# Evacuation Planning under Selfish Evacuation Routing

Der Mercator School of Management, Fakultät für Betriebswirtschaftslehre der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades

eines Doktors der Wirtschaftswissenschaft (Dr. rer. oec.)

genehmigte Dissertation

von

Kerstin Seekircher

aus

Neuss

Referent: Prof. Dr. Alf Kimms
Korreferent: Prof. Dr. Jochen Gönsch
Tag der mündlichen Prüfung: 09.11.2017

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

A growing number of natural disasters like floods, hurricanes, or earthquakes and man-made disasters like chemical or nuclear accidents (EM-DAT (2017)) are the reason for an increasing relevance of disaster management. Disaster management includes a multitude of strategies to prevent catastrophes and the arising negative effects for the residents in the affected area. These strategies include for instance land use control to prevent that floods affect residential areas, relief delivery to take care of the victims after a catastrophe or re-building activities in the affected area. All these strategies and instruments must be in place prior to the occurrence of the catastrophe thus a well organised disaster management is necessary. In the last decades many studies in different disciplines of disaster management were executed, which consider various aspects and perspectives of this research area. Also, a significant increase in the number of publications in operations research journals dealing with disaster management issues can be observed (Altay and Green (2006)). This points out the increasing interest and relevance of operations research in disaster management.

In the context of disaster management evacuation of the affected area represents an important instrument to handle a dangerous situation. This instrument can be used in case of noticed- and no-noticed catastrophes. In case of noticed catastrophes like for instance floods or earthquakes many lives can be saved if the endangered area is cleared prior to the occurrence of the catastrophe. But also in case of no-notice catastrophes, e.g. a nuclear accident, evacuation of the affected area can save the lives of many people and minimises the magnitude of the catastrophe. Since early works by Sheffi et al. (1982) and Sinuany-Stern and Stern (1993) the area of research which uses methods of operations research for evacuation management has rapidly grown. Various strategies are developed to optimise the evacuation of affected areas. These strategies include aspects like traffic management, shelter positioning, and evacuation timing. In the area of traffic management a common strategy is the use of contra-flow which was successfully tested when Hurricane Rita reached in the Gulf of Mexico in 2005 (Sangho et al. (2008)). Other studies determine optimal routes to guide the residents out of the affected area, strategies that include public traffic management or methods of staged evacuation that are used to prevent traffic congestion by evacuating the affected area in different time intervals. In this research field optimisation models as well as simulation model are used (Özdamar and Ertem (2015)). Most of these studies assume that people follow the proposed evacuation strategies (for more details see the review of the related literature in Chapter 2). This assumption is critical because evacuation plans may fail when the people do not behave in the expected way. Abdelgawad and Abdulhai (2013) point out that evacuees behave selfishly in large evacuations scenarios which

leads to user-optimal decisions. This assumption is supported in studies by Murray-Tuite et al. (2012) and Sadri et al. (2014). These studies interviewed people that were involved in hurricane evacuations and, besides others, evaluated the evacuee's route choice decisions. Most of the evacuees stated, that they chose a familiar way and only a small part followed the instructions of authorities. Hence, a more realistic assumption is that evacuees tend to selfish behaviour during evacuations. To handle realistic evacuation scenarios, the behaviour of people must be considered in the evacuation plan. To the best of our knowledge there are just a few studies that develop strategies for evacuation traffic management and take the selfish behaviour of evacuees into account, for example Madireddy et al. (2011) and Huibregtse et al. (2012).
In this thesis the research field of traffic flow optimisation for evacuation scenarios is linked with the research area of selfish routing in traffic networks. As mentioned above, just a few studies have considered this combination before. Accordingly, a research gap exits, which will be closed by this thesis, as it studies the effects of selfish routing in evacuation scenarios and develops strategies to counteract these effects.
With a combination of these two research areas it is hypothesised that the residents of an endangered area behave selfishly. As the behaviour of people in evacuation scenarios is studied in the literature it seems to be a relevant problem during evacuation scenarios. The literature of traffic management (e.g. Braess (1968), Roughgarden and Tardos (2002), Cole et al. (2003)) reveals that selfish behaviour has negative effects on the traffic flow. If people choose their routes without considering the route choice of the other network users, the traffic flow will decrease and the travel time in the network increases. This thesis analyses whether these results can be transferred to an evacuation scenario or not. Methods will be introduced that counteract the observed effects and lead to an increase of traffic flow, which in turn reduces the evacuation time. Known from the literature most people will take routes which they assume as best for themselves in evacuation scenarios. In such situations people prefer familiar routes they know from daily life and these are in most cases the shortest or fastest ones (Sadri et al. (2014)). Thus many people will take the same routes which leads to congestions. The increase of traffic flow can be achieved by forcing the evacuees to take alternative routes. In this thesis a method to regulate the selfish evacuation traffic is developed on the basis of the Braess paradox (Braess (1968)). In his paradox, Braess describes that the decrease of travel time in a network is achieved by adding a time decreasing arc. The paradox occurs because the network users act selfishly and take the best route according to their own estimates. So all network users take the same path which results in congestions and in an increase of travel time for all network users (for a more detailed description the reader is referred to Chapter 4). In this thesis these findings are used the other way around and street blockages are installed to force the evacuees to use alternative routes to optimise the overall traffic flow. The consideration that people follow proposed evacuation strategies can be ruled out with this method and hence it is not necessary to consider expected behaviour patterns any more. An additional advantage is that the communication effort can be reduced in comparison to the effort that is necessary when dealing with the optimal routes for the network users. To implement this approach only the position of the street blockages must be communicated and the evacuees choose their selfish routes according to the optimised network.

The thesis is structured as follows: In Chapter 2 general aspects of evacuation planning are discussed. After a short introduction to disaster management and a discussion of the relevance on this topic in Section 2.1, Section 2.2 summarises the relevant literature of evacuation traffic management. Section 2.3 illustrates the assumptions made for the evacuation scenario in this thesis. Chapter 3 introduces the cell transmission model (CTM), which is applied throughout this thesis to model the traffic flow. In Section 3.1 the basic idea of the CTM is introduced and the relevant assumptions are stated. Afterwards, in Section 3.2 the use of the CTM in evacuation scenarios is explained. Herein, the related literature is summarised first, second a basic model for evacuation traffic optimisation is introduced. Chapter 4 summarises the main aspects of selfish routing in traffic networks. In Section 4.1 the consequences of selfish routing in traffic networks are discussed and relevant terms and definitions are introduced. In conclusion the related literature is presented. Section 4.2 points out and discusses Braess's paradox, which illustrates the negative influence of selfish routing in traffic networks. Besides the general phenomenon, the relevant literature is discussed. The chapter closes with a summary of the main aspects of selfish routing and emphasises the relevance of these aspects with regard to evacuation traffic management. In Chapter 5 selfish routing and evacuation traffic management are combined (Section 5.1) and the test bed used for the computational studies is presented (Section 5.2). Section 5.1.1 illustrates the selfish route selection of evacuees and characterises the evacuee types considered in this thesis. Section 5.1.2 introduces the general concept of blocking street sections, which is specified later in Chapters 6 to 8. In Section 5.2 the used test bed is introduced and reference values are discussed. In Chapter 6 the concept of using sub-networks to guide the selfish evacuation traffic is introduced. The general idea of this concept is illustrated in Section 6.1 and an approach on how to determine these sub-networks is proposed. In Section 6.2 a mathematical model is presented that can be used to compute such sub-networks and in Section 6.3 a heuristic is proposed that can be utilised to compute such sub-networks for networks of realistic size. In Section 6.4 variants of the construction approach, which was presented in Section 6.1, are introduced. The chapter closes with a comprehensive computational study. Chapter 7 delineates a method that uses specific street blockages to guide the selfish evacuation traffic. The problem is formulated as a bi-level model, therefore in Section 7.1 a short introduction to bi-level optimisation is given. Section 7.2 presents a mathematical model for the computation of street blockages taking selfish routing evacuees into account. First, the modelling assumptions are discussed and than the upper- and lower level problems are introduced. In Section 7.3 three solution approaches are presented, that can be used to solve the model formulation stated above. In the last section of this chapter the presented approaches are tested in a computational study. Chapter 8 investigates the minimisation of street blockages which are necessary to guide the traffic out of the affected area. To minimise the number of blockages a second objective function is proposed for the mathematical model which is presented in Chapter 8. Section 8.1 summarises the general idea of multi-objective optimisation and presents some relevant solution methods. Afterwards, the relevance of the second objective function is discussed and in Section 8.3 a solution approach that is based on lexicographical optimisation is developed. In Section 8.4 a solution procedure that is based on the $\varepsilon$-constraint method is illustrated. Following, the presented methods are tested in a computational study and the results are analysed. Chapter 9 summarises the main aspects of this thesis and discusses the potential of further research.

# Chapter 2

# Evacuation planning

Disaster management is an emerging topic in a world with a growing number of natural and technological disasters. In this chapter the most important aspects of disaster and evacuation management in the context of operations research are discussed. Section 2.1 gives an overview of the phases of the disaster management life cycle and the corresponding activities. Afterwards, the stages of evacuation management are explained and the related literature of each stage is presented. In Section 2.2 a comprehensive review of the literature in management of evacuation traffic is given. The chapter closes with Section 2.3, where the evacuation scenario which is relevant for this thesis will be presented.

## 2.1 Disaster Management

The number of disasters is rapidly growing since the 1960s: 40 documented natural disasters in 1960 to 526 documented disasters in the year 2000. The same trend can be observed for technical disasters (EM-DAT (2017)). This development leads to the necessity of well-prepared disaster management, where predefined activities should be performed during, before, and after a disaster. The goal of the disaster management is to prevent the loss of human life, to reduce the impact of the disaster, and to return the affected area into it's original state (Altay and Green (2006)). The disaster management consists of different phases, which result in a disaster management life cycle. However, there is no unified definition on this matter, and the number of phases in the cycle as well as their definition varies depending on the study. One concept with four phases based on the Comprehensive Emergency Management concept was introduced in 1978 and reported in the National Governor's Associated Emergency Preparedness Project. This concept contains the mitigation, preparedness, response, and recovery phases (Altay and Green (2006)). A classification with three phases is presented by Özdamar and Ertem (2015), including (pre-disaster) preparedness, (post-disaster) response and recovery phase. Both concepts contain the same activities, just the allocation to the phases is different. Figure 2.1 depicts the disaster management life cycle with three phases: catastrophe management can be seen as cycle because the phases are repeated continuously. After the consequences of a disaster are removed further activities are necessary to prevent the next disaster. This thesis focuses on the three phases concept and discusses the most important activities as presented by Altay and Green (2006) and Özdamar and Ertem (2015).

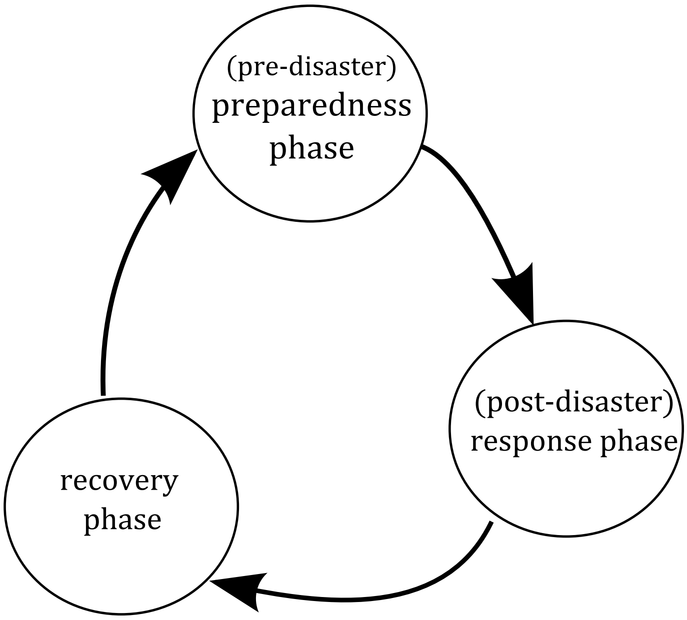



Figure 2.1: Disaster Management Life Cycle.

The preparedness phase includes all activities which either prevent that natural/technical phenomena result in catastrophes or to reduce the impact of a disaster. In order to lower the consequences for example barriers can be constructed, land use can be controlled or building codes can be made to improve disaster resistance of structures (Altay and Green (2006)). Also, activities of inventory, equipment and shelter pre-positioning are included in this phase. Especially a well prepared pre-positioning of shelters results in a coordinated evacuation (Özdamar and Ertem (2015)). The response phase contains all activities that are required to reduce the impact of an occurred disaster: evacuation of the affected area, emergency rescue and medical care, relief delivery, debris collection or road cleaning are just some of the activities (Altay and Green (2006), Özdamar and Ertem (2015)). In the recovery phase, the destroyed area is restored in the original state and humanitarian help is provided to the victims of the catastrophe. The resulting activities are e.g. re-building of infrastructure, roads and key buildings (Altay and Green (2006)). In the last decades, the topic of catastrophe management and humanitarian logistics became more and more important in the field of operations research. A lot of research is done about topics in the different phases of the disaster management life cycle. For a comprehensive collection of different approaches and topics in humanitarian logistics the reader is referred to Altay and Green (2006), Van Wassenhove and Pedraza Martinez (2012) and Özdamar and Ertem (2015).

One part of disaster management and an effective instrument to keep the consequences of a disaster low is the evacuation of the affected area. As mentioned above, the main activities of evacuation planning can be assigned to the (post-disaster) response phase, but there are also activities that can be allocated to the (pre-disaster) preparedness and recovery phase. That the instrument of evacuation is used in all phases, points out the importance of evacuation in disaster management. To get a better understanding of evacuation management first evacuation is defined in general and afterwards the stages of evacuation planning and the resulting activities are explained.

According to Müller (1998) an evacuation is defined as the organised relocation of humans and

animals from an affected area with transport, shelter and necessary supplies. An evacuation is not a short term instrument of disaster management and considers more than the transport of people out of the affected area. Before the relocation of people can take place the evacuation area has to be defined and shelters have to be determined. Moreover, evacuation includes the care of people in the safe areas and the transportation back to their homes when the disaster has ended (Müller (1998)).
Figure 2.2 depicts the stages of evacuation planning processes based on ISO 22315 (ISO22315 (2014)), which illustrates all activities that will be performed in an evacuation. According to the definition Müller (1998) and the activities presented in the stages of evacuation management, each phase of the disaster management life cycle concerns aspects of evacuation planning.

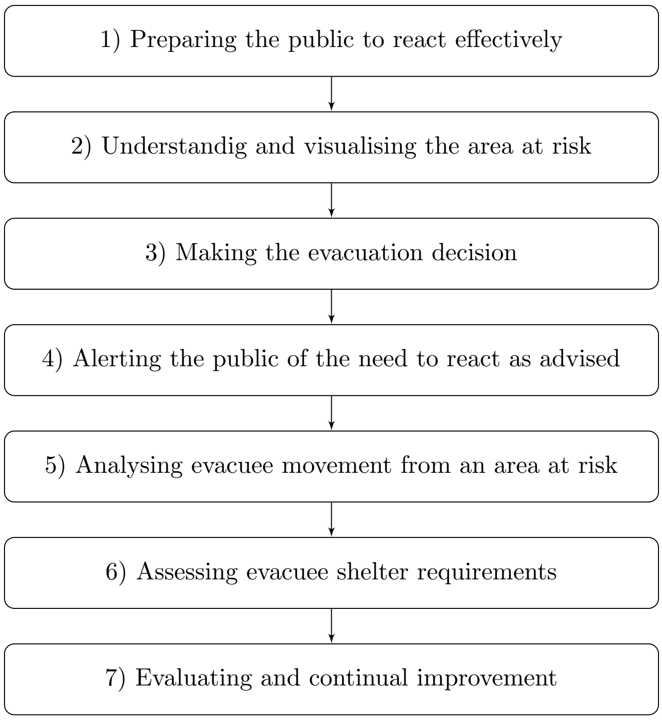



Figure 2.2: Stages of Evacuation Management (Based on ISO22315 (2014)).

Afterwards, the stages of an evacuation process are assigned to the phases of the disaster management life cycle. For each stage the main activities are explained and for the stages that belong to the response phase the related literature is presented.
Stage one can be assigned to the preparedness phase and includes all activities that can be done to be prepared for an evacuation. These activities are for example: evacuation or fire protection training in schools or public buildings, prepared and communicated evacuation plans for instance in regions next to nuclear power plants, alignment of signs which depict escape routes in buildings. These and further activities are possible to prepare the public for an evacuation.
The stages two to six can be assigned to the response phase and include all activities that are necessary directly before, while and after the evacuation.
In stage two the evacuation area is defined, which is really important for the evacuation planning process. In this step the research can be divided into two main parts: the evacuation of buildings and transportation means and the evacuation of urban areas. Since this thesis focuses

on the evacuation of urban areas only, the evacuation of transportation means and buildings will not be discussed in detail. However, for reasons of completeness, a rough overview of studies in building evacuation are presented. A simulation model for the evacuation of buildings by fire is introduced by Shen (2005). With a simulation model the "evacuability" of a building can be estimated. The "evacuability" is defined as the percentage of people that can be successfully evacuated from the building. A comprehensive review of simulation models for the evacuation of build environments is given in Gwynne et al. (1999). Chalmet et al. (1982) present an early optimisation model for the evacuation of buildings. The problem is modelled as a time-depended network flow problem. They present an approach to minimise the total evacuation time and to identify bottlenecks of the evacuation. Like in Shen (2005) they also analyse the "evacuability" of buildings and present evacuation relevant aspects of building design and redesign. Choi et al. (1988) also present a network flow model for building evacuation. They study maximum flow, minimum cost and minmax objectives. In their model they have side constraints, which consider for example variable arc capacities. For networks with "special" structure they use a greedy algorithm to solve the model. For further aspects of optimisation models for evacuation of buildings the reader is referred to Hamacher and Tjandra (2001).
For the evacuation of urban areas it is more complicated to define the affected area. On the one hand, the geographical area has to be determined, and on the other hand, the evacuation demand has do be estimated. The determination of the affected area highly depends on the disaster type. For regions with critical industry like nuclear power plants or chemical industry as well as for regions that are endangered in terms of natural disasters like hurricanes or floods the evacuation areas can be determined preventively. The evacuation area can be determined by using postal codes or easy identifiable natural boundaries (Murray-Tuite and Wolshon (2013)). In most cases the evacuation area is determined by expert judgement. Wilmot and Meduri (2005) formalise this process for hurricane evacuations. Therefore, they use hurricane attributes like track, speed and size to identify the evacuation area. Hsu and Peeta (2014) present a concept to determine evacuation zones for staged evacuation planning. They seek risk based evacuation sub-zones and consider characteristics of the disaster, traffic patterns and network supply conditions. A further complex part of evacuation planning is the evacuation demand modelling which depends on a wide variety of factors. Next to the number of people that has to be evacuated the composition of the evacuee group has to be considered, e.g. which percentage has an own vehicle, are there disabled or elderly people or children in the area that cannot evacuate themselves. Murray-Tuite and Wolshon (2013) identify a lot of factors that influence the evacuate / stay decision which influences the evacuation demand. These factors include environmental factors (distance to threat or present injuries) personal experience of the evacuees (previous experience with hazards or evacuations) and socio-demographic factors (gender, age, income, and children in a household). Dash and Gladwin (2007) present a comprehensive literature review in context of evacuation decision making. They identified a lot of factors that influence the decision of evacuees to evacuate or to stay and investigate the influence of this decision on the evacuation. They summarise that a good evacuation demand forecast is important to estimate the consequences of hazard that depend on evacuation rates, network clearance time or shelter usage. In addition to the total evacuation demand, the time-dependent evacuation rate must be considered for a successful evacuation. Besides the demand resulting from the decision to evacuate or to stay, this demand rate represents when, how many evacuees start the evacuation. To estimate the traffic demand it is necessary to know in which time span the evacuees will use the street network capacities. The travellers departure choice is often modelled with response curves which predict the percentage of departures in each time interval. In literature various

departure time models are used: some assume instantaneous departure times (Chen and Zhan (2008)), uniform distributed departure times (Yuan et al. (2006)) or Poisson distributed times (Cova and Johnson (2002)). With Poisson distributed departure time Cova and Johnson (2002) want to capture that the evacuation demand is low at the onset of evacuation, the evacuation rate then increases to a peak and then gradually tapers off. To depict this behaviour they divide the planning horizon into discrete time periods and determine for each interval the percentage of vehicles that are willing to evacuate. According to Pel et al. (2012) the most realistic way to depict the evacuation departure behaviour are the Weibull distribution and the sigmoid curve. The models described above belong to the sequential travel demand models, where the total evacuation demand is determined first and then the time-depended demand distribution is calculated. Another way is the simultaneous demand modelling, where the processes are executed simultaneously. The travel demand can be calculated in each time span by repeatedly applying a binary logit model, where the evacuees decide in each period to evacuate or to stay (Pel et al. (2012)).

In the third stage, the evacuation decision is done on the basis of the determined risk area and estimated evacuation demand. Evacuation traffic simulation models can be used to estimate the predictability of the evacuation under the given parameters.

The fourth stage deals with public warning. Public warning has a high influence on the number of evacuees which are willing to evacuate and this in turn affects the evacuation demand estimated in the second stage. A lot of studies in social since deal with the right way to communicate the warning massage to the public. Depending on the message more or less people decide to evacuate or stay. Hence, an effective and clear communication of the evacuation has a high influence on the success of an evacuation. Besides the number of evacuees, the warning massage also influences the time-depended evacuation demand, because it affects the decision of the evacuees when to leave the endangered area (Lindell and Perry (2012)). Effective warning is a complex system which is influenced by many factors. Different warning channels will reach different parts of the public and need more or less time to broadcast the message. Effective communication channels are sirens, tone alert, telephone and social media. A combination of indoor and outdoor communication is the most effective way to reach the majority of the people (Sorensen (2000)). A lot of studies focus on the right messages that should be communicated to public. Drabek (1999) stated that in an evacuation message the following questions should be answered: "Who is issuing the warning?, What is threatening?, What exact geographical area is threatened? When is coming? How probable is the event? Are there high risk locations, such as auto mobiles, that require special actions? What specific protection actions should be taken?". A message that answers all these questions is not easy to communicate, particularly because the people that receive the message understand and interpret the message in different ways (Dash and Gladwin (2007)). In case of evacuation it is not easy to communicate with the public and in most cases it is hard to persuade the public to evacuate. In most cases it is impossible to communicate information like route choice, shelter selection etc. to all evacuees.

At the fifth stage the movement of evacuees is considered. This stage deals with questions like: Which ways should the evacuees take to leave the endangered area? How can the street network be adjusted to deal with the evacuation traffic? Or which transportation means are necessary to evacuate the affected area? This is the main topic of this thesis, therefore a comprehensive review of the related literature is presented in Section 2.2.

The sixth stage deals with the emergency sheltering of evacuees. This stage includes decisions made in the preparedness and the response phase of the disaster management life cycle. Two types of shelters have to be differentiated: permanent shelters which are facilities that are not

specialised for evacuation, like churches, stadiums or schools but can be used as shelters in case of emergency evacuation and temporary shelters which can be installed in case of evacuation (Li et al. (2011)). The preparedness phase includes all activities related to the permanent shelters and during the response phase decisions are made about the opening of shelters and the allocation of evacuees to the shelters (Altay and Green (2006), Li et al. (2011)). According to the described activities the literature can be divided into two main field: an area of research that determines optimal locations of shelters and the other field of research deals with traffic routing to or assignment of the evacuees to predetermined shelters. Most research combines both aspects. Sherali et al. (1991) present a location-allocation model to determine the shelters that minimise the congestion-related evacuation time. They point out that the position of shelters can greatly influence the network clearance time. In their model they select the shelters from a pool of potential locations. Kulshrestha et al. (2011) present a robust approach for determining optimal locations for public shelters and their capacities from a given set of possible shelters. They consider demand uncertainties that are associated with the number of people that use a public shelter in case of evacuation. Kongsomsaksakul et al. (2005) also present a location-allocation model. They use a Stackelberg game to determine the shelters that minimise the total evacuation time. In the game, the leader (authority) determines the shelter locations and the follower (evacuees) chooses a destination and route to evacuate. They formulate a bi-level program and solve it with a genetic algorithm. Li et al. (2011) present a two stage stochastic program that considers the aspects of preparedness and response phase and also includes the evacuation traffic. In the first stage decisions about location, capacities, and held resources are made. The second stage allocates evacuees to the shelters and transports resources to the shelters. One of the first models that deals with shelters in emergency evacuations is presented by Yamada (1996). He presents a network flow approach to route the evacuees to predetermined shelters. In a first model the capacities of shelters are not considered. In a second model, a minimal cost flow formulation, the limited capacities of shelters are taken into account.
The seventh stage can be assigned to the recovery phase. Here, the evacuation activities are evaluated and conclusions for later evacuations are made. Moreover, additional activities are identified that need to be installed in the future in order to improve the disaster management. This step then leads to the preparedness phase, clearly highlighting the cyclic character of the disaster management.

## 2.2 Models for Evacuation Traffic Management

As discussed in Section 2.1 one part of the evacuation management deals with the movement of evacuees out of the affected area. In the this section a comprehensive review about traffic models in evacuation management is given. The literature in traffic management for evacuation planning can be divided into simulation and optimisation models. This thesis focuses on the optimisation models for evacuation traffic management, but for the sake of completeness also selected topics of simulation models are presented. One of these selected topics is the analysis of evacuee's behaviour and here the selfish routing model is of special interest. Those models provide some interesting insights in the topic of selfish routing evacuees considered in this thesis. The literature review of this thesis first discusses the relevant literature of simulation models and second the optimisation models for evacuation traffic management are discussed. The literature review on optimisation models is divided into literature that covers the optimisation of the street network and literature that deals with evacuee routing. Studies which use the the Cell

Transmission Model (CTM) for traffic modelling are presented in a comprehensive literature review in Section 3.2 after the general idea of the CTM is explained.
Early models in traffic simulation are introduced by Sheffi et al. (1982) and Pidd et al. (1996). Sheffi et al. (1982) develop the model NETVACI to simulate traffic patterns during emergency evacuations. They estimate the network clearance time for areas surrounding nuclear power plants. The presented model is a macro traffic simulation model that considers the network typology and intersections. The model does not consider individual vehicles but uses mathematical relationships between flow, speed, densities etc. With the model different evacuation scenarios can be simulated and the related network clearance time can be estimated. In contrast to NETVACI the CEMPS model by Pidd et al. (1996) is a microscopic traffic simulation model. In this model a geographic information system (GIS) is combined with a micro-simulator. The roads are constructed by lists of locations and vehicles can flow from one location to another if there is space available in this location. The model is developed to determine suitable evacuation plans. Sinuany-Stern and Stern (1993) present a simulation model to monitor household's actions in evacuations. They consider vehicle based evacuation as well as pedestrian evacuation. They want to examine the sensitivity of network clearance time to several traffic factors and route choice mechanisms. To make the simulation more realistic they implement the interaction between vehicles and pedestrians. Cova and Johnson (2002) present an off-the-shelf microscopic traffic simulator combined with a custom evacuation-scenario generator to simulate evacuation plans for neighbourhoods in fire-prone wild-lands. With the custom scenario generator household trips, departure timing, and destination choice are determined. With the simulation model, evacuation plans for different scenarios for wild-land evacuations are tested. Murray-Tuite and Mahmassani (2004) present a simulation model formulation that integrates household interactions in the evacuation planning. In case of an evacuation the members of a household prefer to evacuate together in a group. This behaviour leads to additional traffic, e.g. parents pick up their children at school or they meet at home instead of directly leaving the affected area. This additional traffic is considered in the model by Murray-Tuite and Mahmassani (2004) and the resulting evacuation time is simulated. Zou et al. (2005) introduce a simulation tool for hurricane evacuation in Ocean City, Maryland. They use a microscopic simulation module and real-time options. Moreover, they integrated an optimisation module that determines a potentially most effective plan under the detected traffic conditions. Therefore this tool takes control strategies like converting a one through lane to a right-turn lane into account. Tu et al. (2010) develop a simulation model for the city of Almere. They consider the driving behaviour of the evacuees and analyse the impact of driving behaviour on the evacuation clearance time. They develop different scenarios in terms of acceleration rate, maximum speed, mean headway, and minimum gap distance. They summarise that an increase in acceleration rate and maximum speed do not have a significant influence on the evacuation time. But a reduction of mean headway and minimum gap distance between two vehicles can reduce the evacuation time. Hence, it is important to consider the driving behaviour by simulating realistic evacuation times. A large part of research in traffic simulation that takes the behaviour of evacuees into account uses agent-based or multi-agent simulation. An agent-based simulation model by Chen and Zhan (2008) analyses the effectiveness of simultaneous and staged evacuation. The traffic flow is modelled at the level of individual acting vehicles. They compare the concept simultaneous evacuee warning with a staged concept where the affected area is divided into zones and the residents in the different zones are informed at different time-point. They test their concept on three network types: a grid road network, a ring road network, and a real network from the City of San Marcos. They conclude that no strategy can be considered as the best across all structures. In addition to

the network structure the performance of their concepts depends on the population density. A further model with agent-based modelling is presented by Chen et al. (2006). They develop a micro-simulation model for the special structure of hurricane evacuations in the Florida Keys. With the agent-based approach they want to capture individual and collective behaviour to create realistic scenarios which consider evacuee's route choice and other driving decisions. By the help of simulation they want to evaluate evacuation plans and help for emergency managers to make decisions during the evacuation organisation. Handford and Rogers (2012) present an agent-based simulation model that considers the specific driving behaviour of drivers in evacuation scenarios. Their model is based on the social force model of crowds and includes evacuation specific desires of drivers like following the routes taken by other drivers. Similar behaviour is considered by Pan et al. (2007) in their simulation studies. They analyse crowd behaviours and observe behaviour patterns like competitive, queuing or herding behaviour. They build the simulation model for pedestrians, but as stated in Handford and Rogers (2012) some of these behaviour patterns can also be observed for drivers in evacuation context. For a comprehensive overview of simulation models in evacuation management the reader is referred to Pel et al. (2012).

