW}]$.

The free path loss formula describes the loss of the signal strength over distance. It is defined as [247]:

$$\frac{P_{transmitter}}{P_{receiver}} = D_{transmitter} D_{receiver} \left(\frac{4\pi df}{c}\right)^2 \tag{C.1}$$

where D is the antenna directivity which is $D_{transmitter} = D_{receiver} = 1$ in my case (isotropic antenna), d is the distance between transmitter and receiver, f is the frequency which is $f = 2,4 \cdot 10^9 1/s (= 2.4\,\mathrm{GHz})$, and $c = 299792358\,\mathrm{m/s}$ is the light speed. The range d is therefore:

$$d = \sqrt{\frac{P_{transmitter}}{P_{receiver}}} \frac{c}{4\pi f} \tag{C.2}$$

For a transmitter power $P_{transmitter} = 2\,\mathrm{mW}$ and a receiver power of $P_{receiver} = 3.16 \cdot 10^{-9} mW (= -85dB)$ [310] the coverage is $250\,\mathrm{m}$ according to Eq. (C.2). Since the diagonal of the rectangular topography is only $d_{\max} = 213\mathrm{m} (= \sqrt{176\,\mathrm{m}^2 + 120\,\mathrm{m}^2})$, nodes are always in coverage if the transmitter power is $\in [2\,\mathrm{mW}, 20\,\mathrm{mW}]$. See Fig. C.1.
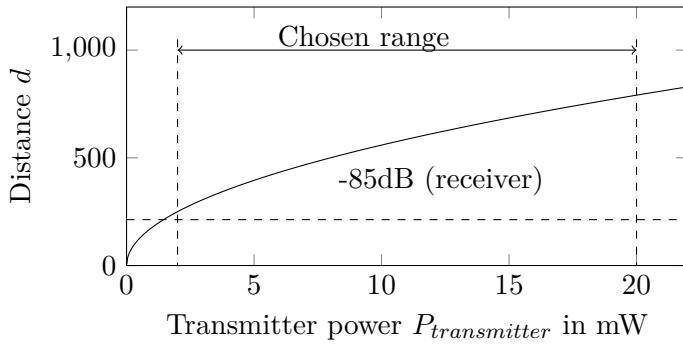
The maximal possible distance between nodes is 213 m.

Figure C.1.: Range over the transmitter power according to the formula (C.2). The greater the transmitter power, the greater the range. With a transmitter power of $2\,\mathrm{mW}$, the range is already $250\,\mathrm{m}$. The maximum distance is $213\,\mathrm{m}$. Therefore all nodes are in coverage for a transmitter power $> 2\,\mathrm{mW}$.

D. Effect of network traffic (Section 4.1)

The study presented here is similar to the study in Section 4.1 and is based on the publication [295]. Code and data associated to the study can be found in the repository `https://github.com/pedestrian-dynamics-HM/crownet-uq-analysis`. The scenario of this study is similar to the scenario presented in Section 4.1 with three exceptions:

- the relay node is not shadowed by walls.
- network traffic is present.
- the arrivals follow an negative exponential distribution.
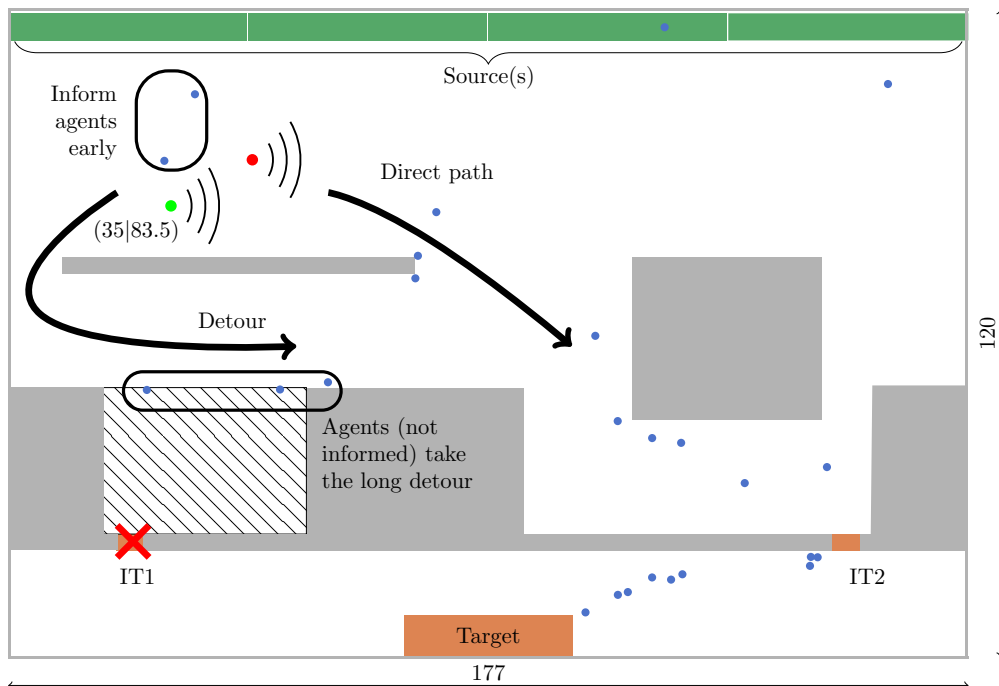
Figure D.1.: Scenario for investigating the effect of network traffic. Scenario similar to the scenario introduced in Sec. 4.1, see Fig. 4.2 with exceptions. The position of the node that starts to disseminate detour information differs from the scenario presented in Section 4.1. In Section 4.1 the node is next to the closed gate. In the publication the node is close to arrival area. Also there is network traffic present. Figure taken from my publication [295].

In the publication [295] I conduct a parameter study. I vary three model parameters and investigate their effect on the information dissemination in a WLAN ad hoc network (IEEE 802.11p). The first parameter is the number of agents n, see Table D.1. Each agent represents a mobile node in the mobile network. The network is continuously changing, due to the agents' motion, and so is the information dissemination through the network. A truncated exponential distribution is used for the number of agents with lower bound $lb = 10$ and upper bound $ub = 2000$:

$$n(x, b) = \frac{e^{-x}}{1 - e^b} \tag{D.1}$$

I use $b = 4$, shift the distribution by $lb = 10$ and scale it by $(ub - lb)/b$. A histogram for 2000 samples is depicted in Fig. D.2. The type of distribution is suited to capture the rare event of shadowing that occurs randomly when the number of agents is low. Also, an exponential distribution, while probably not often completely correct, matches certain arrival situations in a train station sufficiently well for our investigation.
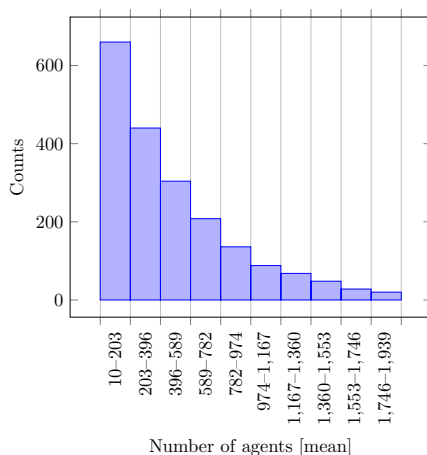
Figure D.2.: Parameter number of agents: histogram. I generate it from a negative exponential distribution. I draw many samples where n is small to detect the effect of shadowing. Figure from my publication [295].

The second parameter is the radio transmitter power p which affects how far agents can communicate. If the power is low, messages only reach agents that are close by. Information dissemination over large distances fails ('range-problem'). The reception in the deployed model is based on the signal to interference plus noise (SINR) ratio. If the SINR of a given transmission exceeds a threshold, the transmission is correctly received. I choose a threshold of of 6 dB to be on the safe side (typically 4 dB are used as default). Furthermore, I assume that the radio transmitter power is equal and constant for all mobile phones within a sample. This is a simplification since in reality, p depends on the local situation and the device itself. I vary the transmitter power in the range

0.5 . . . 2.0 mW Equivalent Isotropically Radiated Power (EIRP).

The third parameter is the network load caused by other apps. While traveling people listen to music, use navigation apps, check their emails. Some even watch movies while walking. To make the simulation more realistic, I assume that agents use apps which produce load in the range of 0...0.20 MB/s. This range contains listening to music (0.01...0.03 MB/s), using navigation apps (0.05 MB/s) and even streaming videos in low-quality (0.06 MB/s). The load tr depends on the packet inter-transmission interval d_{tx} and the packet size s_d

$$tr = s_d \frac{1}{d_{tx}} \tag{D.2}$$

A constant inter-transmission interval is used assuming one-packet messages. I control the packet size s_d to vary the amount of load in the model. The inter-transmission interval d_{tx} is defined by the (re-)broadcast interval, which is fixed to 20 ms. The corresponding packet size s_d is between 0...4000 B. I expect that the information dissemination might be disturbed if the network load is high.

Parameter	Unit	Lower bound	Upper bound	Distribution type
Number of agents n		10	2000	Truncated exponential (Eq. D.1)
Transmitter power p	mW	0.5	2.0	Uniform
Packet size s_d	B	0	4000	Uniform

Dependent parameter	Equation	Unit	Lower Bound	Upper bound	Distribution
Network load tr	see Eq. D.2	MB/s	0	0.20	Uniform

Table D.1.: Uncertain parameters in the network traffic study. I want to to know how three uncertain parameters affect the dissemination time t_{diss}. The packet size s_d controls the amount of network load tr.

I run 20 simulations in parallel on a server with 80 i7 Intel cores and 250 GB RAM. The simulation of 2000 samples takes more than 6 days including two restarts of the script that manages the simulations in parallel.

Effect of shadowing and network load

Fig. D.3 (left) depicts that information is disseminated quickly ($t_{diss} < 10s$) when the number of agents exceeds $n \approx 500$. The higher the network traffic that is indicated by the message size, the longer it takes to disseminate information, see Fig. D.3 (right). Fig. D.4 shows the combined effect of shadowing with network load. When shadowing occurs, agents communicate irregularly and only for short periods. If network load is present during these periods, the information spread can be impeded. The effect seems

to vanish, if the number of agents is large enough, and communication is possible at any time.
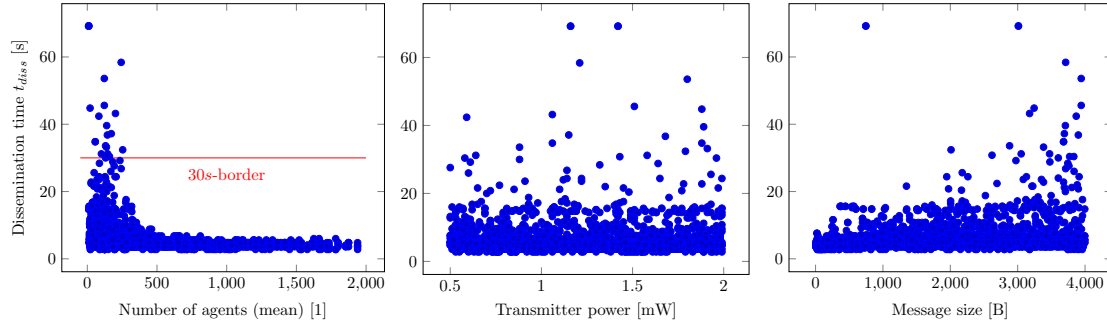
Figure D.3.: Dissemination time t_{diss} in dependency of the number of agents (left), the transmitter power (middle), and the packet size (right) that is proportional to the network load. Only 1.1% of the data points exceed the 30 s-border. Note: The sample values of the two remaining parameters have been projected in the plane. Figure taken from my publication [295].
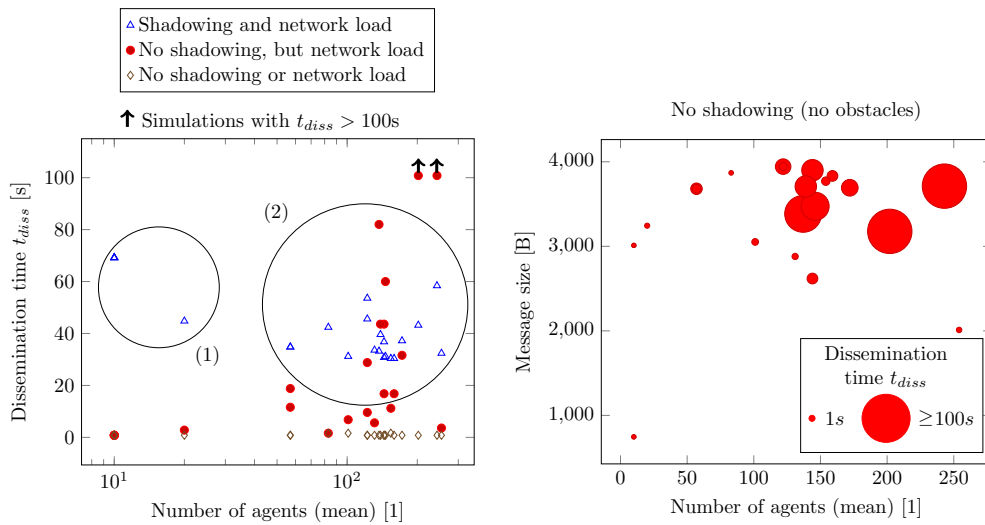
Figure D.4.: Simulations with delayed information dissemination times. Network load can lead to high dissemination times. Figure taken from my publication [295].