Next to the simulation of evacuations a large part of research is engaged in optimising the evacuation process. First, the relevant literature that considers the optimisation of evacuation routes and second the research that focuses on the optimisation of the given street network for the evacuation are presented. Cova and Johnson (2003) develop a network flow model for evacuation traffic. The developed model identifies optimal lane-based evacuation routing plans to prevent traffic delays at intersections. The model is based on a minimum-cost flow problem and in the generated traffic routing plans a reduction in the total travel distance is in conflict with crossing elimination at intersections. Miller-Hooks and Sorrel (2008) introduce a maximal dynamic expected flow problem to determine paths and flows (flow pattern) that maximise the successfully evacuated residents from an affected area in a predetermined time horizon. To capture uncertainty in the number of evacuees that can pass through a passageway, the linked travel times and capacities are assumed to be time-varying discrete random variables with known distribution functions. The model is solved using a metaheuristic based on principles of noisy greed algorithms. Stepanov and Smith (2009) design optimal egress route assignments for given transportation networks. As a performance measure they use clearance time, total travelled distance and blocking probability. The challenge is that the vehicle speed decreases with an increasing number of vehicles that use a road segment and can lead to blocking of highly utilised road segments. For the evaluation of the road dependent travel times a M/G/c/c stage dependent queuing model is used. Saadatseresht et al. (2009) present an optimisation algorithm for the assignment of evacuees to shelters under consideration of optimal routes between the building blocks and the safe places. In the algorithm first (potential) safe areas are designed /selected, then optimal paths between each building block and each safe area are computed and the building blocks are assigned to the safe areas. The problem is defined as a multi-objective problem and different methods for multi-objective optimisation are presented. The approach is tested for an area of Tehran (Iran). Polimeni and Vitetta (2011a) present a shortest path and route design problem in time-dependent networks. The path design formulation is based on dynamic programming and a solution is found using a modification of the Dijkstra algorithm. In the model paths are designed for emergency vehicles operating in the network to support the population involved in the disaster. In the time-dependent network private and emergency vehicles are considered. The goal of the model is to minimise the route costs for the vehicles. To solve the problem described by Polimeni and Vitetta (2011a), Polimeni and Vitetta (2011b) present a further algorithm which is based on the time-generalised Bellman optimality condition and test the algorithm on a real network. A model for traffic flow optimisation is introduced by Bretschneider and Kimms (2011). With their model the traffic is reorganised to minimise the evacuation time by prohibiting conflicts within intersections. The problem is modelled as a dynamic network flow problem with additional variables for the number and direction of used lanes. To handle problems of realistic size a relaxation based heuristic is presented. Bretschneider and Kimms (2012) present a pattern-based approach for evacuating urban areas to solve the problem that is described in Bretschneider and Kimms (2011). The objectives of the problem are the clearance of the affected area as safe as possible and as early as possible. They develop a two stage heuristic to solve the pattern-based evacuation problem. Lim et al. (2012) introduce a capacity constraint network flow optimisation approach for finding evacuation paths that maximise the number of evacuees in short notice evacuation planning. To consider the dynamic character of evacuations they expand the static network flow model to a time-dependent network flow model. For solving the problem they develop an evacuation scheduling algorithm, where first evacuation paths with Dijkstra's algorithm are computed and then the evacuees are assigned to these paths by means of a greedy algorithm. Coutinho-Rodrigues et al. (2012) formulate a multi-objective model to determine evacuation paths and shelters. They consider six objectives like risks associated to paths and shelter location, path length or evacuation time from shelter to other places e.g. hospitals. They test the model for the city of Coimbra (Portugal) and use different methods of multi-objective optimisation like the weighting method or by minimising the distance to the "ideal solution". This solution is obtained by solving the problem for each objective separately. Hamacher et al. (2013) introduce a flow location model, which combines dynamic network flow optimisation and location analyses. With the model they determine the position of facilities in networks, so that the maximum flow will not decrease. They present a single (1-FlowLoc) and a multi facility flow location model (q-FlowLoc). They develop three exact solution approaches for the 1-FlowLoc and a heuristic for the q-FlowLoc problem. Goerigk et al. (2014) develop a multi-criteria optimisation model that combines different aspects of evacuation planning which are normally solved separately. They consider location aspects like choice of shelter location, routing aspects for bus and individual traffic and risk aspects for the chosen routes. The objectives of the problem are the minimisation of the evacuation time, the risk and the number of opened shelters. The problem is modelled as a combination of a dynamic network flow model (private traffic) as well as a multicommodity network flow model (public transport). To solve the mathematical model for realistic problem sizes they introduce a genetic algorithm. Lim et al. (2014) present a dynamic network-based evacuation model that considers uncertain capacities of road links. With the model they plan evacuation routes and investigate the relationship between the clearance time, the number of evacuation paths and congestion probability during evacuations. For a predefined time horizon the presented model selects evacuation paths and flows that results in minimal congestions.

The literature discussed above is based on the strategy of defining optimal routes for the evacuees. Another way to regulate the evacuation traffic is the adjustment of the network. A common method to increase the network capacity in case of emergency evacuations is the so called "contra-flow" introduced by Wolshon (2001). In this concept some or all inbound lands of a motorway are used as outbound lanes by an evacuation. Sangho et al. (2008) adopt this concept of contra-flow and present a model to define the optimal driving direction for each edge in the network. They model this problem on a macroscopic flow model without considering social behaviour of evacuees, operational costs of contra-flow, or traffic signal settings. They focus in their paper heuristics to solve the contra-flow problem for large transportation networks. The objective they consider is the minimisation of evacuation time. They present a greedy and a bottleneck relief heuristic to solve the model formulation. Xie and Turnquist (2011) formulate a lane-based evacuation network optimisation problem. They combine contra-flow and crossing elimination strategies. The model is bi-level where the upper-level determines which lanes are used for contra-flow and where crossing elimination is considered. The lower-level assigns the traffic to routes. To solve the complex problem they use an integrated lagrangian relaxation approach and a tabu search method. Madireddy et al. (2011) present a strategy called "throttling" to cope with selfish acting evacuees. To guide the traffic, road segments are temporarily closed when a specific congestion level is reached and opened when the traffic falls down to a predefined threshold. With this concept they can react on real-time traffic flow and they argue that it is easy to implement in a real life situation. Hence, traffic demand forecasting is not needed anymore. Huibregtse et al. (2012) point out that evacuees will not behave as the flows proposed in an optimal evacuation plan. Therefore, they developed a concept that forces the evacuees to act as the optimal evacuation flow by closing road segments. In their method they identify streets which are not used in an optimal evacuation plan, and close these roads for evacuation traffic. In a case study they show, that this strategy leads to a decrease in evacuation time of up to 13.4 %. Hadas and Laor (2013) present a multi-objective model to design a network that is ideally suited for the daily traffic and also for the evacuation traffic. The presented model optimises both the evacuation time and the network construction costs. To cope with real-sized networks they use a heuristic based on the minimum-cost problem.

## 2.3 Assumptions of the Evacuation Scenario

In Section 2.1 the stages of evacuation planning were assigned to phases of the disaster management life cycle. The described stages of evacuation planning point out that a multitude of characteristics is necessary to define an evacuation scenario. In this section these characteristics are used to define the evacuation scenario studied in this thesis. For this purpose, it is stated which assumptions are made and why these assumptions are reasonable. The evacuation scenario is defined as follows:

- The evacuation of urban areas is considered.
- The behaviour of the evacuees is integrated in the evacuation plan, as the subject of research. Therefore, an evacuation scenario, which considers private vehicles, is planned. For the sake of simplicity the interaction with the other road users is not taken into account.
- To keep it simple in the following one vehicle equals one evacuee, regardless of how many people are in the vehicle. For planning the traffic flow it is important to take the number of vehicles in the street into account.
- It is assumed that the area that is to be evacuated is already defined according to one of the concepts presented in the literature. The evacuation area is composed of an affected area, which the evacuees have to leave and predefined safe places, which the evacuees have to reach. When the evacuees have reached a safe place, it is assumed that the evacuation is finished for them.
- The evacuation scenario described in this thesis does not depend on a specific hazard. It is assumed that the level of hazard is the same in the whole affected area.

- The network and the traffic flow inside the affected area are modelled with the cell transmission model (a detailed description is given in Chapter 3).
- The position of each vehicle in the network is known.
- In the scenario no preparation time for evacuees is considered. All evacuees are ready to start the evacuation at the same time, namely at the very beginning of the planning horizon. In the literature there are different ideas about time dependent evacuation demand modelling. If necessary, these concepts, can be easily implemented in the defined evacuation scenario (see Chapter 5).
- The main objective in the considered scenarios is the clearance of the affected area as fast as possible. As a measure the network clearance time (NCT) is used, which is defined as the first period where all evacuees have left the affected area.
- In the introduced evacuation scenario independent acting evacuees are considered. It is the main contribution of this thesis that the behaviour of selfish acting agents is included in the evacuation planning process (the evacuees are defined in detail in Chapter 5).

# Chapter 3

# The Cell Transmission Model

The evacuation traffic in this thesis is modelled using the cell transmission model (CTM). The general concept of the cell transmission model is presented in Section 3.1. Here the characteristics of a cell are explained and it is demonstrated why it is a useful way to model traffic flow. For this purpose the fundamental diagram of traffic flow is explained and the relation between this diagram and the cell transmission model is highlighted. Section 3.2 explains how this general concept of modelling / simulating traffic can be transferred to an optimisation model for evacuation traffic management. Moreover, literature that uses the CTM in the context of evacuation traffic management is presented.

## 3.1 Basics CTM

This thesis utilises the idea of the CTM as invented by Daganzo (1994) to model the traffic flow of evacuation settings. The CTM is a macroscopic model for a traffic flow and is based on the hydrodynamic theory of traffic flow described by Lighthill and Whitham (1955) and Richards (1956). This theory takes the relation between traffic flow and traffic density into account. Daganzo (1994) introduced a simulation model for traffic flow on a simplified highway with one entrance and one exit. This approach is extended by Daganzo (1995) to cope with more complex network structures like intersections. Kimms and Maassen (2011b) extend the formulation by Daganzo (1994) to optimise evacuation traffic. This thesis makes use of the latter approach to model the traffic flow within evacuation scenarios.
The traffic flow is modelled in the CTM with a time-scan strategy, thus the current traffic conditions are updated with a tick of a clock (Daganzo (1994)). The considered time horizon is divided into periods $t = 1..|T|$ and in each period the traffic conditions are updated. Furthermore, the street network is divided into sections $i = 1..|I|$ where $I$ is the set of all sections. These network sections are called cells, which is why the model is called cell transmission model. These sections are applied to section and flow capacities in order to consider realistic traffic phenomena (building of queues or shock waves). These conditions are illustrated with Figure 3.1, which depicts a simplified street network divided into sections. The length of a section is defined as the distance a vehicle can travel by light traffic in one period. For the first street (sections 1 to 5) in Figure 3.1 a maximum free flow speed of 50 km/h ($\approx$ 13.9 m/s) and for the second street (sections 6 and 7) a maximum free flow speed of 30 km/h ($\approx$ 8.3 m/s) is assumed. In the example the length of a period is 9 seconds. Therefore, a section in street one has a

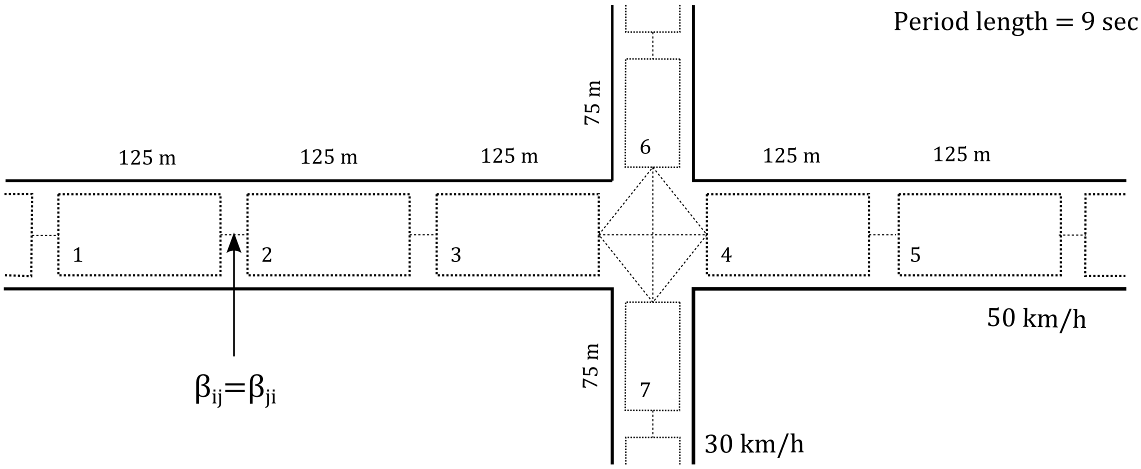



Figure 3.1: A Street Network Represented as Sections.

length of 13.9 m/s × 9 s ≈ 125 m and in the second street a length of appr. 75 m (8.3 m/s × 9 s) results. Thus a vehicle can drive 125 m or 75 m in one period by light traffic. The traffic flow is modelled by updating the conditions of the section in each period. The number of vehicles in section $i$ in period $t$ is $x_{it}$ and for the simplest situation by light traffic on a street without intersections (e.g. flow from section 2 to 3) all vehicles that are in period $t$ in section $i$ were in period $t-1$ in section $j$

$$x_{i,t} = x_{j,t-1}. \tag{3.1}$$

The number of vehicles, which is present in the sections, additionally depends on the traffic flow variable $y_{ijt}$. In Daganzo (1994) only a simple highway with one entrance and one exit is considered: in this model formulation the sections are numbered consecutively. This means the traffic can only flow from section $i$ to section $i+1$. To model more complex networks Kimms and Maassen (2011b) introduce the binary parameter $\beta_{ij}$ which indicates connections between sections. If two sections are connected it will be $\beta_{ij} = \beta_{ji} = 1$ and 0 otherwise. In a network (on an intersection) the traffic from a section $i$ can flow into different sections, similarly the traffic from different sections can flow into section $i$. For the determination of the vehicles, which are in section $i$ in period $t$, the traffic flow between connected sections must be considered. Thus condition 3.1 must be reformulated for each $i \in I$ to

$$x_{it} = x_{it-1} + \sum_{j \in I} y_{jit-1}\beta_{ij} - \sum_{j \in I} y_{ijt}\beta_{ij}. \tag{3.2}$$

The vehicles that are in section $i$ in period $t$ are the vehicles that were in section $i$ in the previous period $t-1$, plus the vehicles that have entered section $i$ from all with $i$ connected sections $j$ ($\beta_{ij} = 1$) in period $t-1$, less the vehicles that leave section $i$ to sections connected with $i$ in period $t$. An incorporation of capacities is necessary to model the traffic flow more realistically and to consider traffic phenomena such as congestions. Let $N_{it}$ be the maximum number of vehicles that can be present in section $i$ in period $t$ and let $Q_{it}$ be the maximum number of vehicles that can flow from one section to the next section in one period. To compute the maximum capacity of a section the length of a section must be divided by the average length of a vehicle and a minimum distance between two vehicles. With an assumed average length of 4.5 m per vehicle and a minimal distance between two vehicles of 1 m, the capacity of a

section is $N_{it} = \frac{125\,m}{5.5\,m} \approx 22$ vehicles for the first street as depicted in the example in Figure 3.1. For the flow capacity, which is $Q_{it}$, a larger safety distance between two vehicles is assumed. This depends on the speed limit in a section. For the first street a safety distance of 13.9 m is assumed and the flow capacity is $Q_{it} = \frac{125\,m}{4.5\,m+13.9\,m} \approx 6$ vehicles. A street with more than one lane is modelled by multiplying the capacity by the number of lanes.
The possible traffic flow between section $i$ and section $j$ in period $t$, is additionally restricted by the available capacity in section $j$: $N_{jt} - x_{jt}$. The values that restrict the traffic flow are derived from the fundamental diagram of traffic flow (Lighthill and Whitham (1955)). The relation between the capacities, which are defined above, and the fundamental diagram of traffic flow, which depicts the correlation between traffic density and traffic flow, will be shortly explained by means of the simplified illustration in Figure 3.2 (for a more detailed explanation the reader is referred to Daganzo (1994)).

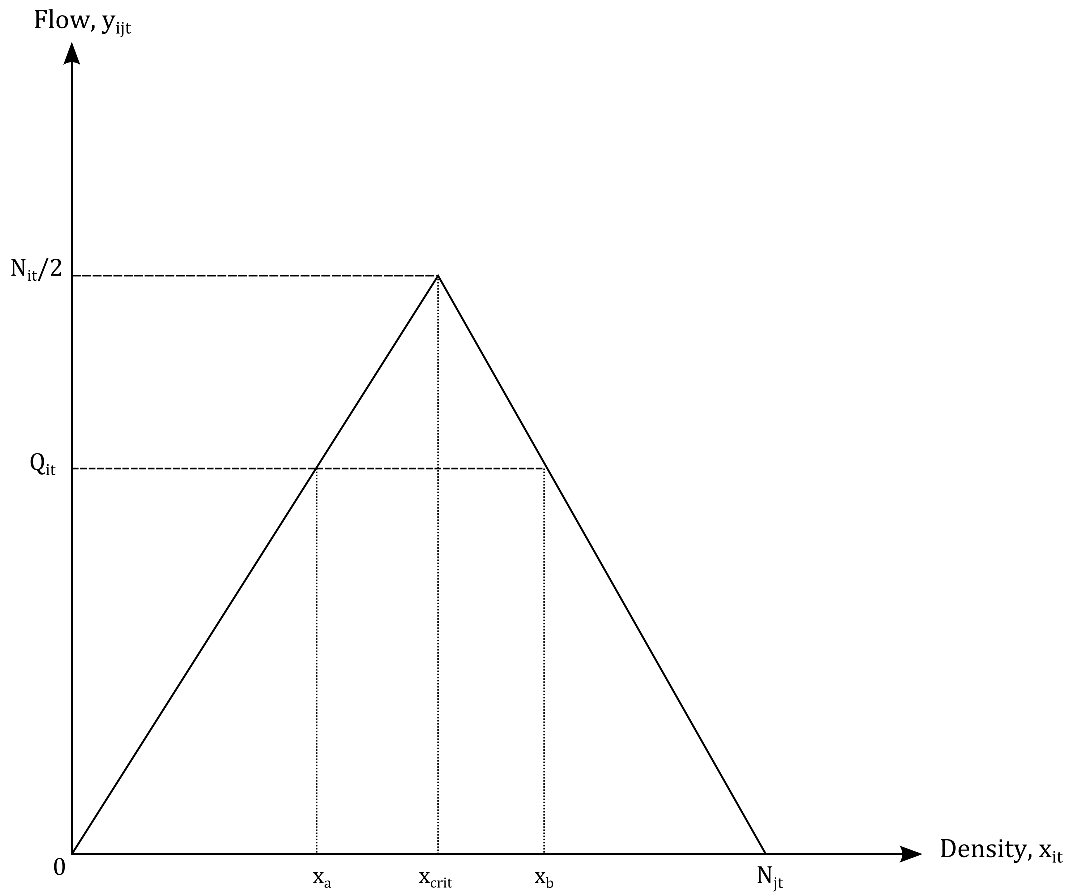



Figure 3.2: Fundamental Diagram of Traffic Flow.

The traffic flow $y_{ijt}$ is the amount of traffic that flows in a street segment (section) in a given time span. The traffic density $x_{it}$ is the number of vehicles in this street segment, $N_{jt}$ is the maximum number of vehicles that fits into the street segment hence correlating to the traffic density in a congested street. The traffic flow on a street is the product of traffic density and the average speed $v$ in this street segment. With a constant average speed $v$, the traffic flow increases with rising up traffic density. This happens as long as there is enough capacity in a street segment. In Figure 3.2 it is the density in the range between 0 and $x_{crit}$: the traffic flow increases with increasing traffic density. If there is not enough capacity in a street section, the traffic flow will decrease with increasing traffic density. Starting from density $x_{crit}$ the traffic flow decreases as the vehicles in a street segment cannot keep the average speed and the number of vehicles that pass through a street segment in a given time span decreases. If the traffic density $x_{jt}$ is equal to $N_{jt}$ no vehicle can move. In this example the free flow speed is equal to the backward moving speed, therefore the point where the traffic flow decreases with increasing density $x_{crit}$ is $N_{jt}/2$. With the traffic flow $Q_{it}$, further conditions of the street segment can be considered, which are more restrictive than $N_{jt}/2$, e.g. a required minimal distance between two vehicles. In the CTM these relations are considered by means of the boundaries of the traffic flow

$$y_{ijt} = min\{x_{it}, Q_{it}, N_{jt} - x_{jt}\}. \tag{3.3}$$

The traffic flow from section $i$ to section $j$ cannot exceed the number of available vehicles in section $i$ $x_{it}$. In Figure 3.2 it is the traffic flow that results from the traffic density in the range between 0 and $x_a$. The traffic flow is restricted by the number of the vehicles that are present in section $i$. More than these vehicles cannot flow from $i$ to $j$. If there are more vehicles in section $i$ than $Q_i$, the flow into section $j$ will be restricted by the flow conservation. More than $Q_i$ vehicles cannot flow out of section $i$. The diagram reflects this traffic density in the range between $x_a$ and $x_b$. The last restriction is the available capacity in section $j$, which is $N_{jt} - x_{jt}$. The number of vehicles that flow into section $j$ cannot exceed the available capacity in this section. It is the traffic density in the range between $x_b$ and $N_j$ in the fundamental diagram of traffic flow. In Daganzo (1994) the traffic flow is simulated by applying all these restrictions and by updating the traffic conditions in each period. By considering these capacities the traffic flow can be modelled realistically. In the next section the basic idea of the traffic flow simulation by Daganzo (1994) is adopted to an optimisation model for traffic flow optimisation in evacuations.

## 3.2 CTM and Evacuation Planning

A lot of studies in the area of evacuation traffic optimisation model the traffic flow with the CTM. This section first summarises the related literature and then a model for optimising evacuation traffic by Kimms and Maassen (2011b) is presented.

*Literature review*

One of the first optimisation models that formulates the traffic flow on the basis of the CTM is introduced by Ziliaskopoulos (2000). It is a simple traffic assignment model without evacuation aspects. However this formulation has been widely adopted by the studies that use the CTM in optimisation models. Kalafatas and Peeta (2006) and Kalafatas and Peeta (2009) investigate contra-flow and signal control strategies for an optimal capacity use of the network in an evacuation. Kalafatas and Peeta (2006) seek links whose capacity ought to be augmented with the contra-flow mechanism. Moreover, they optimise the traffic signal pattern for the evacuation traffic. In their formulation a budget restriction for the network optimisation is considered. In Kalafatas and Peeta (2009) the budget restriction is reformulated to a predetermined number of links that could be used for capacity augmentation with contra-flow mechanism. This reformulation was necessary, because it is too difficult to obtain the required data on staff and budget in a real life situation. Tuydes and Ziliaskopoulos (2006) also propose an optimisation model with contra-flow. To handle real-sized models they present a tabu search-based heuristic. Xie et al. (2010) also use a contra-flow strategy to augment the capacity in a street network. In addition they optimise the network to eliminate crossings. They use a bi-level formulation, where the upper-level optimises the network and the lower-level, which is based on the CTM, assigns the traffic in the network. To solve the model they use lagrangian relaxation and tabu search. Liu et al. (2006) present a staged evacuation planning concept. They divide the affected area in different zones and compute optimal starting times and routes. Bish et al. (2014) also introduce a concept with staging and routing. Additionally, they consider different evacuee types with various destination requirements and shelter capacity restrictions. Chiu and Zheng (2007) define an evacuation plan for multi-priority groups. Different emergency response and evacuation flow groups must be routed in the same network with various destinations and priorities. They define traffic assignment strategies and departure schedules for the complex situation with different groups. An evacuation plan with dynamic resource allocation and movable devices is determined by He and Peeta (2014). In the plan the evacuation resources, for instance message sign systems, can be reallocated dynamically as reaction to the traffic conditions. Ng and Waller (2010) propose a model with demand and capacity uncertainty. They optimise evacuation routes and use a distribution-free approach to provide probabilistic guarantees on the resulting evacuation plan. Yazici and Ozbay (2010) also consider uncertainties in evacuation demand and road capacity. They present two formulations of a stochastic optimisation model to compute a system-optimal traffic assignment. Moreover, they consider individual chance constraints and joint chance constraints. Most of the model formulations, which are based on the CTM, include traffic holding. Shen et al. (2007) introduce a model formulation that prevents traffic holding. They argue that an optimal traffic pattern without traffic holding is more cost efficient and easier to implement in emergency response. Tuydes-Yaman and Ziliaskopoulos (2014) propose a model that focuses on optimal demand management strategies of evacuation traffic. They optimise the evacuation departure times, destination choice and zone scheduling. Kimms and Maassen (2011b) determine optimal routes to guide the evacuees out of the endangered zone with maximum security. Kimms and Maassen (2011a) extend this formulation and use multiple cell sizes to handle larger networks and compute plans for realistic scenarios. Kimms and Maassen (2012b) propose a heuristic approach for the fast computation of evacuation plans for large networks. Kimms and Maassen (2012a) integrate rescue teams in the basic model by Kimms and Maassen (2011b). They differentiate between two situations: in the first case the rescue teams just have to arrive at the origin of danger and in the second case the teams have to commute between a place in the affected area and a safe place. Street sections are reserved for the rescue teams: in the first situation the street can be opened for the normal traffic after the rescue teams have reached the predetermined point; for the second situation this option does not exist. This problem is solved by a three stage heuristic, which is based on the approach introduced by Kimms and Maassen (2012b). Kimms and Maiwald (2015a) introduce a model that takes uncertainties in the street network into account. Furthermore, they introduce the aspect of resilience in context of evacuation planning. The resilience of a street network should be improved by a balanced utilisation of the network capacities. To solve the model they develop a path generation algorithm. In Kimms and Maiwald (2015b) a model formulation is presented that combines the advantages of the CTM with the advantages of a network flow formulation. With the network flow approach instances of realistic size can be considered whereas with the CTM traffic phenomena like shock waves and congestions can be represented. The basic model by Kimms and Maassen (2011b) is at first formulated more restrictively and then transferred into a network flow model. In a computational study they show that a significant reduction in computation time can be achieved, using this new formulation.