E. Capacity estimation (Section 4.2)

To estimate the minimum compliance necessary to avoid congestion in Section 4.2, I employ the jamming criterion that requires the capacity of bottleneck. The capacity can be either measured using a fundamental diagram or taken from handbooks. Fig. depicts the fundamental diagram of the scenario presented in Section 4.2. The densities and velocities were measured in the short route. The specific flow J_S was computed using the relationship between density d and velocity v $J_S = dv$. One observes that the maximum flow is $J_C = 1.58 ped/(ms)$, see Fig. E.1. This value is higher than values from handbooks: Weidmann [51] proposes a value of $1.23 ped/(ms)$, Fruin [311] proposes $1.43 ped/(ms)$, while the SFPE handbook [50] provides a value of $1.30 ped/(ms)$. I think that this peak value is an overestimation of the actual flow caused by the small measurement interval of 0.4s. The average value computed with a polynomial regression model is much lower with $J_C = 1.30 ped/(ms)$ which is in line with the handbook values. Therefore, I use a capacity of $J_C = 1.30 ped/(ms)$ in my flow estimations.
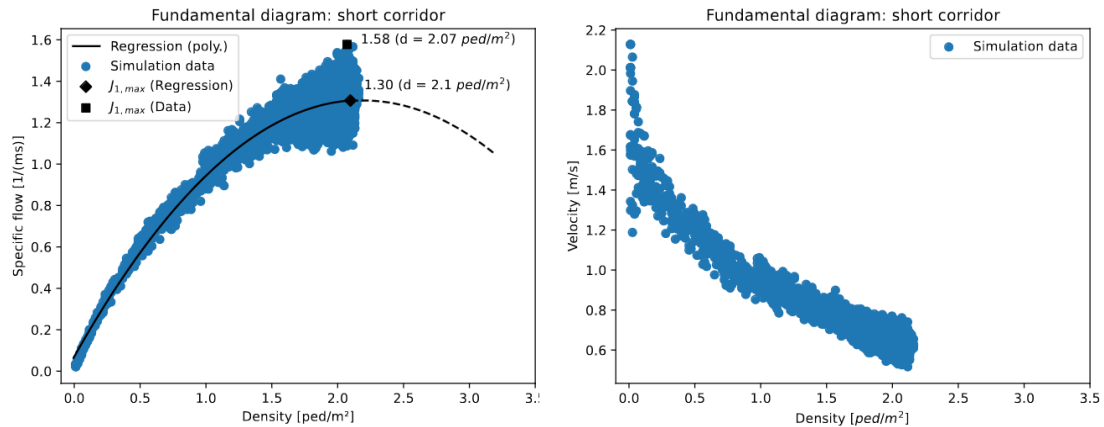
Figure E.1.: Fundamental diagrams for the short route. The specific flow is $J_S = \rho v$. The maximum flow is $J_C = 1.58 ped/(ms)$. Image as in the publication [306]

F. Online survey: ethical approval (Section 4.3)

Ethical approval for the online survey was obtained by the Ethics Committee of Hochschule München University of Applied Sciences (Germany, Munich). The approval comprising the conduction of the survey and the publishing of results. It was approved on 10 August 2021.

Hochschule für angewandte
Wissenschaften München

Fakultät für angewandte
Sozialwissenschaften
Am Stadtpark 20
81243 München

**Prof. Dr. phil. habil.
Stefan Pohlmann**
Vorsitz Ethikkommission
Raum K 235
T +49 89 1265-2300
F +49 89 1265-2330
stefan.pohlmann@hm.edu

Hochschule München · Postfach 200113 · 80001 München

Prof. Dr. Gerta Köster
FK 07
Hochschule München
Lothstr. 34

80335 München

Datum: Dienstag, 10. Austust 2021/
Unser Zeichen: EK008HM–08_21
Antrag bei der Ethikkommission der Hochschule München

Studientitel: Leistungsfähigere Verkehrsinfrastrukturen durch robuste Vernetzung (roVer)
Antragsteller: Frau Prof. Dr. Gerta Köster/ Maria Mayr (Doktorandin)

Sehr geehrte Frau Kollegin Köster

herzlichen Dank für die Einreichung des oben genannten Antrags in der Ethikkommission der
Hochschule München (EKHM). Im Umlaufverfahren haben wir uns in der Kommissionssitzung mit
den erforderlichen Unterlagen befassen können. Nachdem auch die Stellungnahme der
Servicestelle Datenschutz vorliegt, darf ich folgende Entscheidung der Kommission bekannt
geben:

Unter besonderer Berücksichtigung der Stellungnahme des Datenschutzbeauftragten der HM
sieht die Ethikkommission keine Hürden für eine Veröffentlichung der Online-Umfrage in einem
einschlägigen Fachjournal. Wir wären Ihnen dankbar, wenn Sie die ausgewählten Verlage und
Titel dieser Fachzeitschriften noch konkret benennen würden. Weiterhin weisen wir darauf hin,
dass auch bei einer positiven Beurteilung des Vorhabens durch die EKHM die wissenschaftliche
und juristische Verantwortung für die Durchführung und Publikation des Projektes
uneingeschränkt bei der Projektleitung verbleibt. Für die geplante Veröffentlichung wünschen wir
viel Erfolg.

Mit freundlichen Grüßen

Stefan Pohlmann
(Vorsitzender der Ethikkommission)

G. Online survey: collection of statistical tests (Section 4.3)

The following tables contain data from statistical tests and mean values that I refer to in Section 4.3. I have published the tables as supplementary material together with my publication [307]. They are available online under `https://doi.org/10.1371/journal.pone.0284540.s004`.

Prior to information

Route	Group	Route attractiveness			
		mean	median	std	sample size
Long	Fan	2.124	2	0.989	921
Medium	Fan	2.904	3	1.068	921
Short	Fan	4.636	5	0.807	921
Long	Student & faculty associate	2.128	2	1.038	444
Medium	Student & faculty associate	2.631	2	1.042	444
Short	Student & faculty associate	4.732	5	0.700	444

Table G.1.: Route attractiveness prior to information (5-Point Likert scale). Two surveys were conducted: one with students & and faculty associates and one with football fans. For each route, participants were asked how likely it is that they take the respective route when no route recommendation is provided. Since no information is provided, the message design does not have any effect (see Tab. G.3).

Comparison of attractiveness	Group	Z	p
Long route and Medium route	Fan	-11.6806	0.0000
Long route and Short route	Fan	-38.0125	0.0000
Medium route and Short route	Fan	-26.3319	0.0000
Long route and Medium route	Student & faculty associate	-5.4089	0.0000
Long route and Short route	Student & faculty associate	-26.3982	0.0000
Medium route and Short route	Student & faculty associate	-20.9893	0.0000

Table G.2.: Route preference prior to information. For each group, the route attractiveness of the short, medium and long route differ significantly (Dunn's test). In every comparison, the shorter route is preferred ($Z < 0$). I conclude that there is a clear route preference: the short route is favored over the medium route followed by long route (see also the mean values from Tab. G.1).

Information provided

Information	Route	Group	p	H	df
Prior to information	Fans	Long	4.5498	0.7150	7
Prior to information	Fans	Medium	9.4997	0.2190	7
Prior to information	Fans	Short	6.5020	0.4820	7
Prior to information	Students & faculty associates	Long	0.5950	5.5312	7
Prior to information	Students & faculty associates	Medium	0.8430	3.4222	7
Prior to information	Students & faculty associates	Short	0.3660	7.6299	7
After receiving information	Fans	Long	0.0022	22.42	7
After receiving information	Fans	Medium	0.0000	33.04	7
After receiving information	Fans	Short	0.0000	95.76	7
After receiving information	Students & faculty associates	Long	0.8010	3.811	7
After receiving information	Students & faculty associates	Medium	0.0000	32.38	7
After receiving information	Students & faculty associates	Short	0.0000	45.35	7

Table G.3.: Effect of message design on route attractiveness (Kruskal Wallis test). Prior to information (Tab. G.1), the message design has no effect ($p > 0.05$). When information is provided (Tab. G.4), the message design has an effect ($p \leq 0.05$) except for the attractiveness of the long route for the student & faculty associate group.

After receiving information

Route	Group	Message design	Route attractiveness			
			mean	median	std	n
Long	Fan	Arrow	3.427	4	1.340	143
Long	Fan	Arrow + team spirit	3.553	4	1.194	103
Long	Fan	Arrow + top down view	3.550	4	1.270	111
Long	Fan	Arrow + top down view + team spirit	3.743	4	1.259	136
Long	Fan	Congestion info + arrow	3.505	4	1.298	103
Long	Fan	Congestion info + arrow + team spirit	3.980	4	1.219	102
Long	Fan	Congestion info + arrow + top down view	3.897	4	1.160	116
Long	Fan	Congestion info + arrow + top down view + team spirit	3.879	4	1.079	107
Medium	Fan	Arrow	3.266	4	1.156	143
Medium	Fan	Arrow + team spirit	3.078	3	1.135	103
Medium	Fan	Arrow + top down view	3.207	4	1.113	111
Medium	Fan	Arrow + top down view + team spirit	3.066	3	1.206	136
Medium	Fan	Congestion info + arrow	3.534	4	1.119	103
Medium	Fan	Congestion info + arrow + team spirit	3.539	4	1.158	102
Medium	Fan	Congestion info + arrow + top down view	3.655	4	1.039	116
Medium	Fan	Congestion info + arrow + top down view + team spirit	3.542	4	1.093	107
Short	Fan	Arrow	3.643	4	1.286	143
Short	Fan	Arrow + team spirit	3.786	4	1.311	103
Short	Fan	Arrow + top down view	3.414	4	1.254	111
Short	Fan	Arrow + top down view + team spirit	3.265	3	1.312	136
Short	Fan	Congestion info + arrow	2.806	2	1.329	103
Short	Fan	Congestion info + arrow + team spirit	2.471	2	1.224	102
Short	Fan	Congestion info + arrow + top down view	2.784	2	1.357	116
Short	Fan	Congestion info + arrow + top down view + team spirit	2.757	2	1.265	107
Long	Student*	Arrow	3.787	4	1.178	47
Long	Student*	Arrow + team spirit	3.696	4	1.190	56
Long	Student*	Arrow + top down view	3.938	4	1.158	65
Long	Student*	Arrow + top down view + team spirit	3.823	4	1.048	62
Long	Student*	Congestion info + arrow	3.862	4	1.146	58
Long	Student*	Congestion info + arrow + team spirit	3.700	4	1.282	50
Long	Student*	Congestion info + arrow + top down view	4.022	4	1.033	45
Long	Student*	Congestion info + arrow + top down view + team spirit	3.885	4	1.305	61
Medium	Student*	Arrow	2.872	2	1.191	47
Medium	Student*	Arrow + team spirit	2.946	3	1.052	56
Medium	Student*	Arrow + top down view	2.769	2	1.129	65
Medium	Student*	Arrow + top down view + team spirit	2.903	3	1.197	62
Medium	Student*	Congestion info + arrow	3.483	4	1.246	58
Medium	Student*	Congestion info + arrow + team spirit	3.620	4	1.276	50
Medium	Student*	Congestion info + arrow + top down view	3.622	4	1.134	45
Medium	Student*	Congestion info + arrow + top down view + team spirit	3.246	4	1.247	61
Short	Student*	Arrow	3.128	3	1.172	47
Short	Student*	Arrow + team spirit	3.321	4	1.162	56
Short	Student*	Arrow + top down view	3.338	3	1.314	65
Short	Student*	Arrow + top down view + team spirit	3.403	3	1.221	62
Short	Student*	Congestion info + arrow	2.483	2	1.047	58
Short	Student*	Congestion info + arrow + team spirit	2.540	2	1.199	50
Short	Student*	Congestion info + arrow + top down view	2.533	2	1.198	45
Short	Student*	Congestion info + arrow + top down view + team spirit	2.689	2	1.218	61

Table G.4.: Route attractiveness when recommending the long route (5-Point Likert scale). Two surveys were conducted: one with students & faculty associates (student*) and one with football fans. For each route, participants were asked how likely it is that they take the respective route when a route recommendation is provided. The message design always has an effect on the attractiveness of the routes except for the attractiveness of the long route, see Kruskal Wallis tests (see Tab. G.3).