*Assumptions and model formulation*

The models presented in this thesis are based on Kimms and Maassen (2011b). For the sake of completeness the assumptions of this study and the mathematical model are explained shortly:

- The network is divided into an affected area, which the evacuees have to leave and a safe zone, which the evacuees have to reach. The safe zone is modelled as a super sink $S$. The level of danger in the affected area is defined using weights $c_{it}$, for $i \in I$, where $c_{St} = 0$.
- It is the objective of the model, to route the evacuees or vehicles (the terms evacuee and vehicle are used synonymously in this thesis) out of the affected area under maximum security.
- Only the traffic routing in the affected area is optimised, the routing outside this area is out of scope of this study.
- The evacuees start the evacuation from an area outside the street, e.g. a parking lot or a driveway. The number of evacuees that start evacuation in section $i$ in period $t$ is represented by variable $b_{it}$.
- It is assumed, that all evacuees follow the instructions of the authorities, therefore the computed routes are accepted by all evacuees. With the optimisation model a system-optimal solution is computed.
- In the model the optimal departure times and routes for the evacuees are determined. Also the driving directions for the streets are optimised.
- All evacuees must be evacuated in the planning horizon $T$. To achieve a feasible solution, $T$ must be set high enough.
- The number of evacuees starting from each section, which is $E_i$, is known.
- The speed limits for all sections are given and it is assumed that the evacuees keep these limits.

According to the assumptions stated above, the following model to optimise the traffic flow out of the affected area is formulated. The objective of the model is to guide all evacuees out of the affected area with maximum security. Let variable $z_{it}$ be the evacuees that are in each section in the affected area.

$$min \sum_{i \in I} \sum_{t \in T} c_{it} z_{it}. \tag{3.4}$$

The number of evacuees in a section of the affected area is weighted with the danger in a section $c_{it}$. This product is minimised by routing the evacuees with maximum safety outside of the affected area. The number of evacuees in the street and in the area around the street is determined with the following conditions

$$z_{it} = x_{it} + \sum_{j \in I} y_{ijt} + E_i - \sum_{\tau=1}^{t} b_{i\tau} \qquad i \in I, t \in T, \tag{3.5}$$

$$x_{it} = b_{it} + x_{i,t-1} + \sum_{j \in I} y_{ji,t-1} - \sum_{j \in I} y_{ijt} \qquad i \in I, t \in T, \tag{3.6}$$

$$z_{it} \leq N_{it} + E_i - \sum_{\tau=1}^{t} b_{i\tau} \qquad i \in I, t \in T. \tag{3.7}$$

Constraints (3.5) determine for each section $i$ and period $t$ the number of evacuees that use a section. These are all evacuees that are in the street ($x_{it}$ and $\sum_{j \in I} y_{ijt}$) and that wait outside to start the evacuation ($E_i - \sum_{\tau=1}^{t} b_{i\tau}$), which is the number of evacuees that starts the evacuation at section $i$ less all evacuees that have entered the network until period $t$. Constraints (3.6) are an extension of condition 3.2. In restriction 3.2 the traffic starts in a start section and traffic flow is only possible between sections. With this extension vehicles can enter the street network at each section. Let variable $b_{it}$ be the number of evacuees that enter the network at section $i$ in period $t$. Constraints (3.7) restrict the number of vehicles that use a section. This number must not exceed the capacity of a section plus the number of vehicles waiting outside to enter a street.

Besides constraints, which compute the number of evacuees in the affected area, the traffic flow between sections is important for the optimisation of the traffic flow. From condition (3.3) the following constraints can be derived

$$\sum_{j \in I} y_{jit} \leq N_{it} - x_{it} \qquad i \in I, t \in T, \tag{3.8}$$

$$\sum_{j \in I} y_{jit} \leq Q_{it} \qquad i \in I, t \in T, \tag{3.9}$$

$$\sum_{j \in I} y_{ijt} \leq Q_{it} \qquad i \in I, t \in T. \tag{3.10}$$

Constraints (3.8) restrict the incoming flows in every section with regard to the section's capacity. Constraints (3.9) and (3.10) are the flow capacity restrictions for the traffic flow between sections. The traffic flow is not allowed to exceed the flow parameter $Q_{it}$. Constraints (3.11) ensure that the traffic flow takes place only between connected ($\beta_{ij} = 1$) sections

$$y_{ijt} \leq N_{it}\beta_{ij} \qquad i, j \in I, t \in T. \tag{3.11}$$

Constraints (3.12) and (3.13) are specific for the evacuation process

$$\sum_{t \in T} b_{it} = E_i \qquad i \in I, \tag{3.12}$$

$$x_{S|T|} = \sum_{i \in I} E_i + \sum_{i \in I} x_{i0}. \tag{3.13}$$

Conditions (3.12) enforce that all evacuees, who wait outside the street, drive into a street section in the planning horizon $T$. Constraint (3.13) ensures that all evacuees have reached the safe zone $S$ at the end of the planning horizon $|T|$. The domains of the decision variables are defined by (3.14) - (3.17)

$$z_{it} \geq 0 \qquad i \in I, t \in T, \tag{3.14}$$

$$x_{it} \geq 0 \qquad i \in I, t \in T, \tag{3.15}$$

$$y_{ijt} \geq 0 \qquad i, j \in I, t \in T, \tag{3.16}$$

$$b_{it} \geq 0 \qquad\qquad i \in I, t \in T. \tag{3.17}$$

With the given model formulation the traffic flow out of the affected area can be optimised. For this purpose optimal routes for all evacuees are determined. The direction of traffic flow is not given in the model. In fact, the used direction is a result of the optimisation, indication that contra-flow can be captured with this mathematical model.

# Chapter 4

# Selfish Routing in Traffic Networks

In this chapter the main aspects of selfish routing in traffic networks are summarised. Selfish routing is based on the idea of independent self-optimising agents that make their route choice without considering the behaviour of the other network users. Later this aspect will be transferred to evacuation traffic management. Hence, in the following sections the most important aspects of selfish routing in (traffic) networks are discussed. In Section 4.1 the consequences of selfish routing on the traffic flow are illustrated by means of an example and the concepts of user-optimal flow, system-optimal flow and the price of anarchy are defined. Furthermore the related literature of selfish routing in traffic networks is summarised. Section 4.2 introduces a traffic phenomenon called "Braess's paradox" that occurs in networks with selfishly routing network users. This phenomenon is first explained in more detail by means of an example and then it is shown why it exists in traffic networks. Moreover, the relevant literature on the Braess paradox is summarised. The aspects of selfish routing in traffic networks, which are most important for this thesis, are summarised in Section 4.3.

## 4.1 Basics of Selfish Routing

A well-known problem is the network's traffic management. Many network users act independently in an environment with limited capacity; each of them with his or her own goals. In order to reach the defined goals most of the network users will focus on strategies that do not consider the behaviour of the other network users. The users in a network act selfishly, only interested in the best possible realisation of their own interests. The problem, which arises from this behaviour, is the well-studied problem of selfish routing in networks. The first known economist who considers selfish behaviour was Pigou (1920). He investigated a system with different taxations on resources in industries. To illustrate the effects of selfish behaviour he used an example of traffic flow in a network. Roughgarden (2005) adopts this example to point out the consequences of selfish routing in networks: "Selfish behaviour by independent, non-cooperative agents need not produce a socially desirable outcome".
Afterwards, in this section an example is used to illustrate the findings of Pigou (1920) and the adoption by Roughgarden (2005) and the resulting effects of selfish routing in traffic networks are explained. The traffic network is represented by a graph $G(V, E)$ with two nodes $V = \{A, D\}$ and two edges $E = \{a, b\}$. Node A is the source and node D the sink of the network, the traffic flows from A to D. The fraction of traffic that use route 1 traverses edge

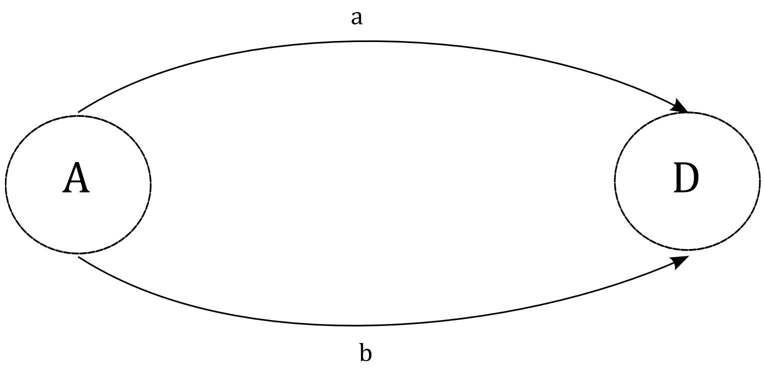



Figure 4.1: Network to Illustrate Selfish Behaviour in Networks.

$a$ and the fraction of traffic on route 2 traverses edge $b$. The total traffic flow, which is in the network ($x = x_1 + x_2 = 1$), consists of the traffic that traverses the route 1 ($x_1$) and the traffic that traverses the route 2 ($x_2$). The travel times are $C_1(x_1)$ on route 1 and $C_2(x_2)$ on route 2. Figure 4.1 depicts the described network that is based on the example of Roughgarden (2005). In the example it is assumed that the traffic on route 1 needs 1 hour to travel from A to D independent of the fraction of traffic that uses this route. Hence, the travel time (in hours) is defined as

$$C_1(x_1) = 1. \tag{4.1}$$

On route 2 the travel time depends on the amount of traffic which traverses this route. The traversing traffic is that fraction of the traffic that does not use the route 1: $x_2 = 1 - x_1$. The travel time (in hours) on route 2 is defined as

$$C_2(x_2) = x_2 \tag{4.2}$$

and corresponds to the fraction of traffic that traverses route 2. In a scenario where the total traffic flow traverses route 2 the travel time will be equal to the travel time on route 1 (1 hour). For demonstrating the effects of a selfish behaviour, this example assumes that the traffic flow consists of selfish agents who prefer the fastest route. Consequently, all network users will take route 2, because the travel time on route 2 is never longer compared to the travel time on route 1. But if a fraction of the network users travel on route 1, the travel time on route 2 will be shorter than on route 1. Therefore, route 2 is advantageous for all network users, that minimise their own travel time. The average travel time is defined as

$$C_{avg}(x_1, x_1) = C_1(x_1) \times x_1 + C_2(x_2) \times x_2. \tag{4.3}$$

This average travel time is used to show the negative impact of selfish behaviour on the travel time. In the scenario with selfish acting agents the traffic flows are $x_2 = 1$ and $x_1 = 0$ on the routes, the resulting average travel time is 1 hour

$$C_{avg}(x_1, x_2) = 1 \times x_1 + x_2 \times x_1 \Leftrightarrow C_{avg}(x_1, x_2) = 1. \tag{4.4}$$

To demonstrate the influence of selfish routing in traffic networks, the average travel time by selfishly routing network users is compared to the minimum average travel time in the network. The average travel time of the network, which is $C_{avg}$, is reformulated with respect to the traffic flow on route 1 as follows

$$C_{avg}(x_1) = x_1 + (1 - x_1) \times (1 - x_1) \Leftrightarrow C_{avg}(x_1) = 1 - x_1 + x_1^2. \tag{4.5}$$

The minimum travel time is reached with the traffic flows $x_1 = 0.5$ and $x_2 = 1 - x_1 = 0.5$. The resulting minimum average travel time (in hours) is: $C^*_{avg}(x_1, x_2) = 1 \times 0.5 + 0.5 \times 0.5 = 0.75$. If one half of the network users takes route B the travel time of no network user increases and the half of the network users can reduce their travel time. The example reveals the findings by Pigou (1920), that selfish behaviour does not necessarily result in a social optimal outcome. Wardrop (1952) later investigates the distribution of traffic on different routes in networks and defines his well-known first and second principles of equilibria (in traffic networks).

*User-optimal Flow*

The first principle defines an equilibrium called the *user-optimal flow*. In this equilibrium the traffic is assigned to the routes in the network in a way that all used routes have an equal travel time and these travel times are less than the travel times of all unused routes. Hence, in this equilibrium no network user can reduce his or her travel time by choosing a different route. The user-optimal flows are illustrated on the example in Figure 4.1. The equilibrium travel time is formally stated with

$$C_1(x_1) = C_2(x_2). \tag{4.6}$$

Using the travel times, which are defined in (4.1) and (4.2), the equilibrium can be calculated as: $1 = x_2$ which is equivalent to $x_1 = 1 - x_2 = 0$. Similar to the solution with selfish network users, the average travel time is 1 hour. There is no incentive for the network users to switch from route 2 to route 1, because the change from route 2 to 1 does not reduce their travel time. Thus the user-optimal flow is equal to the traffic flow resulting from selfish routing. In the literature of transportation science Wardrops user equilibrium (fist principle) is used synonymously with the Nash equilibrium in the classical non-cooperative game theory (Roughgarden (2005)). Hence, in this thesis also both terms are used synonymously.

*System-optimal Flow*

The second principle defines an equilibrium that is called *system-optimal flow*. In this equilibrium the traffic is assigned to the given routes in such a way that the average travel time is minimised. It is calculated according to the minimum average travel time in the example above (fist derivation of (4.5)). The example in Figure 4.1 shows that the equilibrium is reached by traffic flows $x_1 = x_2 = 0.5$ and the resulting average travel time is 45 minutes.
The principles have shown that the distribution of traffic is essential for the travel time in networks. In addition, it was shown that selfish behaviour leads to a suboptimal distribution of traffic in the network, and also worsens the travel time throughout the network.

*Price of Anarchy*

Investigating the influence of selfish routing in networks on the travel time and determining the dimension of increases in travel time compared to the system-optimal flow are important areas of research. To quantify the consequences of selfish behaviour Koutsoupias and Papadimitriou (1999) introduce the *coordination ratio*, and Papadimitriou (2001) later defines the *price of anarchy*. The coordination ratio is the ratio between the user-optimal flow and the system-optimal flow. Papadimitriou (2001) uses the term price of anarchy because it describes the price that results from decisions of uncoordinated and utility-maximising individuals. The price of anarchy for selfish routing games was introduced by Roughgarden and Tardos (2002). Based on their definition the price of anarchy $\rho$ depends on the travel time by system-optimal flows $C^*_{avg}(x^*_1, x^*_2)$ as well as on the travel time by user-optimal flows $C_{avg}(x_1, x_2)$ and is defined as follows

$$\rho = \frac{C_{avg}(x_1, x_2)}{C^*_{avg}(x^*_1, x^*_2)}. \tag{4.7}$$

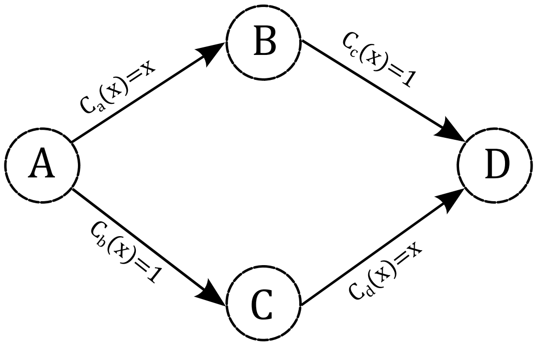
In the example network in Figure 4.1, the average travel times in the user- and system-optimal solution are $C_{avg} = 60$ min and $C^*_{avg} = 45$ min. The price of anarchy can be calculated as: $\rho = \frac{60}{45} = \frac{4}{3}$. The price of anarchy shows the increase of travel time by selfish behaviour compared to the minimum possible travel time in the network. The travel time, which arises by selfish behaviour, equals the minimum possible travel time increased by $\frac{1}{3}$ of this time.

*Literature Review*

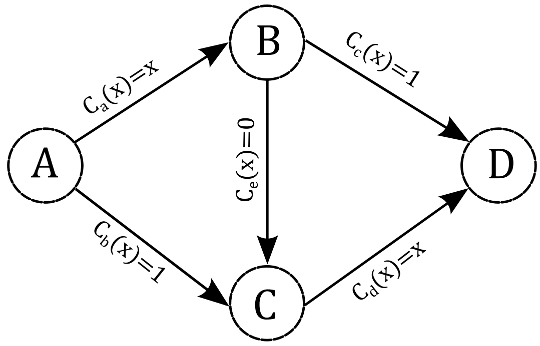
The subsequent paragraph summarises the relevant literature on selfish routing in networks. Besides selfish routing in traffic networks also selfish routing in communication networks is studied. But as this thesis focuses solely on transportation networks, the second application is not considered here. Some relevant papers that deal with selfish routing in communication networks are published by Friedman (2004), Xiao et al. (2008), Hoefer and Souza (2010), and Knight and Harper (2013).
Most studies about selfish routing in traffic networks either deal with the development of concepts to reduce the price of anarchy or investigate network conditions that affect the price of anarchy. Roughgarden (2003) shows that the price of anarchy does not depend on the network topology. In Correa et al. (2004) the price of anarchy for capacitated networks is studied. They show that additionally to the findings by Roughgarden (2003) the price of anarchy does not depend on the network capacity. Cole et al. (2003) investigate how taxes on edges can improve the uncoordinated behaviour of the network users. They determine taxes and minimise the total costs for the network users (general arc costs plus taxes on edges). Roughgarden and Tardos (2002) determine the negative impact of the price of anarchy. For linear latency functions they prove that the travel time in the Nash equilibrium is $\frac{4}{3}$ times the travel time in the system optimum. Karakostas et al. (2007) analyse a system with oblivious network users. They want to investigate a system in which network users do not consider congestion in their route selection. They model network users who select their route only by considering the distance of the route, i.e. the shortest route. Ferrante and Parente (2008) investigate a network with selfishly routing users where specific arcs are completely banned for some users. They analyse under which conditions a Nash equilibrium exists and whether a different equilibrium would exist when some arcs are forbidden in the network. Georgiou et al. (2006) study the problem of selfish routing in networks with incomplete information. The network users route along paths with minimum latency. The challenges of the problem are incomplete information about arc capacities, modelled by user-specific pay-off functions. Karakostas and Kolliopoulos (2009) and Bonifaci et al. (2010) capture the problem of selfish routing by using Stackelberg routing. The network users are divided into two parts, a part of the users takes routes according to a central decision maker strategy and the other part of the users acts selfishly. With their route choices the selfish users react to the decisions of the coordinated ones. The central decision maker optimises the system optimum and the selfish users minimise their own costs. Christodoulou et al. (2014) present a coordination mechanism to reduce the price of anarchy. For the reduction, they identify arcs in the network that result in a high price of anarchy.

## 4.2 Braess's Paradox

That the behaviour of independent acting agents with different goals does not lead to system's optimum is unsurprising. Braess's paradox, a phenomenon that occurs in the theory of transportation science is less intuitive. This paradox was first mentioned by Braess (1968). In his study he investigates the traffic assignment in road networks with given start- and endpoints, and traffic dependent travel times. He distinguishes between traffic assignments close to the definition of the user- and system-optimal flows by Wardrop (1952). His study deals with the construction of road networks and an optimal traffic assignment on these roads. Moreover, he considers individual travel behaviour. He predicts that the optimal travel time for each network user does not result in the optimal travel time for the total system, which was also illustrated in Section 4.1. The travel time on a path does not solely depend on the characteristics of the road e.g. the length or the maximal speed, it also depends on the traffic density on a road. As shown in the fundamental diagram of traffic flow (Figure 3.2), the traffic flow decreases with increasing traffic density until a certain traffic density in a road segment is reached. This high level of traffic density leads to an increase in travel time. Besides the traffic assignment on the given road network Braess studies network construction operations that should improve the travel time in the system. He shows that an additional road can increase the overall travel time in the street network. The phenomenon occurs when the additional road will lead to an individual faster path for the network users and all individuals in the network will optimise their own travel time. In such a situation all network users chooses this additional path. This in turn increases the traffic density along this path, as well as the total travel time in the network. It is not intuitive that a capacity increasing road leads to an increase of travel time, but Braess (1968) shows that this phenomenon can occur with selfishly acting individuals depending on the network structure. The following example, which is based on the example by Roughgarden (2005), is used to illustrate this phenomenon.




(a) Original Network.




(b) Augmented Network.

Figure 4.2: Network to Illustrate Braess's Paradox.

The network is represented by a directed graph $G = (V, E)$, with nodes $V = \{A, B, C, D\}$ and a set $E$ that contains the arcs between the nodes. The traffic flow $x$ travels from the source node $A$ to the destination node $D$. The travel time $C_i(x)$ on an arc ($i \in E$) depends on the traffic flow on this arc. Figure 4.2 illustrates the original network with four arcs (Figure 4.2a) and the augmented network with five arcs (Figure 4.2b). To show the Braess paradox the average travel time of user-optimal flows and system-optimal flows is compared between the original network and the augmented network.

*Original network*

Figure 4.2a shows the original network with four nodes and four arcs. The travel time in the example on the arcs $b$ and $c$ is set to 1 hour and the travel time on the arcs $a$ and $d$ equals the traffic flow $x$ on these arcs (in hours). All network users travel from node $A$ to node $D$ on the routes 1 (A-B-D) and 2 (A-C-D). The fraction of traffic that traverses route 1 is $x_1$ and the fraction of traffic on route 2 is $x_2$, and the total traffic flow within the network, is $x = x_1 + x_2 = 1$. The travel time is $C_1(x_1) = C_a + C_c = 1 + x_1$ on route 1 and $C_2(x_2) = C_b + C_d = 1 + x_2$ on route 2. These definitions lead to an average travel time in the network of

$$C_{avg}(x_1, x_2) = C_1(x_1) \times x_1 + C_2(x_2) \times x_2. \tag{4.8}$$

The system-optimal flows are achieved by the minimum average travel time. With the travel times, stated in the example, this time is reached by traffic flows $x_1 = x_2 = 0.5$. The traffic flow is evenly distributed over both routes and this assignment leads to an average travel time of 1.5 hours. The user-optimal flow is achieved, when the travel time is the same on all used routes

$$C_1(x_1) \times x_1 = C_2(x_2) \times x_2. \tag{4.9}$$

The user-optimal flows are achieved by distributing the traffic evenly over routes 1 and 2: $x_1 = x_2 = 0.5$. The user-optimal flow equals the system-optimal flow, and both flows result in a travel time of 1.5 hours in the original network. It can be summarised that in the network in Figure 4.2a selfish routing does not affect the overall travel time.

*Augmented network*

In the augmented network (Figure 4.2b) an additional arc $e$ between the nodes B and C is built with travel time $C_e(x) = 0$. Intuitively this additional arc between B and C with travel time 0, should reduce the average travel time in the network. With the new arc, an additional route 3 is created to travel from A to D: A-B-C-D, with a travel time of $C_3(x_1, x_2, x_3) = x_1 + x_2 + 2x_3$. The travel time on the existing routes 1 and 2 must be adjusted to $C_1(x_1, x_3) = x_1 + x_3 + 1$ and $C_2(x_2, x_3) = x_2 + x_3 + 1$. Considering these three routes the average travel time in the network is

$$C_{avg}(x_1, x_2, x_3) = C_1(x_1, x_3) \times x_1 + C_2(x_2, x_3) \times x_2 + C_3(x_1, x_2, x_3) \times x_3. \tag{4.10}$$

System-optimal flows are achieved by the minimum average travel time. For the presented example these flows are reached, when the traffic will be evenly distributed over the routes 1 and 2 and no traffic will be assigned to route 3: $x_1 = x_2 = 0.5$ and $x_3 = 0$. With this assignment the minimum average travel time is 1.5 hours, which is the same solution as in the original network. User-optimal flows are achieved when the travel time is the same on all used routes

$$C_1(x_1, x_3) \times x_1 = C_2(x_2, x_3) \times x_2 = C_3(x_1, x_2, x_3) \times x_3. \tag{4.11}$$

User-optimal flows will be reached, when all network users use route 3. The travel time on route 3 will never be longer than the travel time on routes 1 or 2. But if some network users do not use

route 3, the travel time will be less than on the other routes. The selfishly acting network users have an incentive to use the new route 3. But by assuming that all network users take route 3 the average travel time will be 2 hours. The selfish behaviour leads to an increased travel time in the augmented network.
In the original network in Figure 4.2a the system- and user-optimal flows are the same and the traffic is distributed evenly over both routes 1 and 2. In the augmented network in Figure 4.2b the additional arc e between the nodes B and C is not part of the solution of the system-optimal flows and the minimum average travel time remains the same as in the original network. In the solution with user-optimal flows the network users have an incentive to take route 3: if route 3 is not used by all network users the resulting travel time on route 3 will be lower as on routes 1 and 2. But if all network users take route 3, the average travel time will be higher compared to the system-optimal flows. The average travel time in the augmented network with user-optimal flows is increased by half an hour compared to the travel time in the original network. The example reveals the findings by Braess (1968), where in a worst case scenario a travel time decreasing arc can increase the travel time for the total network. Roughgarden (2005) points out two important aspects for selfish routing in networks, which result from the Braess paradox:

- Selfish uncoordinated routing could lead to a solution that none of the network users would prefer and
- the structure of the network in combination with selfish routing has a significant influence on the travel time and defies intuition.

Murchland (1970) shows that Braess's paradox is not only a theoretical concept. The traffic phenomenon described above happened in Stuttgart after building new roads.

*Literature review*

Since Braess has identified this paradox in the theory a lot of research deals with this topic. Since the given literature review focuses on studies that deal with the Breass paradox in transportation networks, for the sake of completeness the reader is referred to some literature about Braess's paradox in communication networks: Korilis et al. (1999), Huang et al. (2006), Nagurney et al. (2007), and Hsu and Su (2011). In the research area of Braess's paradox in transportation management, a part of studies in the literature examines the conditions in the network that lead to the phenomenon and another part develops methods to avoid the paradox. Early studies that investigate network conditions that lead to the paradox are published by Frank (1981) and Steinberg and Zangwill (1983). Frank (1981) presents a complete mathematical characterisation of the paradox for an initial four link network and an augmented five link network. She proves that Braess's paradox even exist when not all arcs are used as originally anticipated in Braess example. Moreover, she determines the critical range of flows where Braess's paradox takes place. Steinberg and Zangwill (1983) state two simple conditions that lead to the paradox: the networks must be congested and the additional routes in the network must be cheaper than the existing routes. Contrary to the simple example by Braess (1968) they investigate networks where a new link leads to more than one additional route and networks with non-linear arc costs. Their findings are later generalised by Dafermos and Nagurney (1984). Pas and Principio (1997) determine the conditions and bounds that either lead or do not lead to the occurrence of Braess's paradox. First they examine demand levels that result in the Braess paradox and they are able to show that this paradox does not occur when significantly low or high demand levels are applied. For a second condition they investigate the marginal arc costs of additional links. They show that the paradox does not occur when particular marginal costs are applied. Nagurney (2010) prove that, an increase in demand can prevent the Braess paradox in networks where it originally occurred. Roughgarden (2006) investigates network designs that prevent Braess's paradox. He examines two cases: removing arcs from and adding arcs to an existing network. For both cases he determines arcs that does not lead to Braess's paradox. Valiant and Roughgarden (2010) prove the existence of Braess's paradox for large random networks and show that in such networks the paradox occurs with a high probability. Moreover, they show that removing arcs from a network can improve the overall network performance. Askoura et al. (2011) introduce the concept of building sub-networks where Braess's paradox does not appear. They formulate an optimisation model that determines routes which prevent the occurrence of the paradox and deletes all routes out of the network that are not used in the optimal solution. A heuristic based on a genetic algorithm is developed by Bagloee et al. (2013). With the heuristic, they detect arcs that cause the paradox of Braess and compute a combination of these arcs that best prevents the paradox. An experimental point of view on the emergence of Braess's paradox is examined by Rapoport et al. (2009) and Rapoport et al. (2014). Some simulation studies are done in Bazzan and Klügl (2005). In their study they influence the behaviour of the network users to avoid the paradox.

## 4.3 Summary

Section 4.1 and Section 4.2 have illustrated the effects of selfish routing on the traffic flow in networks and have demonstrated how the network conditions reinforce these effects. In most cases selfish routing leads to a suboptimal distribution of traffic in networks, because the network users take the best routes for themselves without considering the route choice of the other network users. Under the assumption that all network users have the same preferences, this behaviour leads to a traffic distribution where one route is extensively used while other routes remain unused. The Braess paradox, which was discussed in Section 4.2, clearly illustrates that the network structure intensifies the effects which arises from selfish routing. Moreover it shows that additional arcs in the network can reinforce the effects of selfish routing, if these arcs will lead to a route that is preferred by the network users. When all of them take this route, the network is congested and the traffic flow decreases. The findings of this chapter can be summarised with two points:

- Selfish routing leads to suboptimal traffic flows in a network.
- The network conditions can intensify the effect of selfish routing.

A question that arises from these findings is how these negative effects of selfish routing can be limited. From the first aspect one can conclude that the behaviour of the network users must be influenced, in such a way that their behaviour leads to an optimal traffic distribution. To achieve that, some of the network users must be forced to use alternative routes (others than their original preferences) which will then lead to an optimal traffic distribution. According to the second aspect selfish behaviour can be limited by optimising the network. Braess's paradox points out, that the network structure will have a negative influence on the total traffic flow if the network users behave selfishly. Moreover, it shows that specific arcs, which intensify the effect of selfish routing in a network, cause this paradox. These findings reinforce the thesis that the traffic flow must be allocated to other routes as the selfish ones. But how can selfish acting network users be persuaded to switch their routes? In Section 4.1 and 4.2 various studies are presented that have investigated this question. The results of the most common approaches in literature can be summarised as:

- Selfish acting network user can be persuaded to switch their routes by increasing the costs on arcs which intensify the effect of selfish routing.
- The distribution of network users can be influences by prohibiting certain arcs for some of the network users.
- The Braess paradox can be avoided by removing arcs from the network.
- The Braess paradox can be avoided by building (sub-)networks that do not contain arcs which lead to Braess's paradox.

The first aspect influences the preferences of network users for specific arcs. With increasing costs some arcs become less attractive. Here, the costs are not exclusively monetary, they could also be seen as other instruments that make an arc less attractive to some of the network users, for example by imposing speed limits. In the second aspect an optimised traffic assignment arises from prohibiting some arcs for some network users. The first two aspects influence the traffic without modifying the network. In contrary the third and fourth aspect modifies the street network. These modifications remove arcs from a network or build networks where the Braess paradox does not occur.
This thesis transfers the aspects of selfish routing users in networks to the process of evacuating urban areas. The evacuation is optimised, by preventing the negative effects of selfish routing and avoiding Braess's paradox. Some aspects of the approaches, which are described above, are used to counteract the negative consequences of selfish routing. Chapter 5 focuses on the strategies for temporal street network modification and provides a more detailed explanation to the reader.

# Chapter 5

# Selfish Evacuation Routing

In this chapter the general topic of selfish routing is transferred to evacuation scenarios. Section 5.1 illustrates the effects of selfish behaviour in an evacuation scenario. Section 5.1.1 discusses why the behaviour of humans in evacuations should be assumed as selfish and the effect of selfish behaviour on the traffic flow is shown. Moreover in this section the evacuees are defined and a model for evacuee's route choice is presented. Section 5.1.2 illustrates the prosed strategy of blocking connections between street sections to guide the traffic flow out of the affected area. Section 5.2 presents the test bed, which is used for the computational studies from Chapter 6 to Chapter 8, and proposes reference values to estimate the solution quality of the developed strategy.

## 5.1 Selfish Routing in Evacuation Planning

### 5.1.1 Selfish Route Selection of Evacuees

In an extraordinary situation like an evacuation, it can be assumed that people behave selfishly to protect their lives and those of their families. This general selfish behaviour can be transferred to the route selection made by evacuees. Sadri et al. (2014) investigate routing strategies during hurricane evacuation scenarios. Thus, they have evaluated data from a household survey conducted after hurricane Ivan in September 2004, in the western of Gulf Shores, Alabama. The survey shows that less than 3.9 % of the interviewed people would follow the instructions of authorities. In contrast more than 90 % of the interviewed people prefer a familiar or usual route, which they consider as the shortest, fastest or less congested route. Similar results are presented in Murray-Tuite et al. (2012). They compare the behaviour of people during hurricane Katrina in 2005 and Ivan in 2004. One aspect they investigate is the route selection of evacuees. Also during hurricane Katrina, most evacuees choose familiar routes, which they considered as the fastest or shortest one. Moreover, most evacuees did not like to follow instructions of authorities, which is similar to the observed behaviour during hurricane Ivan. These results facilitate the assumption that evacuees behave selfishly in evacuation scenarios. As is known from the research in selfish routing in traffic networks (see Chapter 4) selfish behaviour has negative effects on the traffic flow. These results have to be adopted to traffic flows in evacuations. If the evacuees behave selfishly the resulting traffic flow will lead to an increase in the time that is necessary to clear the affected area. Hence, it is necessary to consider the behaviour of evacuees when compiling evacuation plans. An example is used to point out the problem of selfish routing in evacuation scenarios. Therefore, the network clearance time (NCT) of a solution with system-optimal flows is compared to the NCT that result from selfishly selected routes.

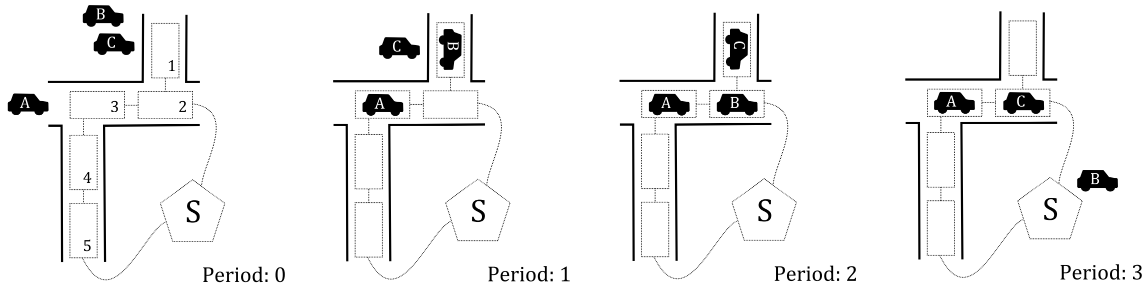



Figure 5.1: Network with Selfish Evacuees.

The network in Figure 5.1 is a simplified street network represented by sections. It is assumed that each section has a capacity of one vehicle per period and that a vehicle can pass maximally one section in one period. There are five sections in the affected area and super sink $S$ depicts the safe places. Furthermore, there are three evacuees A, B, and C in the network: evacuee A begins the evacuation in section 3 and evacuees B and C start in section 1. In this example it is assumed that the evacuees prefer the fastest route, which is a route with a minimum number of sections. These routes will lead to the user-optimal flows. As illustrated in Figure 5.1 evacuee A takes route 3 - 2 - $S$ and evacuees B, C take route 1 - 2 - $S$. The NCT is 5 periods for the example with selfish routing. In this example section 2 has a high traffic density and evacuee A has to wait two periods before he can pass through this section. In the system-optimal solution evacuee A traverses 3 - 4 - 5 - $S$ and evacuees B, C take the route 1 - 2 - $S$. These routes lead to an NCT of 4 periods. In the solution with selfish routing the NCT is higher compared to the solution with system-optimal flows, although evacuee A must pass fewer sections.
The example illustrates that selfish behaviour leads to a suboptimal solution for the network users in case of evacuation. The example detects a drawback of several evacuation plans: most studies in evacuation planning optimise the traffic flows assuming that the evacuees follow the guidance information from authorities, and compute optimal routes, departure times or assignments to destinations. Abdelgawad and Abdulhai (2013) criticise that it cannot be assumed that the evacuees behave as determined in these evacuation plans. They make clear that the evacuees tend to a selfish behaviour which results in user-optimal traffic flows. Thus the behaviour of the evacuees and the findings that most evacuees will not follow the instructions of authorities have to be considered in evacuation traffic management. Hence, it is useful to revert to the strategy described in Chapter 4 to prevent the consequences of selfish routing. Most methods that were developed to prevent the consequences of selfish behaviour try to solve the problem by modifying the street network. In contrast to the long-term modifications, which are used in general traffic management, in evacuation planning just short-term modifications are possible. However, modifications like the addition of arcs or the invention of taxes on certain arcs are not practicable in the case of evacuations. In this thesis possible short-term network modifications are proposed. One key-suggestion is to block connections between street sections to force the evacuees to divert from their selfishly chosen routes to alternative routes, which will finally lead to a more efficient traffic flow. In Section 5.1.2 this strategy is described in more detail.

*Definition of evacuees and route modelling*

Prior to explain how the selfish routes of the evacuees are determined, the assumptions that define the evacuees are explained. In this thesis the following evacuee characteristics are taken as a basis for the modelling:

- Evacuees are modelled as independently acting agents which determine their routes without considering the behaviour of the other network users.
- The evacuees have preferences for their routing strategy. These preferences are expressed by the routes the evacuees take.
- Following the results of Sadri et al. (2014) it is assumed that the evacuees stick to the chosen routes during the evacuation.
- The starting points of evacuees in the network are known.

The selfish routes are modelled as the optimal routes depending on the preferences of the evacuees. As know from the literature evacuees tend to take the most familiar routes, which are in most cases the shortest or fastest ones (see Lindell and Prater (2007), Murray-Tuite et al. (2012), Sadri et al. (2014)). A mathematical model is presented to define the route choice of the evacuees. In this model the different preference types are expressed with weights $q_{ip}$ for every section $i \in I$ depending on the preference type $p \in P$. Where $P$ is the set of the different preference types. For example if the fastest ($p = 1$) route should be considered in the model, $q_{i1}$ will be 1 for each section $i \in I$. One period is needed to pass through a section so the number of sections, which is used in the route must be minimised to compute the fastest route. If the preference type is "shortest route", the weight for the sections will be $N_i/l_i$, which is the capacity of one lane ($N_i$ is the capacity and $l_i$ is the number of lanes of a section). The capacity of a lane reflects the length of a section: when computing this capacity the number of vehicles that fits into a section is multiplied with the vehicle length (plus safety distance between two vehicles). Thus a high capacity reflects a long section. A mathematical program is formulated that computes the best route for every evacuee $e \in B$ from his start section $a_e \in I$ to the super sink $S \in I$. The preference type $p$ of an evacuee $e$ is expressed by the binary parameter $\gamma_{ep}$. The variable $r_{ije}$ determines the routes of the evacuees. The variable will be 1, if sections $i$ and $j$ are in sequence on the route of evacuee $e$, otherwise it will be 0. Note that the problem to be solved is a shortest path problem which can efficiently be solved and a mathematical model is just given for the sake of clarity. A mathematical model formulation to find the selfish route of evacuee $e \in B$ can be stated as follows

$$\min \sum_{i \in I} \sum_{j \in I} \sum_{p \in P} r_{ije} q_{ip} \gamma_{ep} \tag{5.1}$$

s.t.

$$r_{ije} \leq \beta_{ij} \qquad i \in I, j \in I \tag{5.2}$$

$$\sum_{j \in I} r_{a_e je} = 1 \tag{5.3}$$

$$\sum_{i \in I} r_{iSe} = 1 \tag{5.4}$$

$$\sum_{i \in I} r_{ije} = \sum_{k \in I} r_{jke} \qquad j \in I \backslash (a_e \cup S) \tag{5.5}$$

$$r_{ije} \geq 0 \qquad i \in I, j \in I. \tag{5.6}$$

It is the objective of the model to find an optimal route (according to the preferences) for each evacuee $e$ from the start section $a_e$ to the super sink $S$. Thus the objective function (5.1) minimises the weighted sum over the sections, which are in the route. Constraints (5.2) ensure that two sections can be in sequence on a route if the sections are connected in the network ($\beta_{ij} = 1$). Constraints (5.3) and (5.4) ensure that the route starts in section $a_e$ and ends in the super sink $S$. Moreover, the restrictions (5.3) - (5.5) eliminate sub-tours, because it is assumed $\beta_{Si} = 0$ for all $i \in I$. Constraints (5.5) are the traffic flow constraints. The domain of the variable $r_{ije}$ is stated in (5.6). Note that evacuees who are initially located in the same section and who have the same preference type choose the very same route. Also, other evacuees with that preference type who are located in a section on that route would follow them. Hence, these many shortest path problems can be solved better than just enumerating all sections and preference types. The algorithm by Dijkstra (1959) is used to compute the best path for every preference type and every section. From these paths the routes of the evacuees can be derived.

### 5.1.2 Blocking of Street Sections

As described in Section 5.1.1 selfish route selection of evacuees could result in a solution that is unfavourable for all network users. Hence, a strategy is developed that counteracts these negative impacts. As discussed above the network should be modified to cope with these selfish evacuees. In case of an evacuation network modifying possibilities are limited due to e.g. a short time span or sparse relief units that can implement these modifications. Thus the methods from the literature to reduce the consequences of selfish routing e.g. building of new roads or increasing the costs for the network users with taxes, are not feasible in evacuation scenarios. To prevent a practical solution the concept of blocking connections between street sections in the network was developed. With such blockages the network users are forced to deviate from their selfish routes. The Braess paradox (see Section 4.2) demonstrates that an additional, capacity increasing link can increase the travel time in a network for all network users, if they act selfishly. These findings are used the other way round, and street sections are blocked to decrease the travel time and reduce the network clearance time. The temporal blockage of street sections can be quickly implemented without high efforts of relief units.

*The effect of blocking street sections*

In Section 5.1.1 it is demonstrated that selfish network users choose a route, which is optimal according to their preferences, without considering the behaviour of the other network users. In most cases the chosen routes will lead to a suboptimal traffic distribution and result in a high network clearance time. If a large proportion of the network users takes the same routes to leave the network, these routes will be congested. The high traffic density will reduce the traffic flow and the time, which the evacuees spend in the affected area, increases. Hence, it is necessary to reduce the traffic density on congested streets to increase the traffic flow and to decrease the time which the evacuees spend in the affected area. Accordingly, the strategy of blocking connections between street sections is investigated in this thesis. When certain connections between street sections are blocked some evacuees cannot maintain their selfish routes. They have to divert from their original routes and have to choose a new route, which is optimal according to their preferences, but in the adjusted network. These blockages must be positioned between street sections in such a way, that the resulting traffic flow is redirected from the affected area in a faster way. The general concept is illustrated in the simplified street network representation in Figure 5.2.

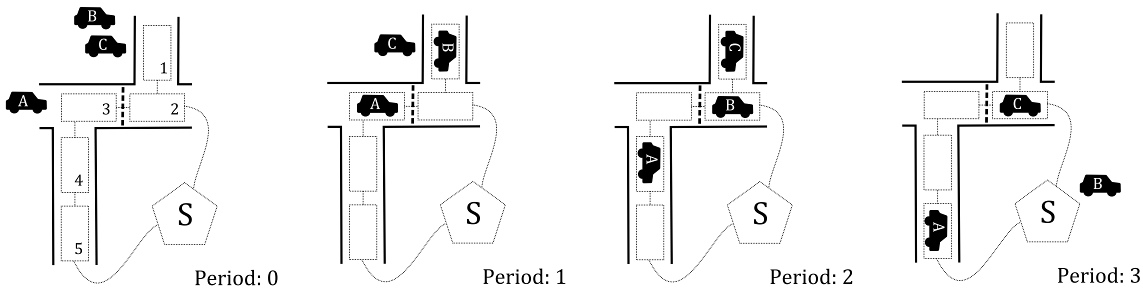


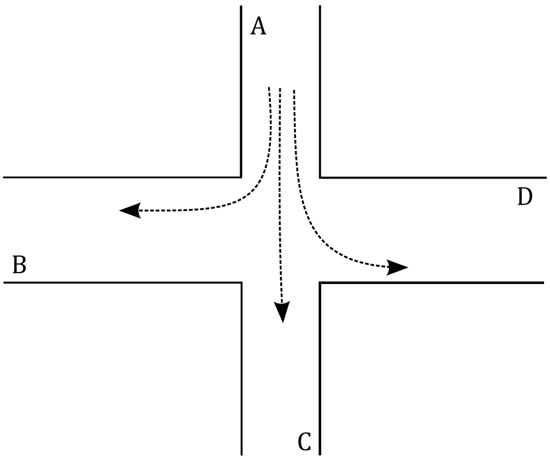

Figure 5.2: Network with Selfish Evacuees and one Blockage.

Figure 5.2 represents an adjusted version of the network from Figure 5.1 and induces a blockage between sections 2 and 3 (dotted line). Without blockage the selfish route for evacuee A was 3 - 2 - $S$ and those of the evacuees B,C were 1 - 2 - $S$. Now, by inducing the blockage between sections 2 and 3, evacuee A is forced to divert from his original selfish route. When assuming that the evacuees prefer the fastest route, in the adjusted network the new fastest route for A is: 3 - 4 - 5 - $S$. Although this route is one section longer, the adjusted route leads to a shorter NCT. The NCT in the network without any blockages was 5 periods, with the blockage the NCT is reduced to 4 periods. By blocking the connection between sections 2 and 3 less vehicles use the formerly highly congested street section 2 and therefore lead to an increase in total traffic flow. In the original network (Figure 5.1) evacuee A had to wait for two periods before he could move on to the next section of his route. In the adjusted network he can drive along his route without any waiting time. The example points out that the induction of blockages between street sections can regulate the traffic in a way that leads to increased traffic flow in the whole network and reduces the time the evacuees have to spend in the affected area.
In such a simplified street network representation it is easy to determine connections that should be blocked to adjust the network in such a way that the network clearance time is reduced. In this example the blockage only affected one route and lead to the usage of formerly unused sections. But in a lager network with many connections between sections it is hard to determine the locations for blockage. In contrast to the simple example network in Figure 5.2 in a lager network one blockage can affect many routes. Moreover, the newly determined routes that result from this blockage lead to an increased traffic flow in one part of the network while it is decreased in another part. Furthermore, not only the traffic density would affect the NCT, also the length of the alternative routes affects the NCT. These impacts are demonstrated by means of the example in Figure 5.2. First it is assumed that an additional evacuee starts in section 3, who prefers the route 3 - 4 - 5 - $S$. The blockage of the connection between sections 2 and 3 leads to a decreased traffic flow in section 2 while it increases in section 4 and 5. Therefore, the resulting traffic would not lead to a decreased NCT. Second it is assumed that there is an additional section 6 between section 5 and the super sink $S$. Accordingly, the alternative route that was generated for evacuee A increases in length: 3 - 4 - 5 - 6 - $S$ (instead of 3 - 4 - 5 - $S$). If evacuee A is guided via the enlarged alternative route a reduction in NCT is not possible. In

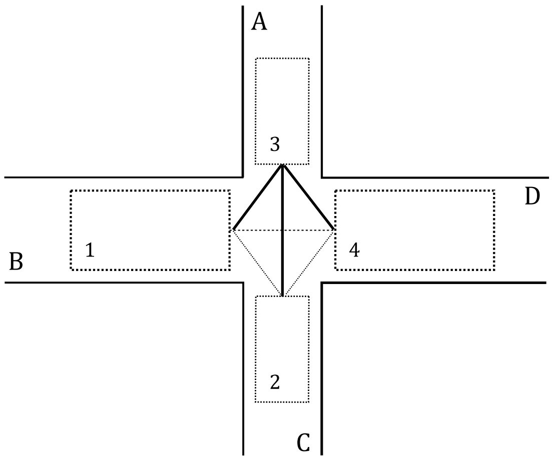
both scenarios the blockage between sections 2 and 3 would not help to reduce the NCT. Hence the example points out that the length of the alternative routes, the traffic flow as well as the resulting traffic density must be considered when determining blockage positions.

*Various possibilities for positioning blockades in the network*

The previous paragraph described that the induction of blockages between adjacent sections is used to re-distribute the traffic in such a way that the resulting traffic flow leads to the smallest possible NCT. The blockage of connections between street sections can lead to a partial blockage of a street section or to a complete blockage of a street section. Figure 5.3 illustrates the different possibilities for positioning blockades in a network. The example shows a regular intersection: the street sections are labelled with letters A to D. Figure 5.3a presents an intersection with all possible connections from the perspective of street A. Being in street A a vehicle can drive to streets B, C, and D.




(a) Intersection with a Multitude of Connections.




(b) Intersection Represented as Sections.

Figure 5.3: Street Network to Illustrate Various Possibilities for Positioning Blockages.

In Figure 5.3b the street network from Figure 5.3a is transformed into sections. Section 3 (corresponding to street A) is connected with sections 1, 2, and 4 (bold lines). In order to block the complete 3rd section the connections to all branching sections must be blocked: 3 - 1, 3 - 2, and 3 - 4. As described above it is possible for instance to prohibit only the left turn, thus only connection 3 - 4 is blocked. In those parts of the network that do not represent intersections there is just one connection between two sections. This means, if a blockage is positioned at this section, always the whole section will be blocked.
The blockage of connections between street sections enables an accurate traffic control without the need to determine routes for network users. For networks with a large number of these connections the combinatorial effort to determine the connections to be blocked is very high. Chapter 6 induces a traffic-guiding concept that builds sub-networks with a single exit, by blocking complete sections of the original network. In Chapters 7 and 8 a concept is introduced where the positions of the blockages are determined more precisely.

## 5.2 Components of the Computational Study

### 5.2.1 Data and Assumptions

In this thesis a test bed with 360 basic instances is used for the computational studies made in Chapter 6 to 8. The computational studies are run on a standard machine with Intel (R) Core (TM)i5-3470, 3.2 GHz and 16 GB Ram. The heuristics and the model are coded in AMPL and the models are solved by Gurobi 6.5.2. In the following chapters the parameters of these basic instances are adapted to different research questions. With the software tool SPSE, which was developed in our research group, nine real-world networks were generated for the basic instances. SPSE is a tool to determine evacuation plans on the basis of the CTM. Hence, SPSE is suitable to generate example networks (determine the $\beta$ parameter) for the computational study (for further information about the software tool SPSE the reader is referred to SPSE (2016)). As already mentioned the instances are constructed by using the nine real-world networks. The networks are subdivided into three categories based on the number of sections:

(1) small networks with 50 - 80 sections,

(2) medium networks with 81 - 120 sections,

(3) large networks with 121 - 160 sections.

The number of exits in each network depends on the conditions of the real network and represents the connections between sections within the affected area and sections outside the affected area. Some connections that area identified as exits (by the software tool), as they connect the affected area with the safe area, are deleted by hand because e.g. they lead to dead-end roads. The networks are illustrated in Appendix A. For reasons of simplicity it is assumed that a maximum speed of 30 $km/h$ is allowed in the streets in all networks; the length of a period is set to 9 seconds. These conditions result in section capacities $N_i = 14$ and $Q_i = 6$ for all $i \in I$ in all networks. Moreover, all streets are modelled with one lane only.
The evacuees in the networks are generated by means of the following conditions: there are four sets of evacuees (500, 700, 800, and 900) which are randomly distributed over the networks. For each network and each set of evacuees 10 different distributions are randomly computed. The preference types are randomly distributed over the set of evacuees, too. The conducted computational studies take three different preference types of evacuees into account: evacuees that prefer the shortest, the fastest and a random path, while the type with the random path should represent an evacuee without knowledge of the network. These preference types are considered, because these are the most common behaviour patterns that are observed in the literature (see Murray-Tuite et al. (2012) and Sadri et al. (2014)). Given that by assumption all sections have the same length the types shortest and fastest path take the same path, which is a path with a minimum number of sections. For the random type, a random number (uniform distributed) between 1 and 8 is generated for each section. The sum of these random numbers is minimised for the random path (see Section 5.1). In Table 5.1 the parameters of all instances are summarised. The ID of each instance consists of the network name, the number of evacuees distributed over the network and a letter $a$ to $j$ that indicates the different random distributions of evacuees over the network.

| Instance ID | Network | Evacuees (x) | Distribution of Evacuees (*) | Number of Exits | Number of Sections |
|---|---|---|---|---|---|
| B_x_* | Berlin (B) | x = 500, 700, 800, and 900 | *= a - j | 14 | 68 |
| S_x_* | Stockholm (S) | x = 500, 700, 800, and 900 | *= a - j | 10 | 76 |
| P_x_* | Paris (P) | x = 500, 700, 800, and 900 | *= a - j | 11 | 79 |
| SY_x_* | Sydney (SY) | x = 500, 700, 800, and 900 | *= a - j | 14 | 85 |
| M_x_* | Melbourne (M) | x = 500, 700, 800, and 900 | *= a - j | 11 | 100 |
| A_x_* | Auckland (A) | x = 500, 700, 800, and 900 | *= a - j | 12 | 120 |
| L_x_* | Lima (L) | x = 500, 700, 800, and 900 | *= a - j | 20 | 140 |
| NY_x_* | New York (NY) | x = 500, 700, 800, and 900 | *= a - j | 20 | 152 |
| D_x_* | Dubai (D) | x = 500, 700, 800, and 900 | *= a - j | 19 | 158 |

Table 5.1: Summary of the Network Parameters of the Test Bed.

### 5.2.2 Reference Values

To estimate the solution quality of the computed solutions reasonable reference values (bounds) are necessary. Thus two reference values are introduced: the NCTs that result from the system-optimal and the user-optimal flows. The computation of these values is described below. The system-optimal traffic flow arises from the routes in the solution where the NCT is minimised. To compute these routes a model that is closely related to the formulation presented in Section 3.2 is used. The variables $\nu_t$ are used to count the periods in which the evacuees stay in the network. The variables $x_{it}$ determine the number of evacuees in sections, the variables $y_{ijt}$ represent the traffic flow between two sections and the variables $b_{it}$ specify the number of evacuees that starts the evacuation in a period. The parameters $N_i$, $Q_i$ are used to restrict the traffic flow and $E_i$ is the number of evacuees which is located at a section. The model formulation that computes the system-optimal flows reads as follows

$$\min \sum_{t \in T} \nu_t \tag{5.7}$$

s.t.

$$x_{it} = b_{it} + x_{i,t-1} + \sum_{j \in I} y_{jit-1} - \sum_{j \in I} y_{ijt} \qquad i \in I, t \in T \tag{5.8}$$

$$x_{it} + \sum_{j \in I} y_{ijt} \leq N_i \qquad i \in I, t \in T \tag{5.9}$$

$$\sum_{t \in T} b_{it} = E_i \qquad i \in I \qquad (5.10)$$

$$x_{S|T|} = \sum_{i \in I} E_i \qquad (5.11)$$

$$y_{ijt} \leq N_i \beta_{ij} \qquad i \in I, j \in I, t \in T \qquad (5.12)$$

$$\sum_{j \in I} y_{jit} \leq N_i - x_{it} \qquad t \in T, i \in I \qquad (5.13)$$

$$\sum_{j \in I} y_{jit} \leq Q_i \qquad i \in I, t \in T \qquad (5.14)$$

$$\sum_{j \in I} y_{ijt} \leq Q_i \qquad i \in I, t \in T \qquad (5.15)$$

$$\nu_t \sum_{i \in I} E_i \geq \sum_{i \in I} E_i - x_{S,t} \qquad t \in T \qquad (5.16)$$

$$b_{it} \geq 0 \qquad i \in I, t \in T \qquad (5.17)$$

$$y_{ijt} \geq 0 \qquad i \in I, j \in I, t \in T \qquad (5.18)$$

$$x_{it} \geq 0 \qquad i \in I, t \in T \qquad (5.19)$$

$$\nu_t \in \{0, 1\} \qquad t \in T. \qquad (5.20)$$

The objective function (5.7) minimises the number of periods where evacuees are in the network. The variable $\nu_t$ indicates that evacuees either stay or do not stay in the network and constraint (5.16) ensure that $\nu_t = 1$ if evacuees are in the network and $\nu_t = 0$, if all evacuees have left the network. Constraints (5.8) - (5.11) regulate that all evacuees start the evacuation and leave the affected area in the planning horizon. And the constraints (5.12) - (5.15) ensure that the traffic flow conditions are considered. The domains of the decision variables are stated in (5.17) - (5.20). The system-optimal flows are a lower bound for the network design problem with selfishly routing evacuees. The model (5.7) - (5.20) is a relaxation of the lower-level problem (7.10) - (7.28) (in Chapter 7), which is used to model the network design problem with selfishly routing evacuees. By relaxing the constraints (7.11) - (7.16) and (7.21) in model (7.10) - (7.28), the formulation stated above is created. Thus from the optimal solution of model (5.7) - (5.20) a lower bound can be derived, for the problem of traffic routing with selfishly acting evacuees.
The user-optimal flows arise from the selfish routes of the evacuees. To compute the user-optimal flows the model formulation above must be extended by constraints (5.21)

$$\sum_{t \in T} y_{ijt} \geq \sum_{e \in B} r_{ije} \beta_{ij} \qquad i \in I, j \in I. \qquad (5.21)$$

The binary parameter $r_{ije}$ defines the routes of the evacuees, and will be one if the sections $i$ and $j$ are in sequence of a route of an evacuee (a detailed description of the computation $r_{ije}$ is given in Section 5.1.1). With $\sum_{e \in B} r_{ije}$, the number of evacuees whose routes encompass the connection between sections $i$ and $j$, is counted. Let $B$ be the set of all evacuees $e$. These constraints ensure that a connection is used by exact the number of evacuees whose routes encompass such a connection. Thereby the user-optimal traffic flows are computed. The NCT that arises from these user-optimal flows is an upper bound of the network design problem. These user-optimal flows are a feasible solution of the model (7.10) - (7.28), which represents the lower-level problem of the network design problem with selfishly routing evacuees (Section 5.1.1). In this formulation the variables $m_{ij}$ indicate the blockage of connections between sections. To compute the user-optimal traffic flow the variables $m_{ij}$ must be fixed to parameter $\beta_{ij}$, and the constraints (7.11) -

(7.10) reduce to constraints (5.21). A solution with $m_{ij} = \beta_{ij}$ is a feasible solution for the model (7.6) - (7.28), because it equals to a solution without blockages. Hence, the optimal solution of the model (5.7) - (5.21) determines an upper bound for the problem of traffic routing with selfishly acting evacuees. In the computational studies theses bounds are used to estimate the quality of the presented concepts and heuristics.