Students & faculty associates

Route	Added component	Message design 1	Message design 2	p	W
Long	Congestion info	Arrow (mean = 3.787)	Congestion info + arrow (mean = 3.862)	0.7592	1317.5
Long	Congestion info	Arrow + top down view (mean = 3.938)	Congestion info + arrow + top down view (mean = 4.022)	0.8557	1434.0
Long	Congestion info	Arrow + team spirit (mean = 3.696)	Congestion info + arrow + team spirit (mean = 3.700)	0.8356	1368.0
Long	Congestion info	Arrow + top down view + team spirit (mean = 3.823)	Congestion info + arrow + top down view + team spirit (mean = 3.885)	0.3327	1708.5
Long	Team spirit	Congestion info + arrow (mean = 3.862)	Congestion info + arrow + team spirit (mean = 3.700)	0.6071	1530.0
Long	Team spirit	Congestion info + arrow + top down view (mean = 4.022)	Congestion info + arrow + top down view + team spirit (mean = 3.885)	0.9810	1376.5
Long	Team spirit	Arrow (mean = 3.787)	Arrow + team spirit (mean = 3.696)	0.6422	1383.5
Long	Team spirit	Arrow + top down view (mean = 3.938)	Arrow + top down view + team spirit (mean = 3.823)	0.3331	2205.5
Long	Top down view	Congestion info + arrow (mean = 3.862)	Congestion info + arrow + top down view (mean = 4.022)	0.5715	1224.5
Long	Top down view	Congestion info + arrow + team spirit (mean = 3.700)	Congestion info + arrow + top down view + team spirit (mean = 3.885)	0.3818	1384.5
Long	Top down view	Arrow (mean = 3.787)	Arrow + top down view (mean = 3.938)	0.4979	1418.5
Long	Top down view	Arrow + team spirit (mean = 3.696)	Arrow + top down view + team spirit (mean = 3.823)	0.6822	1663.0
Medium	Congestion info	Arrow (mean = 2.872)	Congestion info + arrow (mean = 3.483)	**0.0117**	989.0
Medium	Congestion info	Arrow + top down view (mean = 2.769)	Congestion info + arrow + top down view (mean = 3.622)	**0.0003**	884.5
Medium	Congestion info	Arrow + team spirit (mean = 2.946)	Congestion info + arrow + team spirit (mean = 3.620)	**0.0031**	949.5
Medium	Congestion info	Arrow + top down view + team spirit (mean = 2.903)	Congestion info + arrow + top down view + team spirit (mean = 3.246)	0.1243	1598.5
Medium	Team spirit	Congestion info + arrow (mean = 3.483)	Congestion info + arrow + team spirit (mean = 3.62)	0.4871	1341.0
Medium	Team spirit	Congestion info + arrow + top down view (mean = 3.622)	Congestion info + arrow + top down view + team spirit (mean = 3.246)	0.1283	1601.5
Medium	Team spirit	Arrow (mean = 2.872)	Arrow + team spirit (mean = 2.946)	0.6933	1259.5
Medium	Team spirit	Arrow + top down view (mean = 2.769)	Arrow + top down view + team spirit (mean = 2.903)	0.5484	1895.5
Medium	Top down view	Congestion info + arrow (mean = 3.483)	Congestion info + arrow + top down view (mean = 3.622)	0.6519	1240.0
Medium	Top down view	Congestion info + arrow + team spirit (mean = 3.62)	Congestion info + arrow + top down view + team spirit (mean = 3.246)	0.1017	1792.5
Medium	Top down view	Arrow (mean = 2.872)	Arrow + top down view (mean = 2.769)	0.6865	1593.0
Medium	Top down view	Arrow + team spirit (mean = 2.946)	Arrow + top down view + team spirit (mean = 2.903)	0.7897	1783.5
Short	Congestion info	Arrow (mean = 3.128)	Congestion info + arrow (mean = 2.483)	**0.0019**	1804.0
Short	Congestion info	Arrow + top down view (mean = 3.338)	Congestion info + arrow + top down view (mean = 2.533)	**0.0009**	1974.5
Short	Congestion info	Arrow + team spirit (mean = 3.321)	Congestion info + arrow + team spirit (mean = 2.54)	**0.0010**	1895.5
Short	Congestion info	Arrow + top down view + team spirit (mean = 3.403)	Congestion info + arrow + top down view + team spirit (mean = 2.689)	**0.0011**	2514.5
Short	Team spirit	Congestion info + arrow (mean = 2.483)	Congestion info + arrow + team spirit (mean = 2.54)	0.9363	1438.0
Short	Team spirit	Congestion info + arrow + top down view (mean = 2.533)	Congestion info + arrow + top down view + team spirit (mean = 2.689)	0.4671	1270.0
Short	Team spirit	Arrow (mean = 3.128)	Arrow + team spirit (mean = 3.321)	0.4109	1196.5
Short	Team spirit	Arrow + top down view (mean = 3.338)	Arrow + top down view + team spirit (mean = 3.403)	0.7408	1948.0
Short	Top down view	Congestion info + arrow (mean = 2.483)	Congestion info + arrow + top down view (mean = 2.533)	0.9657	1311.0
Short	Top down view	Congestion info + arrow + team spirit (mean = 2.54)	Congestion info + arrow + top down view + team spirit (mean = 2.689)	0.5119	1422.5
Short	Top down view	Arrow (mean = 3.128)	Arrow + top down view (mean = 3.338)	0.4113	1393.0
Short	Top down view	Arrow + team spirit (mean = 3.321)	Arrow + top down view + team spirit (mean = 3.403)	0.6679	1658.5

Table G.5.: Effect of message component on how **students & faculty associates** evaluate the route attractiveness (Mann Whitney U test). If adding a component makes a significant difference ($p < 0.05$) the corresponding p-Value is marked bold. The mean values '(mean = ...)' represent the respective mean values of the route attractivenesses (see Tab. G.4 for the full statistics of the route attractivenesses).

Fans

Route	Added component	Message design 1	Message design 2	p	W
Long	Congestion info	Arrow (mean = 3.427)	Congestion info + arrow (mean = 3.505)	0.6181	7099.5
Long	Congestion info	Arrow + top down view (mean = 3.55)	Congestion info + arrow + top down view (mean = 3.897)	**0.0331**	5433.0
Long	Congestion info	Arrow + team spirit (mean = 3.553)	Congestion info + arrow + team spirit (mean = 3.980)	**0.0047**	4107.0
Long	Congestion info	Arrow + top down view + team spirit (mean = 3.743)	Congestion info + arrow + top down view + team spirit (mean = 3.879)	0.6983	7076.0
Long	Team spirit	Congestion info + arrow (mean = 3.505)	Congestion info + arrow + team spirit (mean = 3.980)	**0.0069**	4157.0
Long	Team spirit	Congestion info + arrow + top down view (mean = 3.897)	Congestion info + arrow + top down view + team spirit (mean = 3.879)	0.6011	6443.0
Long	Team spirit	Arrow (mean = 3.427)	Arrow + team spirit (mean = 3.553)	0.5517	7048.5
Long	Team spirit	Arrow + top down view (mean = 3.55)	Arrow + top down view + team spirit (mean = 3.743)	0.1957	6856.0
Long	Top down view	Congestion info + arrow (mean = 3.505)	Congestion info + arrow + top down view (mean = 3.897)	**0.0305**	5006.0
Long	Top down view	Congestion info + arrow + team spirit (mean = 3.980)	Congestion info + arrow + top down view + team spirit (mean = 3.879)	0.1645	6028.0
Long	Top down view	Arrow (mean = 3.427)	Arrow + top down view (mean = 3.55)	0.5125	7570.5
Long	Top down view	Arrow + team spirit (mean = 3.553)	Arrow + top down view + team spirit (mean = 3.743)	0.1652	6299.0
Medium	Congestion info	Arrow (mean = 3.266)	Congestion info + arrow (mean = 3.534)	0.0731	6423.5
Medium	Congestion info	Arrow + top down view (mean = 3.207)	Congestion info + arrow + top down view (mean = 3.655)	**0.0021**	5006.5
Medium	Congestion info	Arrow + team spirit (mean = 3.078)	Congestion info + arrow + team spirit (mean = 3.539)	**0.0038**	4087.5
Medium	Congestion info	Arrow + top down view + team spirit (mean = 3.066)	Congestion info + arrow + top down view + team spirit (mean = 3.542)	**0.0022**	5678.0
Medium	Team spirit	Congestion info + arrow (mean = 3.534)	Congestion info + arrow + team spirit (mean = 3.539)	0.8731	5188.5
Medium	Team spirit	Congestion info + arrow + top down view (mean = 3.655)	Congestion info + arrow + top down view + team spirit (mean = 3.542)	0.4670	6535.0
Medium	Team spirit	Arrow (mean = 3.266)	Arrow + team spirit (mean = 3.078)	0.1820	8065.5
Medium	Team spirit	Arrow + top down view (mean = 3.207)	Arrow + top down view + team spirit (mean = 3.066)	0.3322	8066.0
Medium	Top down view	Congestion info + arrow (mean = 3.534)	Congestion info + arrow + top down view (mean = 3.655)	0.4556	5644.5
Medium	Top down view	Congestion info + arrow + team spirit (mean = 3.539)	Congestion info + arrow + top down view + team spirit (mean = 3.542)	0.9011	5508.5
Medium	Top down view	Arrow (mean = 3.266)	Arrow + top down view (mean = 3.207)	0.6396	8195.5
Medium	Top down view	Arrow + team spirit (mean = 3.078)	Arrow + top down view + team spirit (mean = 3.066)	0.9352	7045.5
Short	Congestion info	Arrow (mean = 3.643)	Congestion info + arrow (mean = 2.806)	**0.0000**	9855.0
Short	Congestion info	Arrow + top down view (mean = 3.414)	Congestion info + arrow + top down view (mean = 2.784)	**0.0003**	8143.0
Short	Congestion info	Arrow + team spirit (mean = 3.786)	Congestion info + arrow + team spirit (mean = 2.471)	**0.0000**	7940.5
Short	Congestion info	Arrow + top down view + team spirit (mean = 3.265)	Congestion info + arrow + top down view + team spirit (mean = 2.757)	**0.0023**	8857.5
Short	Team spirit	Congestion info + arrow (mean = 2.806)	Congestion info + arrow + team spirit (mean = 2.471)	0.0634	5993.5
Short	Team spirit	Congestion info + arrow + top down view (mean = 2.784)	Congestion info + arrow + top down view + team spirit (mean = 2.757)	0.9759	6192.0
Short	Team spirit	Arrow (mean = 3.643)	Arrow + team spirit (mean = 3.786)	0.3306	6851.5
Short	Team spirit	Arrow + top down view (mean = 3.414)	Arrow + top down view + team spirit (mean = 3.265)	0.3873	8015.0
Short	Top down view	Congestion info + arrow (mean = 2.806)	Congestion info + arrow + top down view (mean = 2.784)	0.8446	6060.5
Short	Top down view	Congestion info + arrow + team spirit (mean = 2.471)	Congestion info + arrow + top down view + team spirit (mean = 2.757)	0.0781	4740.5
Short	Top down view	Arrow (mean = 3.643)	Arrow + top down view (mean = 3.414)	0.1237	8799.5
Short	Top down view	Arrow + team spirit (mean = 3.786)	Arrow + top down view + team spirit (mean = 3.265)	**0.0024**	8555.5