# Chapter 6

# Using Sub-Networks to Guide the Selfish Evacuation Traffic

In Chapter 5 the importance of traffic regulation in evacuation scenarios with selfishly routing evacuees was described. Chapter 6 now introduces an approach that divides the original network into sub-networks with one exit only. This strategy restricts the possible route choices of the evacuees and hence guides the selfish evacuation traffic. Section 6.1 explains the general idea of computing these sub-networks and presents a general approach to implement this idea. Afterwards in Section 6.2 a mathematical model is introduced and the computation of sub-networks with this model is illustrated by an example. In Section 6.3 a heuristic to compute such sub-networks for networks of realistic size is presented. Section 6.4 introduces some implementation variants of the approach developed in Section 6.1. The chapter closes with a comprehensive computational study. The whole chapter is based on the paper by Kimms and Seekircher (2015).

## 6.1 Sub-Networks to Guide Selfish Evacuation Traffic

Section 5.1 pointed out the negative impact of selfish routing evacuees on the evacuation traffic management. Without any traffic regulation the full network capacities cannot be used and congestions in the network lead to decreased traffic flow. Hence, traffic regulations are necessary to optimise the evacuation process. In this section a concept is presented, that restricts the possibilities of evacuee's route choice by dividing the networks into sub-networks with only one exit. These sub-networks are constructed in such a way that the resulting traffic flow leads to a minimal NCT. Each sub-network to be constructed should be a connected graph and the different sub-networks should have different exits. In an iterative procedure the sections are assigned to their exit in a way, that the evacuation time is minimised. To compute the sub-networks the sections $i$ that are directly connected with the super sink $S$ are used as dummy exits, where $s$ is the notation for the exit sections. The set of (dummy) exits $A$ contains all sections which are connected with the super sink $S$: $A = \{i \in I : \beta_{iS} = 1\}$. The parameter $g_{is}$ is used as a weight to assign section $i$ to exit $s$ and indicates the travel time of an evacuee that starts in section $i$ and leaves the dangerous area through exit $s$ under the condition that section $i$ is assigned to exit $s$. To compute $g_{is}$ the traffic flow in the sub-networks is computed and $g_{is}$ is set to the longest travel time an evacuee needs from section $i$ to the exit $s$. The procedure works iteratively and for every $i, s$ combination the largest computed value for $g_{is}$ is saved over all iterations. Depending on the $g_{is}$ values that result from the iterative procedure, sub-networks with minimal evacuation time can be constructed: therefore sections are assigned to the exits in such a way that the sum over all used $g_{is}$ value is minimal. First the general approach on how to compute sub-networks and to determine the $g_{is}$ values will be specified and then it will be illustrated by a small example.

**Step 1**: Initially, all $g_{is}$ values are set to zero.

**Step 2**: Considering the original network, sections $i$ are assigned to the exits $s$ in such a way that the sum over the $g_{is}$ values is minimised taking into account that sections leading to the same exit must form a connected graph (see Section 6.2). The connections between street sections are blocked when these sections are assigned to different exits. Thereby the network is divided into sub-networks.

**Step 3**: The traffic flow in the sub-networks and the NCT are computed. The values for $g_{is}$ are updated when a $g_{is}$ value is larger than in the previous iteration.

**Step 4**: If at least one $g_{is}$ value has been updated in step 3 go back to step 2 using the new values of $g_{is}$, else go to step 5.

**Step 5**: The solution approach stops and the best solution is saved.

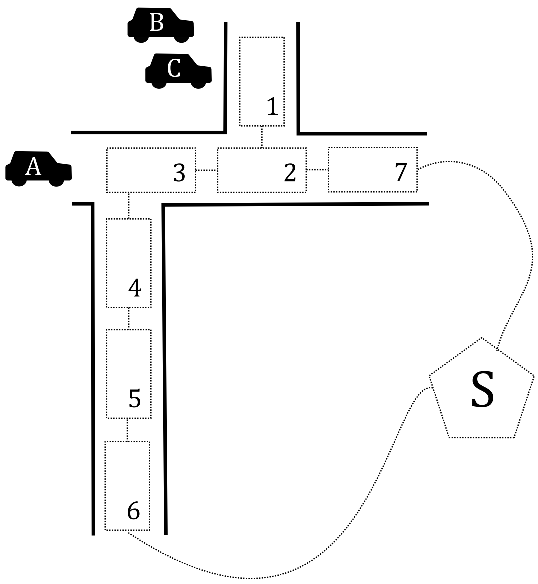



Figure 6.1: Network to Illustrate the Computation of Sub-Networks.

The network in Figure 6.1 is used to illustrate the computation of the parameter $g_{is}$ and the sub-networks. The sections 1 - 7 form the endangered area and the super sink $S$ represents the safe area. It is assumed that the evacuees B and C start at section 1 and evacuee A starts at section 3. All evacuees prefer the fastest route, which is a route with a minimum number of sections. Sections 6 and 7 are used as (dummy) exit sections, because they are connected with the super sink $S$. For the sake of simplicity in the example only sections 1 and 3 are assigned to an exit. The $g_{is}$ values are initially set to zero. It is assumed that in the first iteration the sections 1 and 3 are assigned to exit section 7, therefore all evacuees use exit 7 to leave the endangered area. The resulting evacuation times for the evacuees are presented in Table 6.1a. These times lead to an NCT of 5 periods and to an update of parameter $g_{is}$. The updated values are presented in Table 6.1b. As mentioned before for every $i, s$ combination the longest travel time is taken over all the evacuees that start in section $i$ and leave the network at exit $s$.

| Evacuee | Start Section | Exit | Evacuation Time |
|---|---|---|---|
| B | 1 | 7 | 3 |
| C | 1 | 7 | 4 |
| A | 3 | 7 | 5 |

(a) Evacuation Time for the Evacuees.

| $g_{is}$ | 6 | 7 |
|---|---|---|
| 1 | 0 | 4 |
| 3 | 0 | 5 |

(b) Updated $g_{is}$ Values.

Table 6.1: Results of the First Iteration.

According to the updated values of parameter $g_{is}$, in the second iteration sections 1 and 3 are assigned to exit 6. The new values for the evacuation times are depicted in Table 6.2a and the updated values of parameter $g_{is}$ are shown in Table 6.2b. This assignment leads to a NCT of 7 periods.

| Evacuee | Start Section | Exit | Evacuation Time |
|---|---|---|---|
| B | 1 | 6 | 6 |
| C | 1 | 6 | 7 |
| A | 3 | 6 | 4 |

(a) Evacuation Time for the Evacuees.

| $g_{is}$ | 6 | 7 |
|---|---|---|
| 1 | 7 | 4 |
| 3 | 4 | 5 |

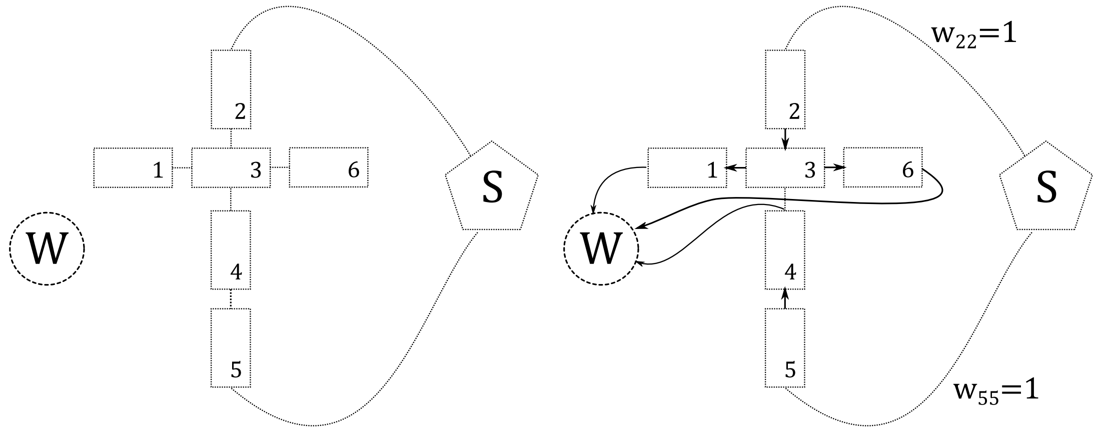
(b) Updated $g_{is}$ Values.

Table 6.2: Results of the Second Iteration.

With the updated values of parameter $g_{is}$, section 1 is assigned to exit 7 and section 3 is assigned to exit 6 in the third iteration. The resulting evacuation times after the third iteration for the evacuees are: (B) 3, (C) 4, and (A) 4. These times do not lead to an update of the values of parameter $g_{is}$ and the solution approach stops. The computed sub-networks are: {1, 2, S} and {3, 4, 5, S}. To construct these sub-networks the connection between section 2 and 3 must be blocked in the original network. The traffic flow in the sub-networks leads to an NCT of 4 periods, the same NCT results from the system-optimal flows.

## 6.2 A Mathematical Model to Compute Sub-Networks

For the optimal computation of sub-networks a mathematical model is introduced that assigns the sections to the exits of the network depending on the parameter $g_{is}$. By definition each sub-network has exactly one exit. $A$ is the set of exit sections and includes all sections which are connected with the super sink $S$: $A = \{i \in I : \beta_{iS} = 1\}$. For the example in Figure 6.2 this set is: $A = \{2, 5\}$.




(a) Example Network. (b) Illustration of a Feasible Solution.

Figure 6.2: Network to Illustrate the Computation of Sub-Networks with the Model (6.1) - (6.10).

To construct these sub-networks, directed paths are computed that connect all sections which belong to the same exit or sub-network, respectively. Therefore, the first section of each path is a section which is directly connected with the super sink, in the example sections 2 and 5. Starting from these sections paths are constructed where each section can be connected with exactly one predecessor section and with one or more successor sections. In Figure 6.2b for the section 3, the section 2 is the predecessor and the sections 1 and 6 are the successors. Moreover, each section belongs exactly to one path / sub-network. The arrows in Figure 6.2b depict a feasible solution for such a partition in sub-networks of the network depicted in Figure 6.2a. The computation of paths where each section has exactly one predecessor leads to an unique assignment from sections to exits (sub-networks). For example if in Figure 6.2b the connection $4-3$ would be also part of the solution, then section 3 would have two predecessors. In this case section 3 would be on the paths starting from section 2 or section 5 and an unique assignment of this section to a sub-network is not possible. Therefore, each section must be connected with exactly one predecessor to compute sub-networks as defined above. The section $W$ is a dummy successor section for the border sections of the sub-networks. The dummy section is necessary because each section needs at least one successor section. The binary variable $\zeta_{is}$ indicates whether section $i$ is assigned to exit $s$ or not. And the binary variable $w_{ij}$ indicates that section $i$ is the predecessor section of section $j$. The mathematical programming model reads as follows

$$\min \sum_{i \in I} \sum_{s \in A} g_{is} \zeta_{is} \tag{6.1}$$

s.t.

$$w_{ss} = 1 \qquad s \in A \tag{6.2}$$

$$\zeta_{ss} = 1 \qquad s \in A \tag{6.3}$$

$$\sum_{i \in I \setminus W : \beta_{ij}=1} w_{ij} = 1 \qquad j \in I \setminus W \tag{6.4}$$

$$\sum_{i \in I \setminus W : \beta_{ij}=1} w_{ij} \leq \sum_{i \in I : \beta_{ij}=1} w_{ji} \qquad j \in I \setminus W \tag{6.5}$$

$$\zeta_{js} \geq (w_{ij} - 1) + \zeta_{is} \qquad i \in I, j \in I, s \in A \tag{6.6}$$

$$\sum_{s \in A} \zeta_{is} = 1 \qquad i \in I \setminus W \tag{6.7}$$

$$w_{ij} + w_{ji} \leq 1 \qquad i \in I, j \in I \tag{6.8}$$

$$\zeta_{is} \in \{0, 1\} \qquad i \in I, s \in A \tag{6.9}$$

$$w_{ij} \in \{0, 1\} \qquad i \in I, j \in I \tag{6.10}$$

The objective of the model is the assignment of sections to exits in such a way that the sum over all used $g_{is}$ values is minimised. The sections that are combined to a sub-network have to be connected. Hence, the general idea of the model is to compute $|A|$ connected graphs with minimal weights over all these sub-networks.

Constraints (6.2) set the (dummy) exit sections $s \in A$ as the first sections of the sub-networks (predecessor for themselves $w_{ss} = 1$) and assign them to themselves as (dummy) exits. Accordingly, for these sections the binary variable $\zeta_{ss}$ is set to 1 in constraints (6.3). Constraints (6.4) ensure that every section has exactly one predecessor in a sub-network and therefore assure that each section can be assigned to exactly one exit section. Only the dummy section $W$ can be connected with more than one predecessors. In the example section 4 has to be connected with the dummy section $W$, because constraints (6.5) ensure that every section has more or an equal number of successors than predecessors. In the example section 4 is not connected with other sections than sections 3 and 5, and section 5 is already fixed as the predecessor of section 4. Constraints (6.6) determine that section $i$ can only be assigned to exit $s$ when section $j$ is assigned to exit $s$ and when $i$ and $j$ are connected in the sub-network. In Figure 6.2 section 6 can be assigned to exit 2 ($\zeta_{62} \geq 1$), because $w_{62} = 1$ and $\zeta_{32} = 1$ lead to a right hand side grater or equal to 1 in constraint (6.6). If one or both conditions are violated, the right hand side of (6.6) is greater than or equal to 0, in combination with the minimising objective function it would lead to $\zeta_{62} = 0$. Restrictions (6.7) ensure that every section $i$ is assigned to exactly one exit $s$. Constraints (6.8) avoid cycles between two sections. The domains of the decision variables are determined in (6.9) and (6.10).

## 6.3 A Heuristic to Compute Sub-Networks

With the optimisation model (6.2) - (6.10) it is not possible to compute optimal solutions for networks of practical size using commercial software. Therefore, to solve such problems the following heuristic is proposed. In the heuristic the sections are assigned iteratively to an exit with the best possible $g_{is}$ value. But a section $i$ can just be assigned to an exit $s$, if this section is connected with a section $j$ that is already assigned to exit $s$. Hence, the algorithm tries to assign a section $i$ to the exits with the lowest $g_{is}$ value for a given number of iterations. If the assignment is not successful after a predefined number of iterations the next best assignment (combination with the next lowest $g_{is}$ value) will be chosen. This procedure will be repeated until the section is successfully assigned to an exit. The set $Z$ includes all sections that are connected with sections which are already assigned to an exit. Parameter $k_i$ counts the number of iterations a section $i$ is in set $Z$. Parameter $r$ counts the number of sections that are taken from $Z$ per iteration. The thresholds for $k_i$ and $r$ are denoted by $K$ and $R$, respectively. For initialisation the sections $i$, which are directly connected with the super sink $S$, are defined as dummy exit sections. In the following the computation of sub-networks by means of a heuristic is described.

**Step 0** Define sections $i$ that are directly connected with the super sink $S$ as dummy exit sections: $i \in A : \beta_{i,S} = 1$.

**Step 1** Assign the sections $i$ that are directly connected with a (dummy) exit to this exit.

**Step 2** Set $k_i = k_i + 1$ for all $i \in Z$ and set $r = 0$.

**Step 3** Add those sections $i$ to $Z$ that are connected with assigned sections and that are not already contained in $Z$. Set the counter $k_i$ for these sections to 1.

**Step 4** If $Z = \{\}$ go to step 10.

**Step 5** Assign all $i \in Z$ that are connected with just one section to the exit the connected section is related to. Remove such sections from $Z$.

**Step 6** Randomly choose a section $i$ from $Z$ and set $r = r + 1$.

**Step 7** Determine the exit $s$ for the section $i$ where $g_{is}$ has the lowest value.

**Step 8** Check the following conditions for the chosen section $i$ and the exit $s$:

> **Step 8a** Section $i$ is connected with a section that is assigned to exit $s$: assign section $i$ to exit $s$ and remove section $i$ from $Z$.
>
> **Step 8b** Section $i$ is not connected with a section that is assigned to exit $s$: if $i$ is connected with unassigned sections and $k_i < K$ then keep section $i$ in $Z$.
>
> **Step 8c** Section $i$ is not connected with a section that is assigned to exit $s$: if $k_i = K$ or section $i$ is not connected with unassigned sections then set $g_{is} = \infty$ and $k_i = 0$.

**Step 9** When $Z = \{\}$ or $r = R$ go back to step 2, otherwise go back to step 6.

**Step 10** The algorithm terminates when all sections are assigned to an exit.

With the heuristic every section should be assigned to that exit where the $g_{is}$ value is minimal. All sections which are assigned to the same exit form a sub-network. Depending on the structure of the network and the assignment of the other sections the assignment to the exit with the actual lowest $g_{is}$ value may not be possible for every section. Hence, for $K$ iterations the algorithm tries to assign a section to the exit with the actual lowest $g_{is}$ value. If the assignment is not successful in these $K$ iterations the exit is forbidden for this section and the exit with the next lowest $g_{is}$ value is chosen. After these $K$ iterations the value $g_{is}$ for the chosen $i, s$ combination is fixed to infinity so that the exit does not have the lowest $g_{is}$ value any longer and another exit will be chosen in the next iteration. To ensure that $Z$ includes all sections that could be

assigned to an exit the sections in $Z$ should be updated after every assignment of a section. In this case the heuristic would start after each assignment with step 2, this leads to a high computational effort. To reduce this effort the sections in $Z$ are updated after $R$ assignments and for less iterations all steps must be executed. This is the compromise between updating $Z$ after every assignment versus updating $Z$ after all sections from $Z$ are assigned. The counter $r$ is used to keep hold on how many sections from $Z$ are chosen (per iteration) until the sections in $Z$ are updated. All sections that are just connected with one other section are assigned to the same exit as the connected section, because another assignment is not possible. Depending on the order the sections are chosen from $Z$ it could lead to different assignments from sections to exits and thereby to different sub-networks. To achieve a high variety in this order the sections are randomly chosen from Z.
The example network in Figure 6.1 is taken to illustrate the algorithm. To simplify the example parameters $g_{is}$ are set to the values in Table 6.3 and the thresholds R and K are set to 2.

| $g_{is}$ | 6 | 7 |
|---|---|---|
| 1 | 2 | 2 |
| 2 | 1 | 1 |
| 3 | 1 | 2 |
| 4 | 1 | 2 |
| 5 | 1 | 2 |

Table 6.3: Initialisation of Weight $g_{is}$.

**Step 0** The sections 6 and 7 are connected with the super sink $S$. So these sections are defined as dummy exit sections.

**Step 1** The sections 5 and 2 are connected with the exits 6 and 7. So these sections are assigned to these exits.

**Step 3** Add all sections that are directly connected with an assigned section (5, 2) to $Z$, $Z = \{1, 3, 4\}$ and set $k_i = 1$ for $i \in Z$.

**Step 5** Check which sections are connected with just one other section. In the example section 1 is just connected with section 2. Hence, section 1 will be assigned to the same exit as section 2 and will be removed of $Z$, $Z = \{3, 4\}$.

**Step 6** Randomly choose a section from $Z$, e.g. section 3 and set r = 1.

**Step 7** Determine the exit $s$ where $g_{3s}$ has the lowest value. For section 3 it is exit 6 (see Table 6.3, row 4).

**Step 8b** Section 3 is not connected with a section that is assigned to exit 6. Section 3 remains in Z because it is connected with unassigned sections and the threshold K is not violated $k_3 < 2$.

**Step 9** Go back to step 6, because |Z| > 0 and $r < 2$.

**Step 6** Randomly choose section 4 from $Z$ and set r = 2.

**Step 7** The best exit for this section is exit 6 (see Table 6.3, row 5).

**Step 8a** Section 4 is connected with section 5, which is assigned to exit 6. Hence, section 4 can be assigned to exit 6. Remove section 4 from $Z$, $Z = \{3\}$.

**Step 9** The threshold for r is violated. Go back to step 2.

**Step 2** Set $k_3 = 2$ and $r = 0$.

**Step 6** Choose section 3 from $Z$ and set $r = 1$.

**Step 7** The best exit for section 3 is exit 6 (see Table 6.3, row 4).

**Step 8a** Section 3 is connected with section 4, and this section is assigned to exit 6. Now section 3 can be assigned to exit 6. Remove section 3 from $Z$, $Z = \{\}$.

**Step 10** All sections are assigned to an exit. The algorithm terminates. From the assignment the sub-networks $\{1, 2, 7\}$ and $\{3, 4, 5, 6\}$ are obtained.

## 6.4 Implementation Variants of the Solution Procedure

With the heuristic described above one possible assignment from sections to exits is computed. To find a better assignment from sections to exits, in the following implementation variants solutions are combined. Afterwards some variants of the general implementation of the heuristic (Section 6.1) are presented: the S-(**S**tandard) implementation equals to the general approach described in Section 6.1 and the implementations SFCON, SFRED, and SFRAN are extensions of the standard implementation.

- **S-implementation**
  In the S-implementation the heuristic from Section 6.1 (step 1 - 5) is used to compute different sub-networks. The sub-networks (step 2) are computed with the mathematical model which was introduced in Section 6.2 or the heuristic which was presented in Section 6.3. In this variant, in addition to the stop criterion in step 5 (heuristic in Section 6.1), the heuristic stops after a predefined number of computed iterations. The solution with the lowest NCT is taken as the final solution.

- **Extensions of the S-implementation**
  The SFCON, SFRED, and SFRAN implementations are extensions of the S-implementation. In these variants solutions computed with the S-implementation are combined to find a further solution with a lower NCT. An iteration of these variants contains the following steps:

  Step 1: Computation of a certain number of sub-networks with the S-implementation.

  Step 2: Selection of two solutions with the lowest NCT.

  Step 3: Fixing the assignments from sections to exits which are equal in both solutions.

  Step 4: If all sections are assigned to an exit, stop. Otherwise, start at step 1 using the network with some fixed sections.

With this fixing a solution should be build step by step on the basis of assignments with good objective values. This approach should increase the possibility to find good solutions within a low number of computed iterations. The implementation variants differ in the way the number of solutions, computed with the S-implementation (step 1), is determined (in each iteration). All variants have in common that a part of the assignments from sections to exits is fixed (The letter **F**indicates it in the short cut) in each iteration. The differences of the variants are described below:

- **SFCON-implementation** : A **CON**stant number of solutions is computed with the S-implementation in each iteration.
- **SFRAN-implementation**: The number of solutions, which is computed with the S-implementation, is **RAN**domly chosen from a predefined interval in each iteration.
- **SFRED-implementation**: The number of solutions computed with the S- implementation is **RED**uced in each iteration. Hence, it starts in the first iteration with a high number of computed solutions. This number is reduced by a predefined number in each iteration until this number is reduced to two (the minimal number of solutions which is necessary for the combination of solutions) or all sections are assigned to an exit. This implementation variant starts with a large number of solutions at the beginning of the computation, where a lot of different assignments from sections to exits are possible. At the end of the computation, when the assignment is fixed for most sections and few assignments are possible for the remaining section, only a small number of further solutions are computed.

## 6.5 Computational Study

In the computational study the algorithm, which is described in Section 6.1, is tested with different combinations of parameters. The sub-networks are computed with the heuristic which is introduced in Section 6.3. The mathematical model which is presented in Section 6.2 is not used, because it is not appropriate for the network sizes that are considered in this thesis (see Section 5.2.1). In general all tests are performed with the standard implementation, with exception of the part where the different implementation variants are compared. First the number of computed iterations is modified and the influence on the NCT is investigated. Afterwards the heuristic for computing sub-networks is tested with different combinations of the parameters K and R and the influence on the NCT and the computation time is investigated. Then the NCTs computed with the implementation variants SFCON, SFRED, and SFRAN are compared to each other and to the S-implementation. Also the NCT computed with sub-networks, is compared to a solution with user-optimal flows and system-optimal flows. In a last step the number of exits in the networks is modified to investigate the influence of exits on the NCT. The computational study runs on the instances described in Section 5.2.1

*Number of computed iterations*

First it is tested which NCT can be achieved according to 10, 25, and 50 computed iterations. Also, the improvements that result from more computed iterations are examined. For each instance the algorithm runs for 50 iterations. In addition the number of iterations which can be used as a valid stop criterion in further experiments is determined in this first test. The parameters K and R are fixed to K = 2 and R = 5. This combination is used to test how many iterations are necessary to achieve a (good) improvement, when considering the worst case combination (an explanation for different K and R values is given in the next paragraph). Table 6.4 depicts the percentage of all instances, that have reached the best NCT computed in 50 iterations after 10 or 25 iterations, respectively. These percentages are presented for each network, evacuee group, in average for each network (row: average) and in average for each evacuee group over all networks (column: average). The results show, that the best NCT, which was computed in 50 iterations, was achieved for 43 % of all instances after 10 iterations and for 77 % of the instances after 25 iterations.

| Network | B | S | P | SY | M | A | L | NY | D | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| #-Iterations | | | | | 10 | | | | | |
| Evacuees | | | | | | | | | | |
| 500 | 90 % | 40 % | 20 % | 100 % | 70 % | 80 % | 50 % | 10 % | 30 % | 54 % |
| 700 | 80 % | 30 % | 10 % | 90 % | 20 % | 20 % | 50 % | 20 % | 40 % | 40 % |
| 800 | 50 % | 70 % | 10 % | 90 % | 50 % | 40 % | 30 % | 0 % | 10 % | 38 % |
| 900 | 50 % | 30 % | 10 % | 100 % | 50 % | 20 % | 50 % | 10 % | 30 % | 39 % |
| Average | 68 % | 43 % | 13 % | 95 % | 48 % | 43 % | 45 % | 10 % | 28 % | 43 % |
| #-Iterations | | | | | 25 | | | | | |
| Evacuees | | | | | | | | | | |
| 500 | 100 % | 70 % | 90 % | 100 % | 80 % | 100 % | 80 % | 100 % | 80 % | 89 % |
| 700 | 90 % | 70 % | 90 % | 100 % | 60 % | 80 % | 80 % | 50 % | 70 % | 77 % |
| 800 | 90 % | 80 % | 60 % | 100 % | 70 % | 80 % | 70 % | 50 % | 40 % | 70 % |
| 900 | 70 % | 70 % | 80 % | 100 % | 60 % | 90 % | 60 % | 40 % | 60 % | 70 % |
| Average | 88 % | 73 % | 80 % | 100 % | 68 % | 88 % | 73 % | 60 % | 63 % | 77 % |

Table 6.4: Percentage of Instances that have Reached the Best Results After 10 or 25 Iterations.

Additionally, the improvement of the NCT (in periods) after 10 or 25 instances is analysed. The results are presented in Table 6.5. The improvement (in periods) is presented in average for each evacuee group and each network, in average for each evacuee group and over all networks (row: average) and for each network over all evacuee groups (column: average). If the best NCT was not computed after 10 or 25 iterations, then the NCT could be reduced in average by 1.7 or 1.3 periods, respectively. Thus the improvement after 10 iterations was higher than after 25 computed iterations. In summary, the best ratio between the computed number of iterations and the resulting NCT could be achieved with 25 iterations. A doubling of iterations from 25 to 50 leads to small additional reductions of NCT for a few instances only. Hence, the stop criterion in the S-implantation is fixed to 25 iterations for the next tests.

| Network | B | S | P | SY | M | A | L | NY | D | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| #-Iterations | | | | | 10 | | | | | |
| Evacuees | | | | | | | | | | |
| 500 | 1.0 | 1.3 | 1.6 | 0.0 | 1.0 | 1.0 | 1.0 | 1.2 | 1.6 | 1.3 |
| 700 | 1.5 | 2.3 | 2.0 | 1.0 | 1.8 | 1.0 | 1.0 | 2.1 | 1.3 | 1.6 |
| 800 | 2.0 | 2.0 | 3.4 | 2.0 | 1.2 | 1.2 | 1.6 | 1.9 | 1.3 | 1.8 |
| 900 | 2.2 | 2.3 | 3.7 | 0.0 | 2.0 | 1.4 | 1.4 | 2.7 | 2.6 | 2.3 |
| Average | 1.7 | 2.0 | 2.7 | 1.5 | 1.5 | 1.1 | 1.2 | 2.0 | 1.7 | 1.7 |
| Network | B | S | P | SY | M | A | L | NY | D | Average |
| #-Iterations | | | | | 25 | | | | | |
| Evacuees | | | | | | | | | | |
| 500 | 0.0 | 1.0 | 2.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 1.2 |
| 700 | 2.0 | 1.0 | 1.0 | 0.0 | 1.5 | 1.0 | 1.0 | 1.3 | 1.0 | 1.2 |
| 800 | 2.0 | 1.0 | 1.8 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.2 |
| 900 | 1.7 | 1.0 | 4.5 | 0.0 | 1.5 | 1.0 | 1.5 | 1.5 | 1.0 | 1.7 |
| Average | 1.9 | 1.0 | 2.3 | 0.0 | 1.3 | 1.0 | 1.1 | 1.3 | 1.0 | 1.3 |

Table 6.5: Improvement of NCT in Periods After 10 or 25 Iterations.

*Different combinations of R and K parameters*

In a next test the influence of the parameters R and K (from the heuristic to compute sub-networks) on the NCT and on the computation time is analysed. These parameters determine the number of iterations for which a section can be assigned to the actual best exit (K) and the number of sections that are assigned to an exit in each iteration until the sections in set Z are updated (R). The hypothesis states that a high K leads to a solution with a low NCT, because many sections could be assigned to an exit with a low weight $g_{is}$. In addition, a low value for R leads to a solution with a low NCT because a frequent update of Z leads to a higher number of sections that can be considered for the assignment. It is hypothesised that a high value for K and a low value for R lead to better result in the NCT but to a higher computation time: either more iterations need to be computed or more reruns per iteration are required. To proof these hypotheses the heuristic is executed with four different combinations of values for parameters K and R. In the test the best value for K is set to 10 and the worst is set to 2. For R the best value is set to 1 and the worst is set to 5. By setting K and R to 10 and 2 or 1 and 5 respectively, ensures that the difference between the best and the worst value is big enough to monitor an effect on the heuristic.

| Combination | K | R |
|---|---|---|
| (1) | 10 | 1 |
| (2) | 2 | 5 |
| (3) | 10 | 5 |
| (4) | 2 | 1 |

Table 6.6: Parameters for the Four Combinations of K and R.

Table 6.6 presents the different combinations of K and R which were tested in the computational study. Combination (1) represents good values for both parameters, while the values of combination (2) represent bad values for both parameters. Combination (3) assigns a good value to parameter K and a bad one to R; in combination (4) it is the other way around. Figure 6.3 presents the average NCT of all networks for each set of evacuees (500, 700, 800, and 900). Each bar in a 4-tuple represents the average NCT for one of the K - R combinations from Table 6.6 (from left to right 1 - 4). The hypothesis that high values for K and low values for R (combination 1) lead to the lowest NCT (best result) and that low values for K and high values for R (combination 2) results in the highest NCT (worst result) can be confirmed, in respective of the evacuee group considered. This is not true for combinations (3) and (4) that represent medium-high values for K and R. Here the resulting NCT is influenced by the considered instances and depends on the size of the evacuee group. But Figure 6.3 additionally shows that all solutions are really close to each other. Hence, contrary to the initial hypothesis the values of K and R do not have a significant impact on the computed NCT.

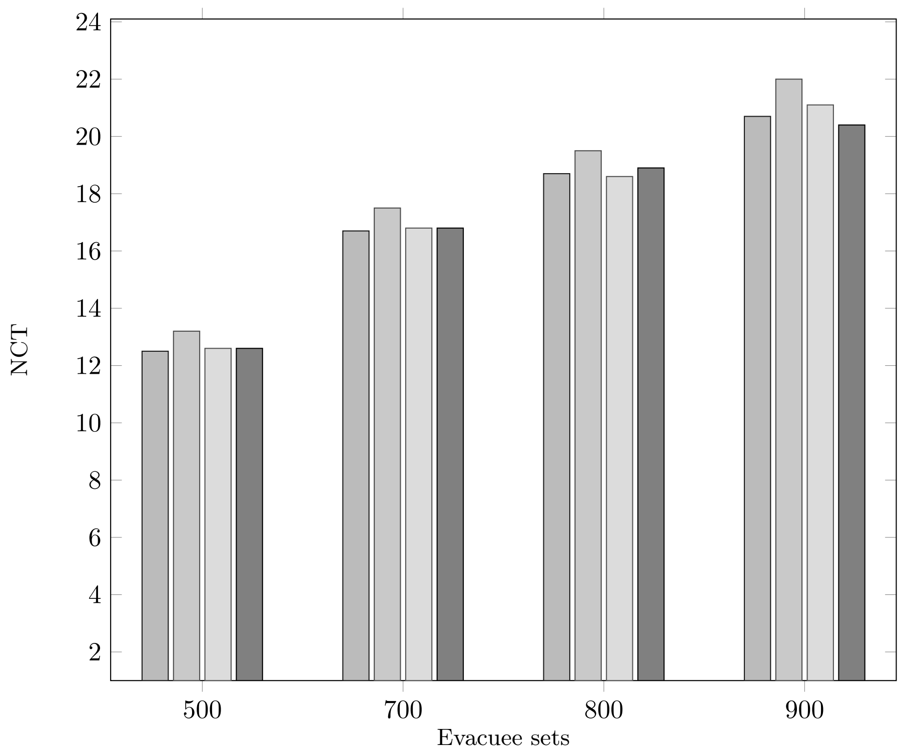



Figure 6.3: Average NCT per Evacuee Set for Combinations K and R.

The average computation time for each evacuee group are presented in Table 6.7. The results show that contrary to the hypothesis stated above combination (1) leads in average to the lowest computation time in each evacuee group. Also, the average (Avg.) computation time is the lowest with this combination. Hence, the best (lowest) NCT with the lowest computation time can be achieved with combination (1). According to these results, the parameters K and R will be fixed to $K = 10$ and $R = 1$ for the following tests.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 500 | **3.80** | 4.68 | 6.31 | 4.36 |
| 700 | **5.37** | 6.25 | 7.89 | 5.85 |
| 800 | **6.28** | 7.23 | 8.83 | 6.71 |
| 900 | **7.23** | 8.15 | 9.78 | 7.50 |
| Avg. | **5.67** | 6.58 | 8.20 | 6.11 |

Table 6.7: Average Computation Time (in CPU sec.) Resulting from Combinations 1 - 4 per Evacuee Group.

*Implementation variants*

In Section 6.4 different implementation variants of the basic heuristic (Section 6.1) were presented. Now it will be analysed whether improvements in NCT can be realised with these different implementation variants. Therefore, the best NCTs computed with the S-implementation after 25 iterations are compared with the NCTs of the variants. Furthermore, the NCTs of the variants are compared to each other. For the variants the following parameters are used: with the SFCON implementation 5 solutions are computed. In each iteration a constant number of 10, 8, 6, 4, and 2 solutions are combined. For the SFRAN implementation 4 solutions are computed and in each iteration a random number between 2 and 10 solutions are selected. For the SFRED implementation 5 solutions are computed. The number of solutions which are combined is reduced in each iteration by 2 (see the description in Section 6.4). The start values for the number of solutions which is combined are: 10, 8, 6, 4, 2.

| | SFCON | | | SFRED | | | SFRAN | | |
|---|---|---|---|---|---|---|---|---|---|
| | < | = | > | < | = | > | < | = | > |
| S | 39 % | 41 % | 20 % | 29 % | 46 % | 25 % | 43 % | 38 % | 19 % |
| SFCON | | | | 14 % | 53 % | 33 % | 33 % | 44 % | 23 % |
| SFRED | | | | | | | 41 % | 41 % | 18 % |

Table 6.8: Comparison of the NCT Between the Four Implementation Variants.

The results of the different implementation variants compared against each other and against the S-implementation are summarised in Table 6.8. The table depicts the direct comparison between two variants and it is shown for which percentage of all instances an NCT is reached that is better (<), equal (=) or worse (>) than in the other variant. For example the S-implementation leads in 39 % of all instances to a better NCT than the SFCON-implementation, in 41 % to an equal NCT and in 20 % to a worse NCT. The other way around the SFCON-implementation leads in 20 % of all instances to solutions with a better NCT than the S-implementation. The results show, that the S-implementation leads to the best results. The implementation variants SFCON and SRFRED lead to better results than the implementation variant SFRAN. In the test the variants SFCON and SFRED were run with a higher number of computed solutions in each iteration. So it can be stated, that the combination of more solutions leads to better results than the combination of a few solutions. In summary the S-implementation works best for most instances in the test whereas the variants do not lead to convincing results. Hence, for most instances the S-implementation should be preferred.

*User-optimal flows vs. flows in sub-networks*

So far the influence of different parameters on the heuristic has been tested and the implementation variants have been compared to each other. In the following, it will be analysed in which extent the NCT of the user-optimal flows can be influenced by subdividing the whole network into sub-networks. Therefore the NCTs of the user-optimal flows, the system-optimal flows and the best solutions computed with the method of sub-networks are compared. In Figure 6.4 to 6.7 the results are plotted as box plots. The NCTs of the system-optimal flows are fixed to 100 % and the NCTs of the user-optimal flows and the solution with blockages (sub-networks) are set in relation to these values. The box plots with the dashed lines represent the NCT which results from user-optimal flows and the box plots with continuous lines depict the NCT which results from the networks with blockages. Each box plot contains the NCT of one evacuee group for one network and presents the results of the different distributions of evacuees in the network. So each box plot depicts the spread of the NCT that results from the different distributions of evacuees in the networks. The results can be summarised as follows: the distribution of evacuees in the network has an influence on the NCT. This influence is higher for the solution with user-optimal flows as in the solution with sub-networks. Hence, the differences in the NCTs resulting from the distribution of evacuees in the network can be adjusted by computing sub-networks. Thus the NCT is only minimally influenced by the distribution of evacuees. Each Figure 6.4 to 6.7 presents the results for all networks for one evacuee group. The figures show that the introduction of blockages significantly reduces the NCT. The box plots show that the NCT resulting from the blocked networks leads to a lower % of system-optimal flows than the NCT resulting from user-optimal flows. A further observation is that in instances with a higher number of evacuees the distance between the user-optimal solutions and the solution with sub-networks is higher. This leads to the conclusion that a strict coordination of the traffic flow is more important for networks with a high number of evacuees than for networks with a low number of evacuees. Therefore, in case of an evacuation, with extraordinary traffic in the network, the use of sub-networks is an appropriate way to guide the traffic.

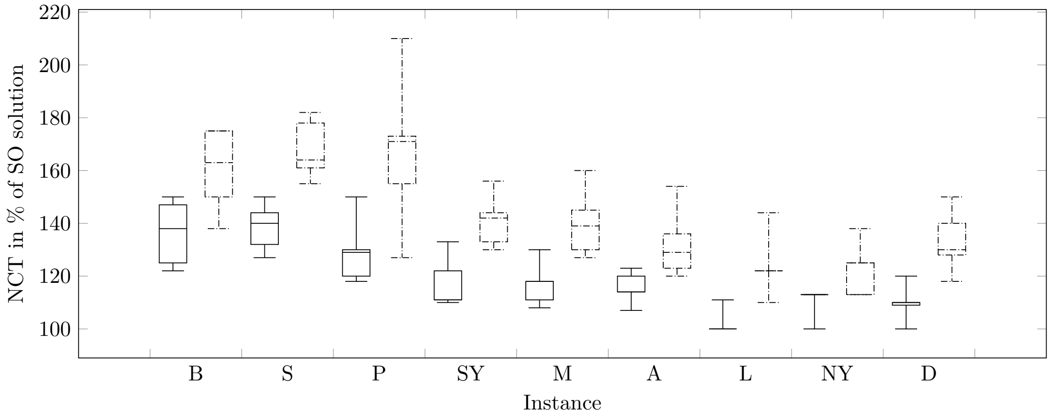



Figure 6.4: NCT for the User-Optimal Solution and the Solution with Sub-Networks in Relation to the System-Optimal Solution (SO); Instances with 500 Evacuees.

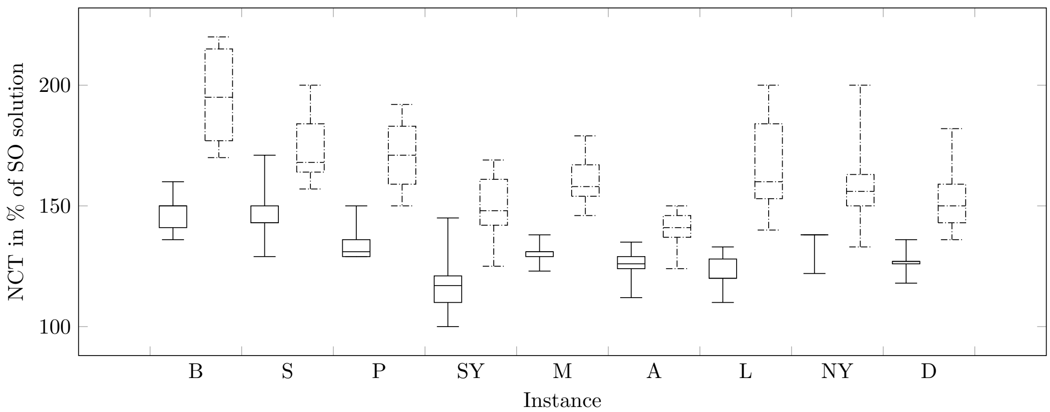



Figure 6.5: NCT for the User-Optimal Solution and the Solution with Sub-Networks in Relation to the System-Optimal Solution (SO); Instances with 700 Evacuees.

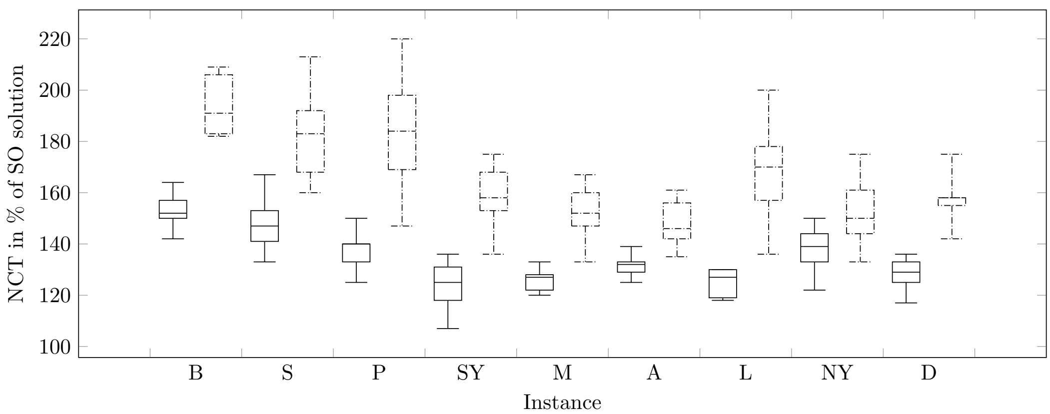



Figure 6.6: NCT for the User-Optimal Solution and the Solution with Sub-Networks in Relation to the System-Optimal Solution (SO); Instances with 800 Evacuees.

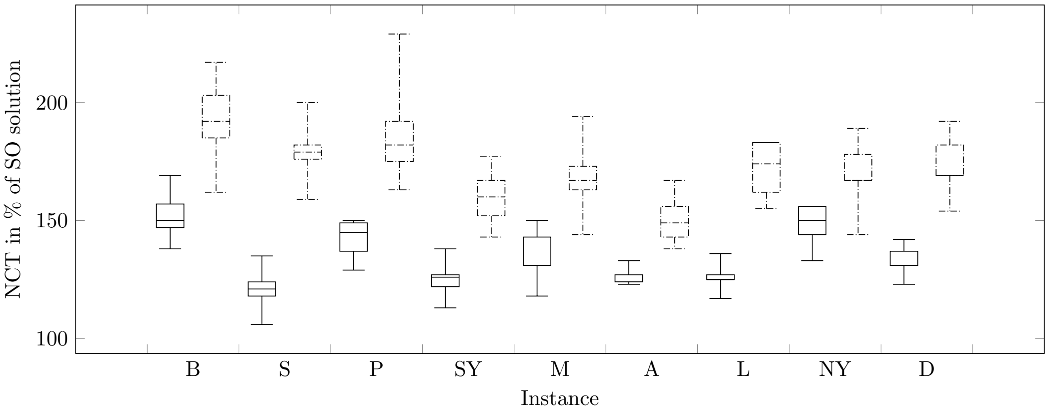



Figure 6.7: NCT for the User-Optimal Solution and the Solution with Sub-Networks in Relation to the System-Optimal Solution (SO); Instances with 900 Evacuees.

*Modification of exits in the networks*

Additionally, the influence of the number of exits on the NCT either generated by user-optimal flows or by the introduction of sub-networks is investigated. It is hypothesised that a reduction of exits leads to an increase of NCT, because the exits are the bottlenecks of the system. With less exits, less evacuees can leave the endangered zone per period. Moreover, it is analysed whether the number of exits has a larger impact on the NCT resulting from the user-optimal solution or on the NCT resulting from the introduction of sub-networks. In the test bed for each network (represented by abbreviations B - D) the number of exits is reduced to 75 % and 50 % of the original number of exits (the removed number of exits is always rounded down). The test is executed for each network with the instance *X_800_a*. The exits which are removed from the networks are chosen randomly and three different exit sets (a - c) for each network are tested. The removed exits for each instance are presented in Table 6.9. In the tests, where 50 % of the original number of exits is removed, the set of exits which is stated in the row 50 % and additionally the set of exits which is stated in the row 75 % is removed. The sum over the number of both sets yields to 50 % of the original number of exits.

| Network | | (a) | (b) | (c) |
|---|---|---|---|---|
| B | 75 % | $\{14, 27, 34, 58\}$ | $\{1, 16, 32, 53\}$ | $\{18, 27, 32, 53\}$ |
| | 50 % | $\{18, 26, 45\}$ | $\{21, 38, 41\}$ | $\{1, 45, 58\}$ |
| S | 75 % | $\{11, 44, 63\}$ | $\{50, 57, 74\}$ | $\{8, 55, 66\}$ |
| | 50 % | $\{8, 66\}$ | $\{53, 55\}$ | $\{11, 74\}$ |
| P | 75 % | $\{17, 40, 47\}$ | $\{18, 44, 56\}$ | $\{23, 56, 65\}$ |
| | 50 % | $\{6, 23, 65\}$ | $\{40, 65, 66\}$ | $\{17, 44, 47\}$ |
| SY | 75 % | $\{1, 18, 42, 51\}$ | $\{4, 6, 13, 14\}$ | $\{4, 13, 42, 46\}$ |
| | 50 % | $\{60, 64, 70\}$ | $\{46, 61, 79\}$ | $\{18, 51, 70\}$ |
| M | 75 % | $\{10, 22, 51\}$ | $\{17, 62, 52\}$ | $\{37, 52, 65\}$ |
| | 50 % | $\{20, 37, 60\}$ | $\{10, 65, 75\}$ | $\{20, 22, 26\}$ |
| A | 75 % | $\{16, 29, 35\}$ | $\{22, 40, 46\}$ | $\{29, 40, 98\}$ |
| | 50 % | $\{46, 93, 98\}$ | $\{29, 53, 79\}$ | $\{22, 79, 93\}$ |
| L | 75 % | $\{9, 20, 98, 106, 108\}$ | $\{32, 42, 54, 62, 71\}$ | $\{9, 42, 62, 101, 139\}$ |
| | 50 % | $\{93, 101, 116, 126, 127\}$ | $\{20, 133, 134, 136, 139\}$ | $\{32, 98, 106, 126, 133\}$ |
| NY | 75 % | $\{14, 31, 41, 68, 71\}$ | $\{37, 85, 97, 98, 99\}$ | $\{4, 31, 41, 111, 136\}$ |
| | 50 % | $\{4, 29, 45, 61, 69\}$ | $\{103, 111, 136, 140, 145\}$ | $\{29, 45, 85, 99, 103\}$ |
| D | 75 % | $\{7, 17, 22, 83\}$ | $\{14, 19, 87, 137\}$ | $\{19, 22, 45, 90\}$ |
| | 50 % | $\{38, 45, 46, 57\}$ | $\{7, 84, 90, 143\}$ | $\{46, 57, 84, 137\}$ |

Table 6.9: Sets of Removed Exits.

In Table 6.10 the NCTs of the networks with 75 % and 50 % remaining exits are compared to the NCTs of the original networks. The increase in the NCT resulting from the sub-networks are presented in the row denoted by "Sub-Net." and from the user-optimal flows in row "User". For each instance the percentage increase in NCT of the modified network compared to the unmodified network is presented. The results in Table 6.10 show that the NCT increases when the number of exits is reduced. For example in instance B_800_a, for the network with blockages the number of exits is reduced by maximally 25 % and the NCT is increased for (a) by 39 %, for (b) by 17 % and for (c) by 61 %. The results confirm the hypothesis, that less exits result in a higher NCT. In the instances (a) - (c) for each network different exits are removed. In each instance the remaining exits are at different positions in the network. The results demonstrate that in addition to the number of exits the position of exits has a significant influence on the NCT. For example in instance M_800_a, network divided into sub-networks: in instance (a) the reduction of the number of exits by 25 % results in an increase of the NCT by 79 %; in (c) the same number of removed exits leads to an increase of the NCT by only 32 %. In the solution resulting from user-optimal flows for instance B_800_a, the same NCT as in the original network with less exits was computed in instance (b). In average, in the solutions with sub-networks the NCT is increased by 44 % or 95 % and in the solutions resulting from user-optimal flows the NCT is increased by 36 % or 90 % for the networks with 75 % or 50 % remaining exit. Thus, in the solutions with user-optimal flows, the number of exits has a lower impact on the NCT compared to the solution with sub-networks.

In summary, the number of exits represents an important factor for the NCT. The tested instances show, that more exits lead to a faster evacuation in most cases. Additionally, the significant difference in the NCT resulting from the removal of different exits shows that the position of the exits also plays a crucial role for the NCT. Moreover, the method of computing sub-networks works better with more exits in the network. Thus, with more exits the original network can be divided into more sub-networks and the traffic can be routed more specifically.

| Instance | | | (a) | (b) | (c) |
|---|---|---|---|---|---|
| B_800_a | Sub-Net. | 75 | 39 % | 17 % | 61 % |
| | | 50 | 61 % | 56 % | 122 % |
| | User | 75 | 9 % | 0 % | 27 % |
| | | 50 | 91 % | 50 % | 109 % |
| S_800_a | Sub-Net. | 75 | 33 % | 29 % | 57 % |
| | | 50 | 124 % | 76 % | 67 % |
| | User | 75 | 31 % | 46 % | 42 % |
| | | 50 | 65 % | 100 % | 42 % |
| P_800_a | Sub-Net. | 75 | 33 % | 38 % | 19 % |
| | | 50 | 90 % | 100 % | 110 % |
| | User | 75 | 10 % | 3 % | 0 % |
| | | 50 | 24 % | 69 % | 107 % |
| SY_800_a | Sub-Net. | 75 | 27 % | 73 % | 73 % |
| | | 50 | 53 % | 87 % | 87 % |
| | User | 75 | 37 % | 53 % | 53 % |
| | | 50 | 68 % | 58 % | 89 % |
| M_800_a | Sub-Net. | 75 | 79 % | 47 % | 32 % |
| | | 50 | 126 % | 216 % | 100 % |
| | User | 75 | 109 % | 41 % | 50 % |
| | | 50 | 114 % | 159 % | 82 % |
| A_800_a | Sub-Net. | 75 | 40 % | 20 % | 24 % |
| | | 50 | 108 % | 80 % | 68 % |
| | User | 75 | 44 % | 7 % | 30 % |
| | | 50 | 111 % | 130 % | 70 % |
| L_800_a | Sub-Net. | 75 | 77 % | 46 % | 54 % |
| | | 50 | 100 % | - | 200 % |
| | User | 75 | 89 % | 22 % | 11 % |
| | | 50 | 117 % | - | 217 % |
| NY_800_a | Sub-Net. | 75 | 42 % | 33 % | 50 % |
| | | 50 | 75 % | 83 % | 58 % |
| | User | 75 | 36 % | 36 % | 36 % |
| | | 50 | 43 % | 86 % | 50 % |
| D_800_a | Sub-Net. | 75 | 67 % | 40 % | 33 % |
| | | 50 | 113 % | 153 % | 147 % |
| | User | 75 | 41 % | 53 % | 53 % |
| | | 50 | 153 % | 135 % | 194 % |

Table 6.10: Increase in NCT by Remaining 75 % and 50 % of Exits.

# Chapter 7

# Using Street Blockages to Guide the Selfish Evacuation Traffic

In Chapter 6 the idea of dividing a network into smaller sub-networks with one exit to regulate the traffic flow was presented. To obtain these sub-networks the connections between street sections that belong to different sub-networks are blocked. One drawback of forming such sub-networks is the lack of traffic regulation inside these networks. Thus only inaccurate traffic regulation is possible with this concept of sub-networks. Therefore, this concept is improved and specific connections between street sections are blocked to regulate the traffic flow more precisely. To determine the positions of blockages a bi-level model is formulated. First the general concept of bi-level optimisation is explained in Section 7.1. Afterwards a mathematical model to determine the position of blockages is presented in Section 7.2. In this bi-level formulation the routes of the evacuees are computed in the upper-level problem while the lower-level problem determines the blockages as a reaction on these routes. The model formulation based on the paper by Kimms and Seekircher (2016). In Section 7.3 three heuristic solution approaches are presented. Section 7.3.1 introduces an iterative solution approach, which is extended in Section 7.3.2 by a preprocessing step. Both approaches are taken from the paper by Kimms and Seekircher (2016). In Section 7.3.3 another heuristic is presented where further blockage combinations are tested. This section is based on the paper by Kimms and Seekircher (2017). The chapter closes with a comprehensive computational study.

## 7.1 Bi-Level Optimisation

Bi-level optimisation is an important field of research in mathematical programming (Talbi (2013)) and affects problems with two or more decision makers with individual objective functions that act and react in a non-cooperative sequential manner (Bard (2011)). The decisions of one party affect the decisions of the other one and vice versa. The concept was first introduced by Bracken and McGill (1973) as optimisation models with optimisation problems in the constraints. The name bi-level or multi-level optimisation was introduced by Candler and Norton (1977).
A bi-level optimisation model has a hierarchical structure of two levels, with an optimisation problem on each level: the upper-level and the lower-level problem (Talbi (2013)). The structure of the model leads to a comparison with a Stackelberg game (Stackelberg (1952)) in the game

theory, where a leader makes a decision and a follower decides on the basis of the leaders decision. A bi-level model is a static, non-cooperative version of this game (Bard (2011)). Hence, the upper- and the lower-level problems are often called leader and follower problems. In bi-level models each decision maker optimises his own objective function, without considering the objective functions of the other players, but his objective value and his decision space depend on their decisions (Talbi (2013)).