Table G.6.: Effect of message component on how **fans** evaluate the route attractiveness (Mann Whitney U test). If adding a component makes a significant difference ($p < 0.05$) the corresponding p-Value is marked bold. The mean values '(mean = ...)' represent the respective mean values of the route attractivenesses (see Tab. G.4 for the full statistics of the route attractiveness)

Students & faculty associates: Route attractiveness (mean value)

Route	Condition	Prior to info.	Info. provided	p	W
Long	Arrow + top down view	2.062	3.938	**0.0000**	3654.5
Long	Arrow + team spirit	2.196	3.696	**0.0000**	2542.5
Long	Congestion info + arrow	2.241	3.862	**0.0000**	2758.0
Long	Arrow + top down view + team spirit	2.145	3.823	**0.0000**	3245.5
Long	Arrow	2.149	3.787	**0.0000**	1834.0
Long	Congestion info + arrow + top down view + team spirit	2.016	3.885	**0.0000**	3133.5
Long	Congestion info + arrow + top down view	2.311	4.022	**0.0000**	1720.5
Long	Congestion info + arrow + team spirit	1.940	3.700	**0.0000**	2105.0
Medium	Arrow + top down view	2.708	2.769	0.7852	2168.0
Medium	Arrow + team spirit	2.643	2.946	0.1015	1825.5
Medium	Congestion info + arrow	2.672	3.483	**0.0004**	2298.5
Medium	Arrow + top down view + team spirit	2.677	2.903	0.2906	2122.0
Medium	Arrow	2.723	2.872	0.5057	1185.5
Medium	Congestion info + arrow + top down view + team spirit	2.607	3.246	**0.0045**	2391.0
Medium	Congestion info + arrow + top down view	2.600	3.622	**0.0000**	1503.5
Medium	Congestion info + arrow + team spirit	2.380	3.620	**0.0000**	1924.0
Short	Arrow + top down view	4.754	3.338	**0.0000**	847.0
Short	Arrow + team spirit	4.732	3.321	**0.0000**	465.0
Short	Congestion info + arrow	4.603	2.483	**0.0000**	284.0
Short	Arrow + top down view + team spirit	4.806	3.403	**0.0000**	644.0
Short	Arrow	4.681	3.128	**0.0000**	342.0
Short	Congestion info + arrow + top down view + team spirit	4.787	2.689	**0.0000**	357.5
Short	Congestion info + arrow + top down view	4.667	2.533	**0.0000**	223.0
Short	Congestion info + arrow + team spirit	4.800	2.540	**0.0000**	204.0

Table G.7.: Effect of information provision on how **students & faculty associates** evaluate the route attractiveness (Mann Whitney U test). If information provision makes a significant difference ($p < 0.05$) the corresponding p-Value is marked bold. Through information the long route becomes always more attractive (the means increase), while the short route always becomes less attractive (the means reduce). Also, the medium route becomes more attractive. The increase of the means is not always significant.

Fans: Route attractiveness (mean value)

Route	Condition	Prior to info.	Info. provided	p	W
Long	Arrow + top down view	2.207	3.550	**0.0000**	9524.0
Long	Arrow + team spirit	2.146	3.553	**0.0000**	8471.5
Long	Congestion info + arrow	1.981	3.505	**0.0000**	8584.5
Long	Arrow + top down view + team spirit	2.125	3.743	**0.0000**	15171.0
Long	Arrow	2.035	3.427	**0.0000**	15846.0
Long	Congestion info + arrow + top down view + team spirit	2.131	3.879	**0.0000**	9755.0
Long	Congestion info + arrow + top down view	2.259	3.897	**0.0000**	11034.0
Long	Congestion info + arrow + team spirit	2.118	3.980	**0.0000**	8914.0
Medium	Arrow + top down view	2.793	3.207	**0.0053**	7424.5
Medium	Arrow + team spirit	2.816	3.078	0.0991	5969.0
Medium	Congestion info + arrow	2.718	3.534	**0.0000**	7383.0
Medium	Arrow + top down view + team spirit	3.029	3.066	0.8674	9352.0
Medium	Arrow	2.909	3.266	**0.0045**	12116.5
Medium	Congestion info + arrow + top down view + team spirit	3.028	3.542	**0.0005**	7214.5
Medium	Congestion info + arrow + top down view	2.974	3.655	**0.0000**	8926.5
Medium	Congestion info + arrow + team spirit	2.922	3.539	**0.0001**	6799.5
Short	Arrow + top down view	4.523	3.414	**0.0000**	2913.5
Short	Arrow + team spirit	4.670	3.786	**0.0000**	3165.0
Short	Congestion info + arrow	4.583	2.806	**0.0000**	1690.5
Short	Arrow + top down view + team spirit	4.669	3.265	**0.0000**	3666.5
Short	Arrow	4.692	3.643	**0.0000**	5235.0
Short	Congestion info + arrow + top down view + team spirit	4.654	2.757	**0.0000**	1507.5
Short	Congestion info + arrow + top down view	4.560	2.784	**0.0000**	2294.0
Short	Congestion info + arrow + team spirit	4.725	2.471	**0.0000**	935.5

Table G.8.: Effect of information provision on how **football fans** evaluate the route attractiveness (Mann Whitney U test). If information provision makes a significant difference ($p < 0.05$) the corresponding p-Value is marked bold. Through information the long route becomes always more attractive (the means increase), while the short route always becomes less attractive (the means reduce). Also, the medium route becomes more attractive. The increase of the means is not always significant.

H. Familiarity with environment (Section 4.3)

Findings from the online survey strongly indicate that people are basically willing to follow instructions provided by mobile applications. However, the survey assumed that people are familiar with the environment.

To find out how well people can follow the recommendation under real environmental conditions, I conducted an on-site survey at the metro station Münchner Freiheit (Munich, Germany) on 26 September 2023 (14:00-15:00) at its surface level. The procedure was as follows:

- Approach people (say: I'm from Munich University of Applied Sciences and I'm conducting a scientific survey together with MVG. One question for you - it will only take 10 seconds.)

- Then show the picture (see fig. H.1) on a tablet and ask: "How well can you find the way. Please rate with a school grade."

- Note the school grade.

18 people took part in the survey. 17 out of 18 people provided an answer. 11 out of 17 people provided information on their local knowledge without being asked.
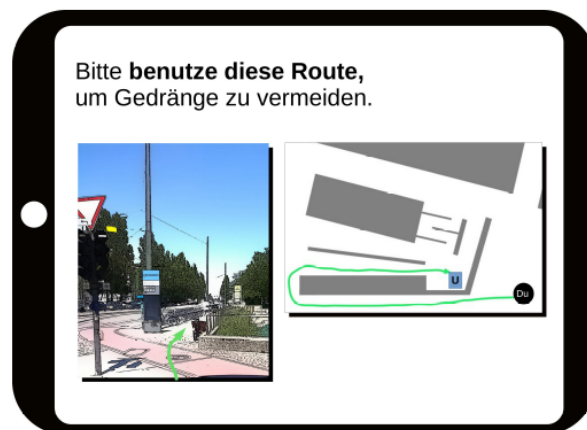

Figure H.1.: Message design

The survey indicates that people familiar with the area can easily find the recommended route, see Fig. H.2. All 6 people with local knowledge gave the school grade "very good" (1). People who are not familiar with the area have problems finding the route (rating:

4 or 5). However, 2 out of 5 people found the non-recommended direct route thanks to the signposting on site. These two people also stated that they would find it helpful if the suggested route was signposted. No tendency can be observed for people who did not indicate their local knowledge. 2 out of 6 people found it easy to find the route ("very good"). 2 out of 6 people think that finding the route is challenging ("satisfactory"). 2 out of 6 people find it difficult (grade 4 or 6).

The survey sample was small and is not representative. For a representative study, a much larger number of survey participants would be necessary, as well as the recording of other factors such as day of the week, time of day, age distribution, etc. Nevertheless, a tendency can already be identified from this small sample. People who are familiar with the area seem to have no problems finding the recommended route. People who are not familiar with the area may have problems finding their way. One way to make orientation on the surface of Münchner Freiheit generally easier would be to signpost alternative routes with the aid of static signs. If an app were to be developed, orientation could be supported, for example, with a live view such as Google Maps Live View.
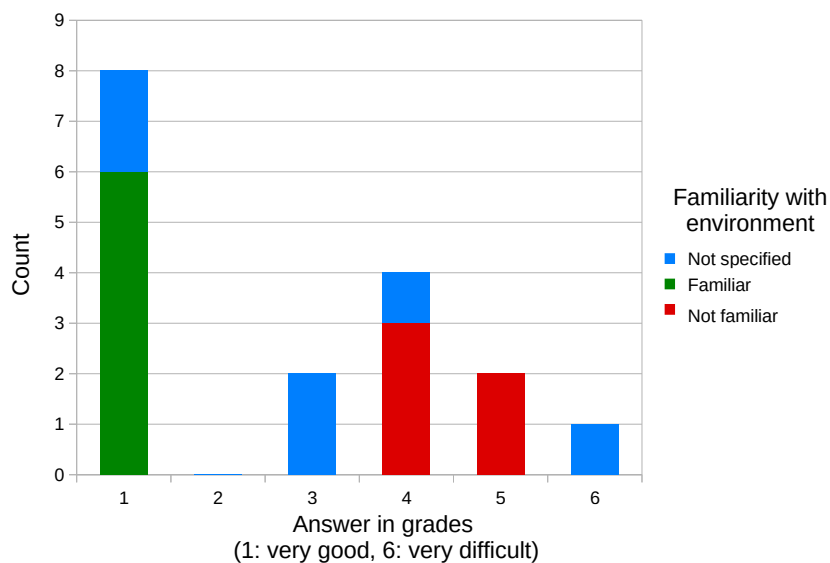

Figure H.2.: Effect of familiarity on navigation.