A formal description of the problem can be given with the following notation: the decision maker on the upper-level has his own objective function $F(x, y(x))$ and so does the decision maker on the lower-level $f(x, y(x))$. The variables $x \in X \subseteq \mathbb{R}^{n1}$ and $y(x) \in Y \subseteq \mathbb{R}^{n2}$ are the decision variables of the upper-level and lower-level problem, respectively. Where $y(x)$, points out that the decision on the lower-level depends on the decision (variables) of the upper-level problem. General formulations of bi-level models are presented for example by Colson et al. (2007) or Bard (2011). These formulations are summarised, to the following general formulation of a bi-level model

$$min_{x \in X} \;\; F(x, y(x)) \tag{7.1}$$
$$s.t. \;\; G(x, y(x)) \leq 0, \tag{7.2}$$

$$min_{y(x) \in Y} \;\; f(x, y(x)) \tag{7.3}$$
$$s.t. \;\; g(x, y(x)) \leq 0. \tag{7.4}$$

The upper-level problem optimises over the decision variables $x$ and indirectly on $y(x)$. In the lower-level problem decisions are only made via the $y(x)$ variables. The constraint sets of the upper- and lower-level problems are $G(x, y(x))$ and $g(x, y(x))$. In a static bi-level model each player has only one move. The upper-level model is solved first with the objective function and the constraint set of this problem. The lower-level problem is solved by considering the solution of the upper-level problem. Hence, in bi-level models the upper-level decision maker can influence the behaviour of the lower-level decision maker but cannot completely control him.

## 7.2 A Bi-Level Model for Emergency Evacuation Planning

As described above, connections between sections are blocked to regulate the selfish evacuation traffic in a network. When determining the position of blockages the behaviour of evacuees must be considered. The blockages have to be positioned in a way that they influence the route choice of the evacuees during an evacuation. To compute the position of these blockages a bi-level model is introduced. Bi-level optimisation is a common method for network design problems in traffic networks. In the related studies the network (re-)design decisions are made in the upper-level problem and the traffic assignments are computed in the lower-level as a reaction on the upper-level decisions (Abdelgawad and Abdulhai (2013)). In the bi-level formulation presented in this chapter the route choice decisions of the evacuees are made in the upper-level problem. In the lower-level problem a decision maker introduces network optimisations as a reaction to the previous route choices in order to regulate the evacuation traffic. In this thesis the network will be improved as reaction on the behaviour of the evacuees. Therefore it is modelled the other way around, compared with the usual practice: the routes which are computed in an upper-level problem are selfish routes that optimise the preferences of each network user without considering the behaviour of the other network users. In a lower-level problem the positions of blockages are determined taking these route choices into account.

*Assumptions and notation*

The bi-level model which is based on the cell transmission model and the traffic flow formulations (flow capacities, formulation of the affected area / safe zone) are taken from the formulation by Kimms and Maassen (2011b). To deal with selfishly routing evacuees and network blockages the flowing assumptions are made and additional notations are necessary.

- The set $B$ contains all evacuees $e$ which are in the affected area. The route choice of evacuees is defined in Section 5.1.1.
- Each connection $(i, j)$ between sections can be blocked separately. But if such a connection is blocked, the connection $(j, i)$ will be blocked as well.
- The blockages are determined with $m_{ij}$. In the lower-level problem $m_{ij}$ is defined as a binary variable. If a connection between two sections $i, j$ is blocked then is $m_{ij} = 0$, otherwise is $m_{ij} = 1$ and it has no influence on the network structure. In the upper-level problem $m_{ij}$ is used as a (binary) parameter and forms the network which was computed in the lower-level problem.
- The routes of the evacuees are represented with $r_{ije}$. In the upper-level problem $r_{ije}$ is a binary variable. If two sections $i$ and $j$ are in sequence on a route $r_{ije} = 1$, 0 otherwise. In the lower-level problem $r_{ije}$ is a binary parameter to consider the route choice of evacuees in the network design problem.
- The individual start position of each evacuee $e$ is given with $a_e$. The number of evacuees that starts the evacuation in a section $i \in I$ are $E_i = |\{e \in B | a_e = i\}|$.
- The binary variable $s_e$ indicates that a part of the route of evacuee $e$ computed in the upper-level problem is either blocked ($s_e = 1$) or not ($s_e = 0$) in the lower-level problem. Routes with blockages are determined in the lower-level problem according to the objective of the lower-level problem.
- The objectives of the upper-level problems are the individual objectives of the network users, who determine routes according to their preferences. The objective of the lower-level model is the minimisation of the NCT.

*Formulation of the upper-level problem*

The upper-level problem determines the route for every evacuee. As described in Section 5.1.1 the evacuees choose their routes without considering the behaviour of the other evacuees. It is assumed that the evacuees make their decisions on the basis of the network structure and their private preferences. Most of the evacuees will take a familiar way and they are not willing to take an unknown route. These familiar routes may be the shortest or fastest ways. Beside that also different types of routes can be integrated into the modelling approach, for example a way with maximum security. To compute these routes the upper-level problem for each evacuee $e \in B$ is the mathematical model (5.1) - (5.6) which was described in Section 5.1.1. The problem is formulated in an iterative bi-level optimisation framework (see Section 7.3). Therefore constraints (5.2) are substituted by constraints (7.5) to consider in this iterative framework the network improvements which are determined in the lower-level problem

$$r_{ije} \leq \beta_{ij} m_{ij} \qquad i \in I, j \in I. \tag{7.5}$$

As described above $m_{ij}$ is used as a parameter in the upper-level problem and indicates the blocked connections between sections, which are computed in the lower-level problem. The re-designed network is represented by multiplying $\beta_{ij}$ with $m_{ij}$. Connections which can be passed in the original network (represented by $\beta_{ij} = 1$) cannot be passed if these connections are blocked $m_{ij} = 0$.

*Formulation of the lower-level problem*

In this lower-level problem positions for street blockages will be determined to guide the selfish evacuation traffic. If a blocked connection is part of the selfish route of an evacuee then the route of this evacuee will be determined according to the objective of the lower-level problem, which minimises the number of periods where evacuees are in the network. The resulting number of periods is a period less than the NCT defined in this thesis. As stated above the blockages are determined by considering the selfish routes of evacuees. In this way, the blockades should lead to a traffic flow which corresponds to that of the system-optimal flow. No blockages will be necessary, if the selfish behaviour automatically leads to these flows. The blockages are used to force the evacuees to use alternative routes that result in more efficient traffic flow. Hence, blockages must be determined on routes where the alternative routes of the evacuees lead to an optimisation of traffic flow. But blockages that force a part of the evacuees to better alternative routes on the one side (from the view of the system-optimal flows) can, on the other side, prohibit good routes for other evacuees. An example is used to illustrate the general idea of blockage determination in the mathematical model. Further it is used to clarify the problem that is described above. Figure 7.1 depicts a network with five sections and three evacuees in the affected area. The user-optimal and system-optimal routes are presented on the right hand side of the figure. In this example the user-optimal routes all pass through connection $2-S$. To influence the routing of all evacuees, one possible blockage could be positioned between section 2 and the super sink $S$. In this case all evacuees have to take an alternative route. For evacuee A this blockage would result in the same route like in the system-optimal solution. But the use of the alternative routes for evacuees B, C would increase the NCT compared to the NCT by user-optimal flows. This blockage would prevent the use of the fastest routes for evacuees B, C. Alternatively the connection between sections 2 and 3 could be blocked. For evacuee A this blockage would lead to the same route as in system-optimal solution, but would not block the fastest routes for the evacuees B and C. Moreover, the blockage of the connection $2 - 3$ would result in system-optimal flows. If connections in the network are blocked, the alternative routes and the resulting traffic flow from these routes must be considered.

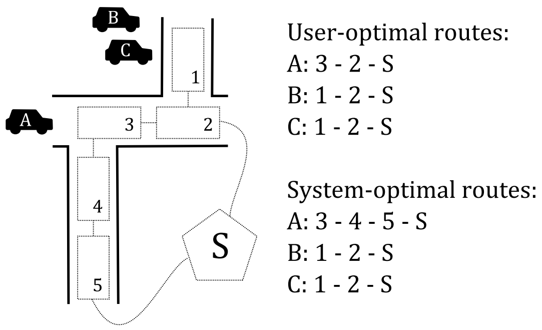



Figure 7.1: Network to Illustrate the Positioning of Blockages.

In the following a mathematical model is introduced to determine the best possible position of blockages to regulate the selfish evacuation traffic. To count the number of periods where evacuees are in the affected area the binary variable $\nu_t$ is used. If evacuees are in the network in period $t$ then is $\nu_t = 1$, and 0 otherwise. The mathematical model formulation can be stated as follows

$$\min \sum_{t \in T} \nu_t \tag{7.6}$$

s.t.

$$x_{it} = b_{it} + x_{it-1} + \sum_{j \in I} y_{jit-1} - \sum_{j \in I} y_{ijt} \qquad i \in I, t \in T \tag{7.7}$$

$$x_{it} + \sum_{j \in I} y_{ijt} \leq N_i \qquad i \in I, t \in T \tag{7.8}$$

$$\sum_{t \in T} b_{it} = E_i \qquad i \in I \tag{7.9}$$

$$\sum_{t \in T} y_{ijt} \geq \sum_{e \in B} r_{ije} \beta_{ij} m_{ij} - \sum_{e \in E} s_e r_{ije} \qquad i \in I, j \in I \tag{7.10}$$

$$s_e \sum_{i \in I} \sum_{j \in I} r_{ije} \geq \sum_{i \in I} \sum_{j \in I} (\beta_{ij} - m_{ij}) r_{ije} \qquad e \in B \tag{7.11}$$

$$s_e \leq \sum_{i \in I} \sum_{j \in I} (\beta_{ij} - m_{ij}) r_{ije} \qquad e \in B \tag{7.12}$$

$$m_{ij} = 0 \qquad (i, j) \in Fix^0 \tag{7.13}$$

$$m_{ij} = 1 \qquad (i, j) \in Fix^1 \tag{7.14}$$

$$y_{ijt} \leq m_{ij} \sum_{\tau \in I} E_\tau \qquad i \in I, j \in I, t \in T \tag{7.15}$$

$$x_{S|T|} = \sum_{i \in I} E_i \tag{7.16}$$

$$y_{ijt} \leq N_i \beta_{ij} \qquad i \in I, j \in I, t \in T \tag{7.17}$$

$$\sum_{j \in I} y_{jit} \leq N_i - x_{it} \qquad t \in T, i \in I \tag{7.18}$$

$$\sum_{j \in I} y_{jit} \leq Q_i \qquad i \in I, t \in T \tag{7.19}$$

$$\sum_{j \in I} y_{ijt} \leq Q_i \qquad i \in I, t \in T \qquad (7.20)$$

$$m_{ij} = m_{ji} \qquad i \in I, j \in I \qquad (7.21)$$

$$\nu_t \sum_{i \in I} E_i \geq \sum_{i \in I} E_i - x_{S,t} \qquad t \in T \qquad (7.22)$$

$$b_{it} \geq 0 \qquad i \in I, t \in T \qquad (7.23)$$

$$y_{ijt} \geq 0 \qquad i \in I, j \in I, t \in T \qquad (7.24)$$

$$x_{it} \geq 0 \qquad i \in I, t \in T \qquad (7.25)$$

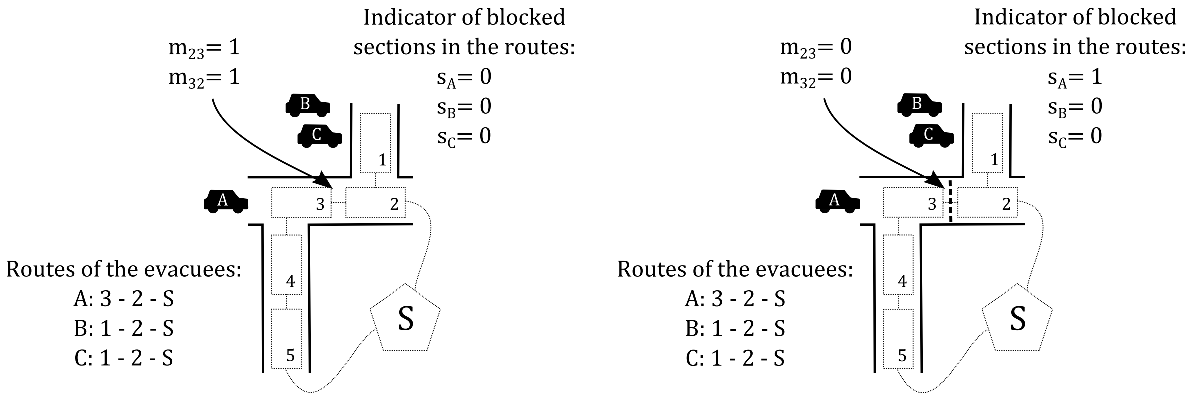
$$\nu_t \in \{0,1\} \qquad t \in T \qquad (7.26)$$

$$m_{ij} \in \{0,1\} \qquad i \in I, j \in I \qquad (7.27)$$

$$s_e \in \{0,1\} \qquad e \in B \qquad (7.28)$$

Objective function (7.6) minimises the number of periods in which evacuees are in the affected area. Constraints (7.7) - (7.8) determine the amount of traffic in the sections. Variable $x_{it}$ indicates the number of evacuees which is in a section at the end of a period. This number is determined with constraints (7.7) and includes all evacuees that start evacuation in a period $b_{it}$, which have remained in a section ($x_{it-1}$) or have reached a section ($\sum_{j \in I} y_{jit-1}$) in the previous period ($t-1$). The evacuees who left a section in period $t$ must be subtracted ($\sum_{j \in I} y_{ijt}$). With constraints (7.8) the adherence of section capacity is ensured. No more vehicles may be present in a section than the maximum capacity allows. Constraints (7.9) make sure that all evacuees start the evacuation within the planning horizon $T$.

Constraints (7.10) - (7.12) determine the blockage of connections between sections and the resulting traffic flow. If there are no blockages in the network the traffic will be routed along the selfish routes of evacuees. In this case, the last sum in constraints (7.10) will be 0. Therefore, the number of evacuees that flows from section $i$ to section $j$ must be equal to the number of evacuees that uses this connection in their route ($\sum_{e \in B} r_{ije}$). When blockages are positioned in the network this number must be reduced by the number of evacuees which uses a blocked connection in their selfish routes ($s_e = 1$). These evacuees have to take an alternative route. Constraints (7.11) in combination with (7.12) determine whether a part of the route of evacuee $e$ is blocked ($s_e = 1$) or not ($s_e = 0$). If connections are blocked alternative routes can be determined for all evacuees that have these connections in their routes. These alternative routes are determined according to the objective function of the lower-level problem. Therefore, in constraints (7.10) the number of evacuees that has to use a specific connection $i, j$ is reduced. The routes that lead the evacuees out of the affected area are then determined independently of their selfish routes. Figure 7.2 is used to illustrate the functionality of constraints (7.10) - (7.12). In the network depicted in Figure 7.2a no connection is blocked. According to the selfish routes of evacuees (stated on the left hand side of Figure 7.2a) two evacuees have to travel from section 1 to section 2, one from 3 to 2 and three from section 2 to $S$. For example constraint (7.10) with $i = 2, j = S$ is $\sum_{t \in T} y_{2St} \geq 3$. This number can be reduced by the number of evacuees that has a blocked connection on their route. For all evacuees $e$, which have a blocked connection on their route is the variable $s_e = 1$. In constraints (7.10), the number of evacuees, that has to use connection $i, j$ is reduced by all evacuees who have a blocked connection ($s_e = 1$) and the connection $i, j$ on their route ($r_{ije} = 1$). In the network in Figure 7.2b the connection between sections 2 and 3 is blocked. This connection is part of the route of evacuee A. The blocked connection is determined by $m_{23} = 0$ and $m_{32} = 0$. In constraints (7.11) the blockage leads to a right hand side greater than 0 for evacuee A. To satisfy this constraint the variable $s_A$ must equal 1 and the variable indicates that there is a blockage on the route of evacuee A.




(a) Network without Blockage.

(b) Network with Blockage.

Figure 7.2: Example to Clarify the Functionality of Constraints (7.10) - (7.12).

Constraints (7.12) ensure that variable $s_e = 0$, when no connection is blocked in a route of an evacuee $e$. For all $i, j$ which are part of the route of evacuee A in constraints (7.10) the right hand sides are reduced by one and an alternative route for evacuee A can be computed according to the objective function of the lower-level problem. By blocking the connections between sections the number of evacuees that has to use specific connections can be reduced, but these blockages also reduce the number of usable paths.

Later the upper- and lower-level problems will be solved iteratively. For this purpose additional notations are introduced. The set $Fix^0$ contains all pairs $(i, j)$ for which $m_{ij}$ has turned out to be zero in previous iterations. Initially the set $Fix^0$ is empty. In an improved version of the solution approach, a set $Fix^1$ will be determined which contains pairs $(i, j)$ for which $m_{ij}$ must equal one, see (7.13) and (7.14).

Constraints (7.15) ensure that there is no traffic flow between blocked connections. Constraint (7.16) assures that all evacuees have reached the super sink $S$ (safe area) at the end of the planning horizon $|T|$.

With (7.17) - (7.20) the traffic flow is restricted. Constraints (7.17) state that traffic flow is only possible between connected sections. Constraints (7.18) ensure that not more evacuees flow into a section than capacity is available in this section. Moreover, the traffic flow between sections is restricted by the flow capacity constraints (7.19) and (7.20).

It is assumed that a connection between two sections is blocked in both directions if it is blocked at all. This is regulated by constraints (7.21). Constraints (7.22) ensure that the variable $\nu_t$ takes the value 1 if evacuees are still in the network in period $t$. If all evacuees are in the super sink $S$, the right hand side of (7.22) will be 0, and $\nu_t$ will become 0, too. The domains of the decision variables are stated in (7.23) - (7.28).

## 7.3 Solution Procedures

### 7.3.1 A Basic Solution Procedure

An iterative optimisation framework is used to optimise the network taking into account the selfish routes of the evacuees. When the lower-level problem is solved and the routes from the recent upper-level problem have been considered, it cannot be guaranteed that all necessary connections are blocked that would force the evacuees to use the alternative routes that minimise the overall NCT. If in the lower-level problem connections between sections are blocked then those evacuees that use routes that contain these connections, will be routed according to the objective function of the lower-level problem. These routes can vary from the alternative selfish routes the evacuees would take in the re-designed network. Hence, the selfish routes of the evacuees must be computed again for the network with blockages. If the selfish routes of the evacuees and the alternative routes which are determined in the lower-level problem are identical, then no additional blockages are necessary. If this is not the case, additional blockages are required to force the evacuees to use the alternative routes. The iterative solution procedure stops, when no additional blockages are computed in the lower-level problem. Otherwise, the algorithm starts again with computing new routes in the lower-level problem. The iterative procedure can be summarised by the following steps:

**Initialisation:** Let $m_{ij} = 1$ for all $i, j \in I$. $Fix^0 = \{\}$.

**Step 1:** Take the current values $m_{ij}$ (blockage of connections) as parameters, and solve the upper-level problem. As a result get new values for $r_{ije}$ (routes of the evacuees).

**Step 2:** Take the current values $r_{ije}$ (routes of the evacuees) as parameters and solve the lower-level problem. Result: further blockages $m_{ij} = 0$, update $Fix^0 = \{(i,j)|m_{ij} = 0\}$ with these values.

**Step 3:** If there are no changes in the variable $m_{ij}$ then stop. Else start with step 1.

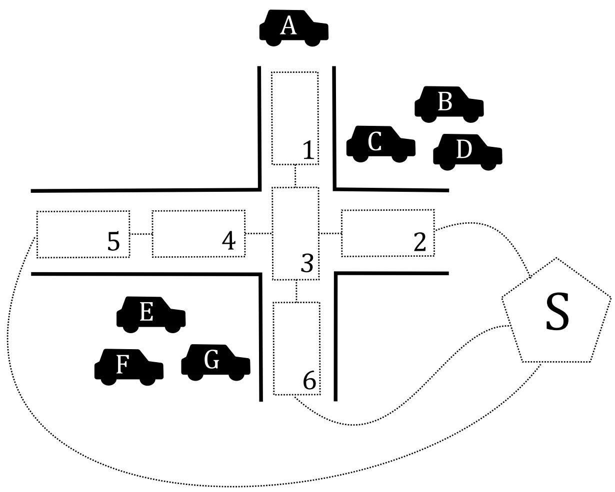



Figure 7.3: Network to Illustrate the Determination of Blockages with the Iterative Solution Approach.

A numerical example is given to explain the iterative procedure: Figure 7.3 depicts a network with sections 1 - 6 in the affected area and the safe area super sink $S$. It is assumed that there are seven evacuees in the network, where 3 evacuees start in section 2 and 6, and 1 evacuee starts in section 1. The algorithm begins in the first iteration with a network without blockages, so $m_{ij} = 1$ for all $i, j \in I$ and $Fix^0 = \{\}$ .

- In step 1 the selfish routes with the upper-level problem are computed. Under the assumption that all evacuees prefer to take the fastest route (i.e. with the minimum number of sections), the following routes are computed: the evacuees B, C, and D take the route 2 - S, the evacuees E, F, and G the route 6-S and evacuee A takes the route 1 - 3 - 2 - S.
- In the second step by considering these routes the lower-level problem is solved. Without any blockage the evacuees have to take their selfish routes and the NCT would be 5 periods. By introducing a blockage between section 2 and section 3, the route of evacuee A can be modified. The alternative route for evacuee A is 1 - 3 - 4 - 5 - $S$. That leads to a NCT of 4 periods. Hence, it is set $m_{23} = 0$ and $m_{32} = 0$, $Fix^0 = \{(2, 3), (3, 2)\}$.
- From step 3 go back to step 1, due to the change in variable $m_{ij}$.
- In step 1 the routes of the evacuees for the new network are computed. There are no changes in the routes for the evacuees B, C, D, E, F, and G. The new (fastest) route for A is 1 - 3 - 6 - S.
- In step 2 the blockage from the first iteration is fixed for variables $m_{23} = 0$, $m_{32} = 0$ and a blockage of the connection between section 3 and 6 is determined to force evacuee A to use the alternative route 1 - 3 - 4 - 5 - $S$. It is fixed $m_{36} = 0$, $m_{63} = 0$ and set $Fix^0 = \{(2, 3), (3, 2), (3, 6), (6, 3)\}$ is updated.
- From step 3 go back to step 1 again, due to changes in values of $m_{ij}$.
- In step 1 the routes for the evacuees B, C, D, E, F, and G are the same as in the previous iteration. The new (fastest) route for A is 1 - 3 - 4 - 5 - $S$. In step 2 these routes do not lead to additional blockages and the algorithm stops in step 3.

In the network blockages between sections 2 and 3 and also between 3 and 6 are determined. With these blockages the traffic flows according to the system-optimal solution.

### 7.3.2 A Preprocessing Procedure

Each network contains connections whose blockage prevents that a part of the evacuees can leave the network. Hence, the blockage of these connection is not possible and in order to reduce the computational effort these connections are identified in a preprocessing step. The variables $m_{ij}$ are fixed to 1 for these connections before the lower-level problem is solved. As already defined $Fix^1$ represents the set of connections that must not be blocked. There are two cases that prevent the blockage of connections: sections have just one connection to another section or have just one connection to a path that leads to an exit. These properties are used to identify all connections in the network that must not be blocked. The network modifications are considered with parameter $\beta_{ij}^0$ in the iterative solution procedure

$$\beta_{ij}^0 = \begin{cases} 0, & \text{for (i,j)} \in Fix^0 \\ \beta_{ij}, & \text{otherwise} \end{cases} . \tag{7.29}$$

After each network modification the preprocessing step must be executed again in the iterative solution approach. If it is determined that a connection cannot be blocked then the decision applies to all subsequent iterations. Hence, in $\beta_{ij}^0$ all decisions are stored, so fewer connections must be tested in the following iterations.
To identify connections that must not be blocked for each section the number of connections to other sections is counted. Therefore, the row sum $\omega_i$ of $\beta_{ij}^0$ is computed. It is obvious that each section that has only one connection to another section, this connection cannot be blocked. Thus the algorithm starts with the sections where $\omega_i = 1$ and adds these sections to $\Delta$, the set of sections where the blockage of connections is not possible. Then a section $i \in \Delta$ is chosen and deleted from the set $\Delta$, and $m_{ij} = 1$ and $m_{ji} = 1$ are fixed for the existing connection ($\beta_{ij} = \beta_{ji} = 1$), $Fix^1 = \{(i,j),(j,i)\}$. The value $\omega_j$ is reduced by one for section $j$ that is connected with section $i$. The connection to section $i$ does not lead to an exit. Hence, section $j$ has $\omega_j - 1$ connections that could lead to an exit. If section $j$ has only one remaining connection to another section, then section $j$ will be added to the set $\Delta$, because the connection cannot be blocked. If there is more than one connection, then it is possible to block some of these connections, and section $j$ is not added to the set $\Delta$. The number of connections that could lead to an exit is iteratively reduced. In this way all connections that must not be blocked are identified. The procedure is repeated for all sections in $\Delta$ until the set is empty. The preprocessing procedure can be summarised with the following steps:

**Initialisation**: $\Delta = \{\}$ and $Fix^1 = \{\}$; Compute the row sum $\omega_i$ of $\beta_{ij}^0$ for all $i \in I$. If $\omega_i = 1$ then add section $i$ to set the $\Delta$.
**Repeat:**

- Select a section $i$ from $\Delta$ and eliminate this section from set $\Delta$.
- For $\beta_{ij}^0 = 1$: let $Fix^1 = Fix^1 \cup \{(i,j),(j,i)\}$ .
- Let $\omega_j := \omega_j - 1$.
- If $\omega_j = 1$ then add $j$ to $\Delta$.

**until** $\Delta = \emptyset$.

The network in Figure 7.3 is used to explain the algorithm by means of an example. The parameters $\beta_{ij}^0$, represent the connections between sections in the network (see Table 7.1):

| $i$ \ $j$ | 1 | 2 | 3 | 4 | 5 | 6 | S | $\omega_i$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 2 |
| 3 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 4 |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 |
| 5 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 |
| 6 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 2 |
| S | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 3 |

Table 7.1: Parameter $\beta^0_{ij}$ and the Row Sum for Every Section $i$.

First the row sum $\omega_i$ of $\beta^0_{ij}$ is computed for every section $i \in I$ (last column of Table 7.1). Section 1 is added to the set $\Delta$, because this section is connected with only one other section, thus $\omega_1 = 1$. For all other sections is $\omega_i > 1$. Then section $i = 1$ is taken from the set $\Delta$ and $j = 3$ is chosen because of $\beta_{13} = 1$. Parameter $\omega_3$ is reduced by 1 and it is $\omega_3 = 4 - 1 = 3$. There is more than one connection that can lead to an exit, thus section 3 is not added to the set $\Delta$. The algorithm stops, because the set $\Delta$ is empty. A look at Figure 7.3 makes clear, that the connection between section 1 and 3 is the only connection whose blockage would prevent that all evacuees can leave the dangerous area. The set $Fix^1$ which results from the preprocessing procedure is $Fix^1 = \{(1,3),(3,1)\}$.

The algorithm from Section 7.3.1 extended by the preprocessing has the following steps:

**Initialisation:** Let $m_{ij} = 1$ for all $i,j \in I$, $Fix^0 = \{\}$ and $Fix^1 = \{\}$.

**Step 1:** Take the current values $m_{ij}$ (blockage of connections) as parameters, and solve the upper-level problem.

**Step 2:** Execute the preprocessing procedure to determine $Fix^1$.

**Step 3:** Take the current values $r_{ije}$ (routes of the evacuees) as parameters, solve the lower-level problem and update $Fix^0 = \{(i,j)|m_{ij} = 0\}$.