I. Pedestrian streams at Münchner Freiheit (Section 4.4)

On 29th Oct 2020 I conducted a counting study at the metro station Münchner Freiheit (Munich, Germany) [312]. To carry out the counting experiment, I had obtained permission from the local transport provider Münchner Verkehrsgesellschaft mbH (MVG) in Munich, Germany. Fig. I.1 depicts a map of the intermediate floor of the metro station.
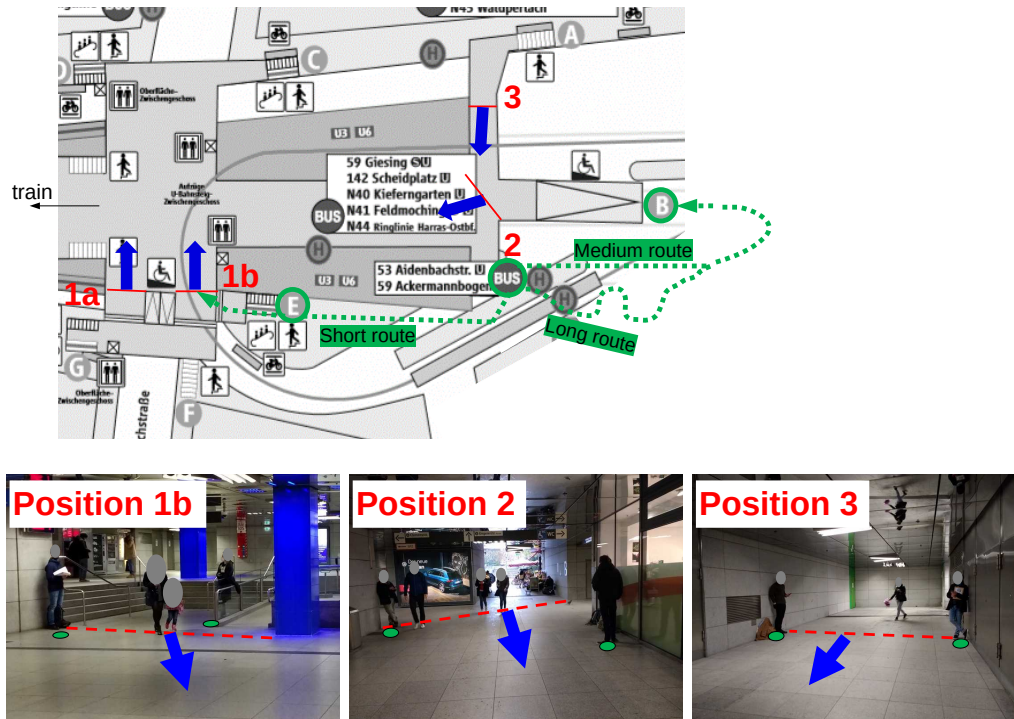

Figure I.1.: Counting experiment: location of the measurement lines. Students clicked an app whenever a person crossed an imaginary line towards a certain direction. The map in the background (top) was taken from https://efa.mvv-muenchen.de/sta/muenchnerfreiheit.pdf (16th Jan. 2023). I added arrows and routes. Own photographs (bottom).

Students counted pedestrians at various points on the intermediate floor of the station during the afternoon in three time slots of each 30 minutes. Students clicked an app

whenever a person crossed an imaginary line towards a certain direction.

Arrivals over time

Fig. I.2 depicts the results of the counting experiment. I would like to mention that the counting experiment took place during a lull in the Covid pandemic. Hence, the measured flows might be lower than before or after the pandemic. The flow at the measurement line 1b is always larger than the flow at the other measurement line. Unfortunately, I do not know where the arrivals come from. The arrivals measured at line 1b can either come from the bus or side entrances. One can observe that for the measurement line 1b, the number of arrivals varies over time.
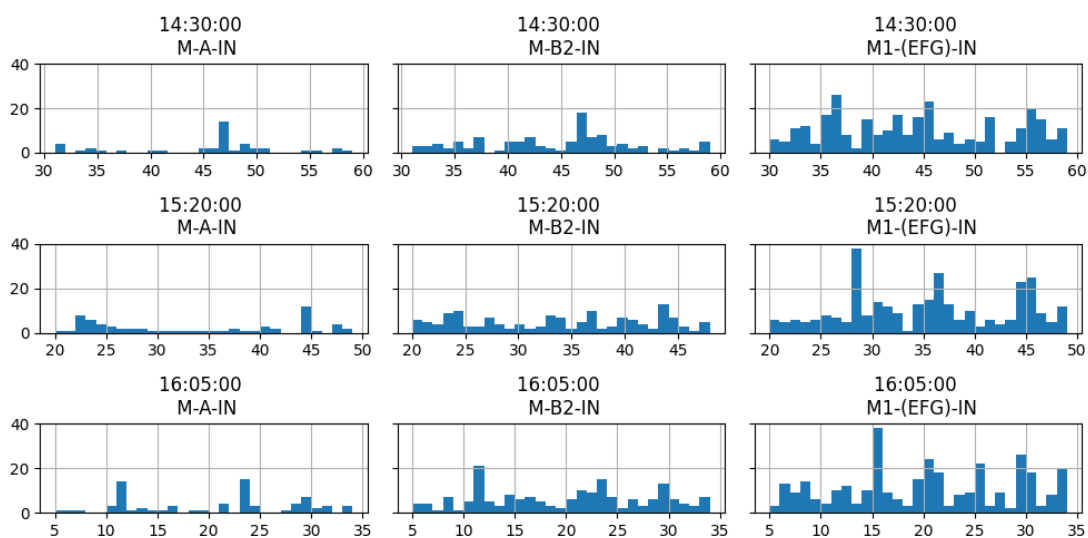

Figure I.2.: Counting experiment: pedestrian flows over time

Estimating the probability to take the short route

When people walk from the bus to the metro, the direct short route congestion can occur in the intermediate floor which is not visible at the surface level. I want to know how likely it it is that passengers take the short route and experience congestion. I suggest to estimate the probability from the count data. I assume that the probability to take the short route is the ratio of the flow along the short route and the in-flow:

$$p_{short} = \frac{flow_{short}}{flow_{in}} \tag{I.1}$$

The in-flow is defined as the sum of all flows of arriving passengers at the surface level, that are, arrivals from the bus or tram. I assume that the inflow is the sum of flows

measured at measurement line 1b and 2. Please note that this only holds for a steady state flow and does not consider additional flows from side entrances.

Tab. I.1 shows the route choice probabilities for the short route. Please be aware that there is uncertainty in this values: there can be other flows involved from side entrances which means the values can be either higher or lower depending on the strength of the unknown side flows. Also, the counting experiment was conducted during a lull in the corona pandemic. This mostly affects the absolute flow values, but I cannot guarantee that it does not shift the distribution of pedestrians over the routes.

Time slot	M-A-IN	M-B2-IN	M1-(EFG)-IN	M-B-IN-cleaned	Probability: short route
14:30:00	43	111	302	68	0.82
15:20:00	65	147	307	82	0.79
16:05:00	72	179	329	107	0.75

Table I.1.: Counting experiment: route probability of the short route for validation. Total number of arrivals for each time slot and corresponding probability for the short route.

Interestingly, the probability of the short route decreases over the number of arrivals, see Fig. I.3. This indicates that the more people, the more congestion and the more likely people take alternative routes. However, I cannot provide statistical evidence with only three data points.
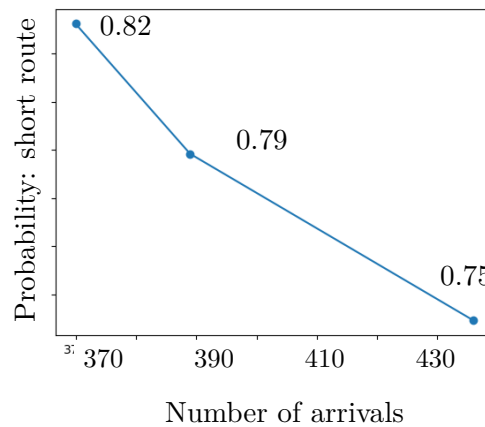

Figure I.3.: Estimated probability for the short route over arrivals. The estimated probability to take the short route decreases over the number of arrivals.

Last page of the document