**Step 4:** If there are no changes in the variable $m_{ij}$ then stop, else start with step 1.

### 7.3.3 A Blockage-Combination Heuristic

In Section 7.3.1 an iterative solution approach was used to solve the bi-level problem. Blockages that guide the selfishly routing evacuees are determined to optimise the NCT. The blockages which are necessary to guide the traffic determine the network structure. Thus different blockage combinations will lead to different solutions for the network design problem and result in different NCTs. Therefore, a particular solution of the network design problem is a specific combination of blockages. To compute such a blockage combination the upper- and the lower-level problems (Section 7.2) are solved iteratively until either the traffic can be guided with the blockages according to the objective of the lower-level problem or no further blockages can be introduced, due to the network structure.

In each iteration the optimal solution of the lower-level problem is computed with the routes of the evacuees (optimal solutions of the upper-level problems). But the optimal solution of the lower-level problem per iteration does not automatically lead to the optimal blockage combination for the total network. In fact, computing further blockages by means of the iterative approach is detrimental, because the iterations do not consider the global view of the optimisation problem. The blockages are computed on the basis of the routes computed in the previous iteration without considering the resulting selfish route choice in the following iterations of the solution approach.

*General idea of the heuristic*

To deal with this drawback a heuristic that computes additional blockage combinations on the basis of the blockages determined in the optimal solution of the lower-level problem is introduced. In the following a specific blockage combination is denoted by $g$. In each iteration of the solution approach for each blockage combination $g$ the upper- and the lower-level problems are solved once. As described above the upper- and the lower-level problems have to be solved alternately until all blockages for the network design problem are determined. A blockage combination is complete, when no additional blockages can be determined in the lower-level problem. To distinguish between blockage combinations where all necessary blockages are computed and not all blockages are computed, they are designated as *complete* and *incomplete* blockage combinations. So it is the general idea of the heuristic to compute different combination of blockages to identify the combination that leads to the lowest $NCT$.

*Elimination of non-promising combinations*

In order to prevent that all possible combinations of blockages are tested, blockage combinations with a non-promising $NCT$ are eliminated in the algorithm. Therefore, the reference value $NCT^*$ is introduced. $NCT^*$ is the $NCT$ of the best complete blockage-combination that is already determined. Beginning the heuristic $NCT^*$ must be initialised. For the initialisation two possibilities are proposed:

- $NCT^1_{init}$ is set to the best solution that is computed with the iterative procedure described in Section 7.3.1. In this case the blockage-combination heuristic is used to improve the best solution which is computed with the iterative approach.
- $NCT^2_{init}$ is set to the user-optimal solution described in Section 5.2.2. Then the heuristic is used to find a solution with blockages.

In the best case both initialisations result in the same blockage combination, with a minimal $NCT$. If $NCT^*$ is initialised with $NCT^2_{init}$, in most cases more iterations have to be computed compared to the initialisation with $NCT^1_{init}$. The value $NCT^1_{init}$ is closer to the minimal $NCT$ than $NCT^2_{init}$. When $NCT^1_{init}$ is used for initialisation the heuristic is a post-optimisation process, when $NCT2_{init}$ is used the heuristic is an independent solution approach.
Incomplete blockage combinations whose $NCT$ is not better than $NCT^*$ can be deleted, because the resulting complete blockage combination will mainly result in the same but not in a lower $NCT$. With each iteration that is computed for $g$ the NCT becomes worse or remains the same, because the solution space is reduced with each determined blockage.

In the following the before mentioned statements will be illustrated by an example. In the network in Figure 7.3 (without any blockages) the selfish route of evacuee A is 1 - 3 - 2 - $S$. In a system-optimal solution evacuee A has to take route 1 - 3 - 4 - 5 - $S$. When one connection in the route of evacuee A is blocked, in the lower-level problem for A the best possible alternative route according to the objective of the lower-level problem can be computed. This route leads to the lowest possible $NCT$ for the given network. With any additionally determined blockage the number of possible routes is reduced, which either results in a worse or the same $NCT$ than in the iteration before.

*Building new blockage combinations*

A blockage combination $g$ can be defined with sets $Fix_g^0$ (connections between sections that are blocked) and $Fix_g^1$ (connections that are not allowed to be blocked). These sets exist for each blockage combination $g$ and substituted the sets $Fix^0$ and $Fix^1$ in constraints (7.13) and (7.14) in the iterative procedure of this heuristic. The bi-level model will be separately solved for each blockage combination $g$ within this heuristic.
As described above in each iteration for each blockage combination $g$ the upper- and lower-level problems are solved once. After each run, the solution of the lower-level problem provides additional blockages for those pairs $i, j$ where $m_{ij} = 0$. These blockages are denoted by $AFix_g^0 = \{(i,j) \in I | (i,j) \notin Fix_g^0 \wedge i > j \wedge m_{ij} = 0\}$. For each blocked connection $(i,j) \in AFix_g^0$ a new blockage combination $g'$ is derived. Every new blockage combination $g'$ is derived from a current $g$ in such a way that only one blocked connection at a time is additionally forbidden to be blocked in a new solution. Therefore, $g$ and $g'$ differs in a status of one particular connection between $(i,j)$. So $g'$ can be defined with $Fix_{g'}^0 = Fix_g^0 \setminus \{(i,j),(j,i)\}$ and $Fix_{g'}^1 = Fix_g^1 \cup \{(i,j),(j,i)\}$.
For each newly defined blockage combination $g'$ the $NCT$ from $g$ (i.e. $NCT_g$) is used as initialisation for the $NCT$ of $g'$ (i.e. $NCT_{g'}$). Solution $g'$ cannot get a better $NCT$ than the incomplete blockage combination $g$, because the solution space is more restricted in $g'$. By initialising $NCT_{g'}$ with $NCT_g$, blockage combination $g'$ can be deleted if a complete solution with a better $NCT$ is found and if for blockage combination $g'$ no iteration has been executed yet.

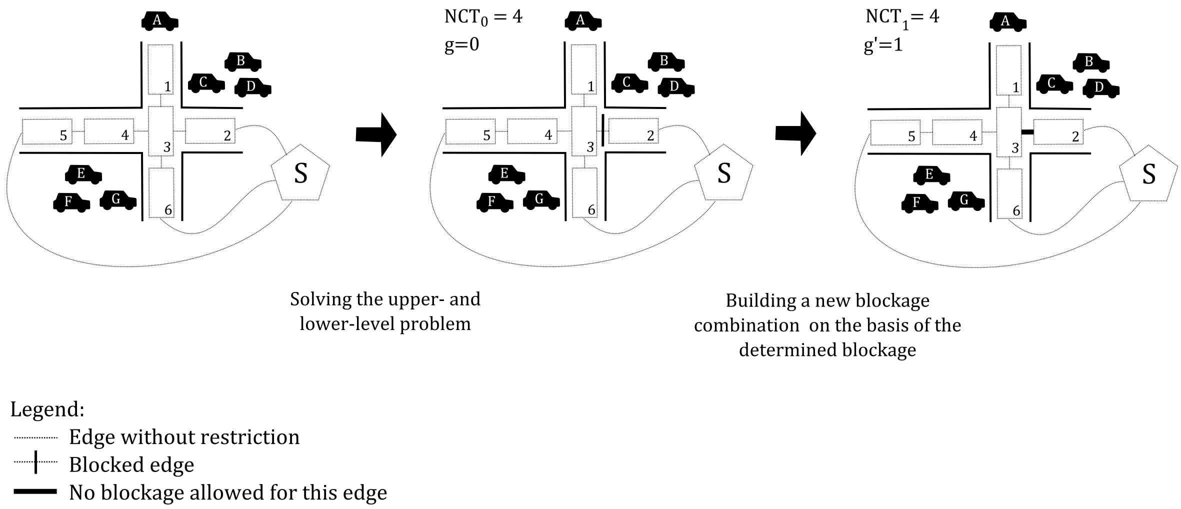



Figure 7.4: Network to Illustrate the Building of a New Blockage Combination.

Figure 7.4 illustrates the formation of a new blockage combination. When the upper- and lower-level problems (Figure 7.4, first network) are solved once this is assumed to result in one blockage only which is located between sections 2 and 3, $AFix_g^0 = \{(2,3)\}$ (Figure 7.4, second network). The blockage combination $g = 0$ can be expressed with the sets $Fix_0^0 = \{(2,3),(3,2)\}$ and $Fix_0^1 = \{\}$, with $NCT_0 = 4$. On the basis of the solution computed for $g = 0$ one further blockage combination $g' = 1$ is build. Which is $g' = 1$ with $Fix_1^0 = \{\}$ and $Fix_1^1 = \{(2,3),(3,2)\}$ and the $NCT$ is set to $NCT_1 = NCT_0 = 4$ (Figure 7.4, third network). For this blockage combination it is forbidden to block the connection between section 2 and 3 (bold line).

*Heuristic procedure*

In each iteration of the heuristic the upper- and lower-level problems are solved only once for several blockage combinations. The incomplete blockage combinations are stored in the lists $F$ and $F^{'}$. Complete combinations are just saved, if the resulting $NCT$ is better than $NCT^*$. The list $F$ contains all incomplete blockage combinations that already exist at the beginning of an iteration. List $F^{'}$ contains all incomplete blockage combinations that are newly defined by an iteration. Both lists are necessary, because the upper- and lower-level problems are only solved for those blockage combinations that already exist at the beginning of an iteration. In each iteration for several blockage combinations in list $F$ the upper- and lower-level problems are solved and further blockage combinations are determined until a stop criterion is reached. List $F$ is initialised with a blockage combination $g = 0$ and $F =< 0 >$. The blockage combination $g = 0$ represents the network without any fixed blockages. Initially list $F'$ is defined empty (i.e. $F' =<>$), because new solutions will be computed in the first iteration and thus do not exist at this point of time. After each iteration the new defined solutions in list $F'$ are added to list $F$ and are deleted from $F'$.

*Stop criterion*

It is expected that the blockage combinations which are in the front part of list $F$ lead to the best results. Hence, it is desirable to compute fast a complete solution for these combinations. Moreover, it is useful to compute a complete blockage combination at the very beginning which then updates $NCT^*$ in order to eliminate non-promising incomplete combinations. Therefore, in each iteration of the algorithm the upper- and lower-level problems are solved once again only for a part of the blockage combination in list $F$.
The number of blockage combinations which is taken from $F$ is defined in each iteration and depends on the criterion $|F'| \geq \frac{|F|}{b}$. Let $|F|$ and $|F'|$ be the number of items in lists $F$ and $F'$, respectively. Hence, the number of blockage combinations considered in each iteration depends on the existing incomplete blockage combinations in list $F$ and on the new defined combinations in list $F'$. Thus in each iteration just a fraction of the already existing combinations can be newly defined. This fraction is determined with parameter $b$. With $b \geq 1$ fewer or equivalent new blockage combinations than the already existing combinations are needed to stop an iteration; with $b < 1$ more combinations than the existing ones are required. Parameter $b$ will be determined according to the respective network. The parameter must be set to a number grater than zero. With this criterion the balance between quickly computing complete blockage combinations for a small part of the items in list $F$ and between computing a lot of different combinations, is considered. At the beginning multiple new blockages are determined for each $g$ leading to a significant increase in new blockage combinations in list $F^{'}$. By doing so the

condition $|F^{'}| \geq |F|$ is rapidly achieved, and the heuristic starts again with the first item in list $F$. If $g$ tends to a complete blockage combination, a lower number of additional blockages are computed and less combinations are added to $F^{'}$. In subsequent iterations the number of elements in list $F$ would be higher than in preceding iterations and therefore additional blockage combinations can be added to $F^{'}$ until it reaches the criterion $|F^{'}| \geq |F|$ and for a greater part of solutions the upper- and lower-level problem is solved. Additionally an iteration stops, if all solutions are chosen from list $F$.

*Example*

Figure 7.5 illustrates the second iteration of the blockage-combination heuristic. The $NCT^{*}$ is initialised with the user-optimal solution $NCT^{2}_{init} = 5$ and $b = 0.4$. The lists are $F =< 0, 1 >$ and $F^{'} =<>$.

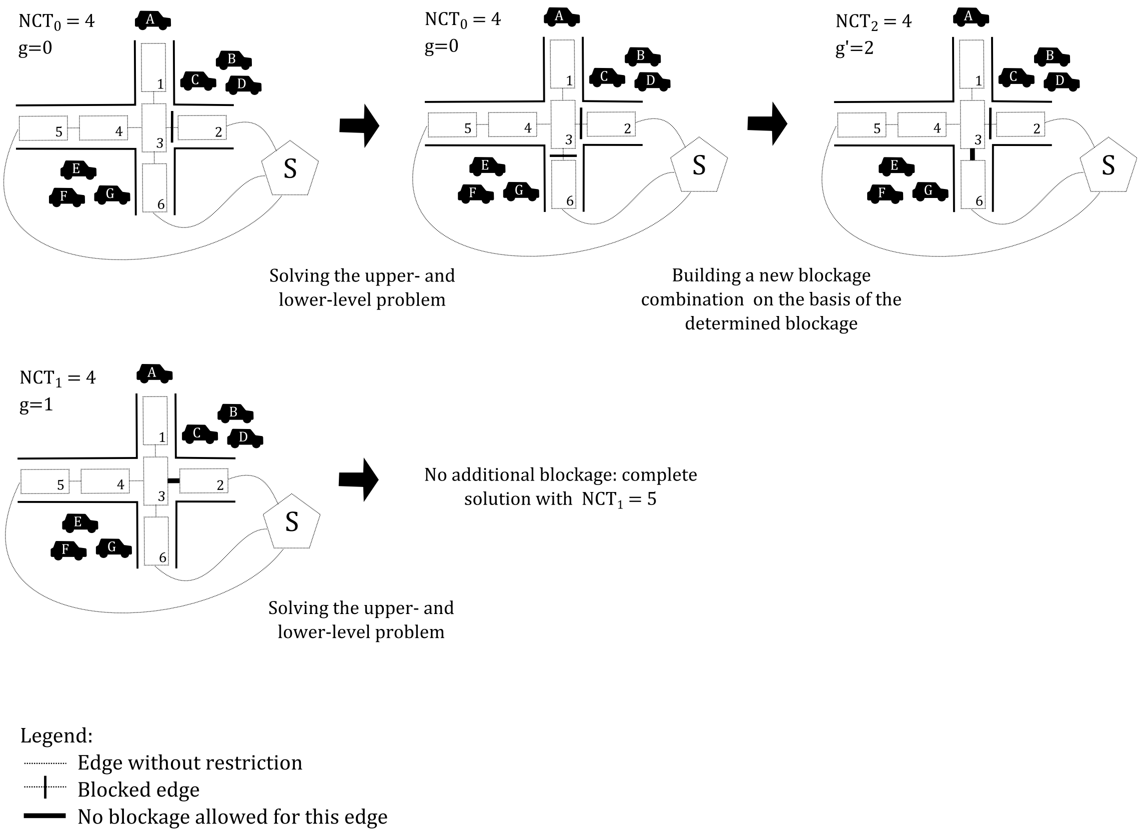



Figure 7.5: Example for the Second Iteration of the Blockage-Combination Heuristic.

The algorithm starts with the first blockage combination in list $F$ $g = 0$ and solves the upper- and the lower-level problems once. It leads to one additional blockage between sections 3 and 6 ($AFix^{0}_{0} = \{(3, 6)\}$) and $NCT_0 = 4$ (Figure 7.5 second network in the first row). It is not a complete blockage combination (because of the additional blockage) and $NCT_0 < NCT^{*}$, thus $g = 0$ remains in list $F$. The sets are $Fix^{0}_{0} = \{(2, 3), (3, 2), (3, 6), (6, 3)\}$ and $Fix^{1}_{0} = \{\}$. For each new pair $(i, j)$ in $AFix^{0}_{0}$ a new blockage combination is derived, so one combination $g' = 2$ is built with $Fix^{0}_{2} = \{(2, 3), (3, 2)\}$ and $Fix^{1}_{2} = \{(3, 6), (6, 3)\}$ and is added to list $F' =< 2 >$

(Figure 7.5, third network in the first row). The stop criterion is not reached $|F| = 2 \geq \frac{|F'|=1}{0.4}$. The next $g$ from list $F$ is taken ($g = 1$) and the upper- and the lower-level problems are solved once for $g = 1$ (Figure 7.5 first network in the second row). For this combination no additional blockages are computed: it is not allowed to block the connection between section 2 and 3 and the blockage of any other connection does not affect the route of evacuee A (1 - 3 - 2 - $S$) or does lead to an infeasible / worse solution. This blockage combination results in $NCT_1 = 5$. It is a complete combination with $NCT_1 \geq NCT^*$, thus $g = 1$ is deleted from list $F =< 0 >$.
At the end of each iteration the new computed solutions from list $F'$ are transferred to the end of list $F$. The lists are $F =< 0, 2 >$ and $F' =<>$. The algorithm is repeated until list $F$ is empty.

With this heuristic a lot of different blockage combinations are tested to identify one that leads to the lowest $NCT$. The pseudocode for the blockage combination heuristic is presented in Algorithm 1.

**Algorithm 1** Blockage-Combination Heuristic

**Initialisation**: $NCT^*, F =< 0 >, F' =<>, Fix_0^0 = \{\}, Fix_0^1 = \{\}$.
**repeat**
    Take the next $g$ from list $F$,
    solve the upper- and the lower-level problem once for $g$ and
    update $NCT_g$.
    **repeat**
        **if** $NCT_g \geq NCT^*$ **then** delete $g$ out of list $F$
        **else if** $g$ is a complete blockage combination **then**
            update $NCT^* = NCT_g$ and
            delete all $g$ with $NCT_g \geq NCT^*$ out of list $F$.
        **else**
            **for all** $(i,j) \in I | (i,j) \notin Fix_g^0 \wedge i > j \wedge m_{ij} = 0$ **do**
                let $Fix_g^0 = Fix_g^0 \cup \{(i,j),(j,i)\}$
                let $AFix_g^0 = AFix_g^0 \cup \{(i,j)\}$
            **end for**
            **for all** $\{(i,j) \in AFix_g^0\}$ **do**
                Define $g'$:
                    let $Fix_{g'}^0 = Fix_g^0 \setminus \{(i,j),(j,i)\}$
                    let $Fix_{g'}^1 = Fix_g^1 \cup \{(i,j),(j,i)\}$
                    let $NCT_{g'} = NCT_g$
                let $AFix_g^0 = AFix_g^0 \setminus \{(i,j)\}$
                Add $g'$ at the end of list $F'$
            **end for**
        **end if**
    **until** $|F| \geq \frac{|F'|}{b}$ or all items from list F are considered
    Add all $g'$ of list $F'$ at the end of list $F$.
    Delete all $g'$ out of list $F'$.
**until** list $F =<>$

## 7.4 Computational Study

In this section several aspects of the presented solution approaches are investigated. As test bed the instances presented in Section 5.2 are used. First, the solution approaches in Section 7.3.1 and Section 7.3.2 are compared to each other regarding the computation time and the NCT. The algorithm in Section 7.3.2 is an extension of the algorithm in Section 7.3.1, where connections that cannot be blocked are identified by means of a preprocessing procedure and the decision variables are fixed for these connections. It is expected that the reduced number of decisions reduces the computation time. Afterwards, an additional test bed is used to investigate at which evacuation demand the blockage of sections is useful. Then, the solutions computed by means of the iterative approach are improved by applying the blockage-combination heuristic. Finally, the NCTs either computed with the sub-network method (Section 6) or with the specific blockage method (Section 7) are compared to each other.

*Solution approach with and without preprocessing*

In a first test all instances are solved with the basic iterative solution approach (Section 7.3.1) and with the preprocessing approach (Section 7.3.2). In this test the NCT, the computation time that is necessary to solve the lower-level problem (solver time), the time that is used to run all the other processes (run-time for iterations) and the sum of both times (which is the necessary time to run the complete approach) are compared. Both approaches lead to solutions with the same NCT, thus the results are not discussed here. The computation times are compared to each other: the absolute (in CPU seconds) and percentage deviation of the results computed with and without preprocessing are presented in Table 7.2. The average values of each set of evacuees for each network are presented. The columns with $PT$ (absolute (abs) and relative (%)) present the process time, the columns with $ST$ (absolute (abs) and relative (%)) the solve time and in the columns with $SUM$ (absolute (abs) and relative (%)) the total computation time of the algorithms. A positive deviation states that the computation time in the approach with preprocessing is less than the computation time in the approach without preprocessing. For example in instance B_500_* the solve time of the approach with preprocessing is in average 3.83 CPU seconds (abs.) less than in the approach without preprocessing, which is a reduction of 12 %. In summary in approx. 90 % of all instances the computation time (PT, ST, and SUM) could be reduced by preprocessing and in average a reduction of approx. 10 % of the computation time could be achieved. For the instances with small- and medium-sized networks the preprocessing approach always reduces the computation time. Just for instances with large-sized networks the preprocessing approach leads to an increased computation time. Here, the additional time that is required to run the preprocessing approach is higher then the reduction in solve time by fixing variables. For example this is the case for the instances L_900_* or D_700_*.

| | NCT | PT_abs | PT_% | ST_abs | ST_% | SUM_abs | SUM_% |
|---|---|---|---|---|---|---|---|
| B_500_* | 10.60 | 3.83 | 12% | 9.75 | 13 % | 13.58 | 13 % |
| B_700_* | 16.70 | 5.83 | 12 % | 17.62 | 12 % | 23.45 | 13 % |
| B_800_* | 21.50 | 7.20 | 11 % | 28.99 | 12 % | 36.19 | 12 % |
| B_900_* | 22.10 | 8.32 | 11 % | 35.11 | 12 % | 43.43 | 12 % |
| S_500_* | 12.00 | 4.90 | 14 % | 7.88 | 13 % | 12.77 | 13 % |
| S_700_* | 16.30 | 8.10 | 17 % | 12.28 | 13 % | 20.38 | 14 % |
| S_800_* | 22.80 | 9.60 | 11 % | 30.71 | 12 % | 40.31 | 12 % |
| S_900_* | 22.20 | 13.64 | 16 % | 23.47 | 12 % | 37.11 | 13 % |
| P_500_* | 12.20 | 6.23 | 16 % | 11.38 | 13 % | 17.61 | 14 % |
| P_700_* | 16.10 | 12.05 | 17 % | 16.38 | 12 % | 28.44 | 14 % |
| P_800_* | 20.00 | 19.89 | 19 % | 25.54 | 12 % | 45.43 | 14 % |
| P_900_* | 21.20 | 17.86 | 14 % | 33.69 | 12 % | 51.55 | 13 % |
| SY_500_* | 10.00 | 6.27 | 20 % | 3.97 | 13 % | 10.24 | 16 % |
| SY_700_* | 13.20 | 8.08 | 18 % | 5.13 | 13 % | 13.21 | 16 % |
| SY_800_* | 16.40 | 19.44 | 26 % | 13.10 | 12 % | 32.54 | 18 % |
| SY_900_* | 18.90 | 15.15 | 15 % | 13.56 | 12 % | 28.71 | 14 % |
| M_500_* | 12.50 | 13.20 | 16 % | 13.11 | 13 % | 26.31 | 14 % |
| M_700_* | 15.80 | 20.40 | 12 % | 31.47 | 12 % | 51.87 | 12 % |
| M_800_* | 20.30 | 31.41 | 12 % | 59.08 | 12 % | 90.49 | 12 % |
| M_900_* | 23.30 | 15.58 | 5 % | 56.63 | 12 % | 72.21 | 9 % |
| A_500_* | 14.30 | 4.67 | 5 % | 6.66 | 13 % | 11.32 | 8 % |
| A_700_* | 17.80 | 8.65 | 4 % | 24.39 | 12 % | 33.04 | 7 % |
| A_800_* | 21.90 | 15.35 | 4 % | 37.62 | 13 % | 52.97 | 8 % |
| A_900_* | 24.50 | 14.61 | 4 % | 68.75 | 13 % | 83.36 | 9 % |
| L_500_* | 9.20 | 12.04 | 6 % | 0.41 | 0 % | 12.44 | 5 % |
| L_700_* | 11.20 | 2.01 | -3 % | 0.17 | 0 % | 2.18 | -1 % |
| L_800_* | 13.50 | -0.98 | -1 % | 2.28 | 1 % | 1.31 | 0 % |
| L_900_* | 14.10 | -8.64 | -5 % | 1.38 | 0 % | -7.27 | -2 % |
| NY_500_* | 8.10 | 13.94 | 4 % | 17.29 | 12 % | 31.23 | 6 % |
| NY_700_* | 10.20 | 31.30 | 4 % | 42.02 | 13 % | 73.32 | 7 % |
| NY_800_* | 10.90 | 30.13 | 8 % | 56.04 | 13 % | 86.17 | 10 % |
| NY_900_* | 12.00 | 41.23 | 8 % | 86.46 | 13 % | 127.69 | 11 % |
| D_500_* | 10.80 | 6.59 | 1 % | 0.20 | 0 % | 6.79 | 1 % |
| D_700_* | 12.30 | -3.78 | -3 % | 2.89 | 1 % | -0.89 | -1 % |
| D_800_* | 14.10 | 13.97 | 3 % | 1.22 | 0 % | 15.19 | 2 % |
| D_900_* | 15.50 | -0.77 | -1 % | 3.63 | 0 % | 2.87 | 0 % |

Table 7.2: Summary and Comparison of the Results Computed with the Iterative Solution Approaches with and without Preprocessing.

*Modification of demand levels*

The next step analyses if the blockage of street sections always reduces the NCT. Pas and Principio (1997) present some criteria that induce the Braess paradox. Only if Braess's paradox occurs in the network, it will be useful to guide the traffic by blocking connections between street sections. One criterion that was mentioned by Pas and Principio (1997) is the demand level of network users. If this level is significantly low or high Braess's paradox does not occur in the network. Hence, in the computational study different demand levels are tested to investigate at which level the blockage of street sections does not lead to a reduction of the $NCT$. In this thesis the demand level is defined as the proportion between the network capacity and the number of evacuees in the network. The network capacity is set to the total free-flow capacity $Cap_{free} = \sum_{i \in I} Q_i$. The demand level ($Net_{Dem}$) is defined as follows

$$Net_{Dem} = \frac{|B|}{\sum_{i \in I} Q_i}. \tag{7.30}$$

For the computational study all networks presented in Section 5.2.1 are used and four demand levels $Net_{Dem} \in \{0.5; 1; 2; 3\}$ are defined. The number of evacuees for the different demand levels in each network is summarised in Table 7.3.

| Network | $Cap_{free}$ | $Net_{Dem} = 0.5$ | $Net_{Dem} = 1$ | $Net_{Dem} = 2$ | $Net_{Dem} = 3$ |
|---|---|---|---|---|---|
| Berlin (B) | 402 | 201 | 402 | 804 | 1206 |
| Stockholm (S) | 450 | 225 | 450 | 900 | 1350 |
| Paris (P) | 468 | 234 | 468 | 936 | 1404 |
| Sydney (SY) | 504 | 252 | 504 | 1008 | 1512 |
| Melbourne (M) | 594 | 297 | 594 | 1188 | 1782 |
| Auckland (A) | 714 | 357 | 714 | 1428 | 2142 |
| Lima (L) | 834 | 417 | 834 | 1668 | 2502 |
| New York (NY) | 906 | 453 | 906 | 1812 | 2718 |
| Dubai (D) | 942 | 471 | 942 | 1884 | 2826 |

Table 7.3: Number of Evacuees for the Four Demand Levels and Different Networks.

For the instances in Table 7.3 the NCTs of the solution with blockages ($NCT_{block}$) are compared to the NCTs of the user-optimal solution ($NCT_{user}$). In Figure 7.6 the ratio between $NCT_{user}$ and $NCT_{block}$ for the instances given in Table 7.3 are plotted. A ratio equal to 1 says that both solutions have the same NCT, a ratio lower than 1 indicates that the NCT in the solution with blockages is less than the NCT of the user-optimal solution and a ratio greater than 1 indicates that the NCT in the user-optimal solution is lower than the NCT in the solution with blockages. Due to the high computation time, only a limited set of instances could be tested. Especially those instances with a high demand level result in high computation times, e.g. for the network D with $Net_{Dem} = 3$ the computation time constituted more than 48 hours. Hence, it is difficult to state whether different demand levels generally impact the suitability of blockages or not. But the comparison between the results at demand levels 0.5 / 1 and 2 / 3 tents to the assumption that the induction of blockages does not significantly decrease the NCT when the demand levels are high. In most cases the induction of blockages results in an increased NCT when the demand level equals 3.

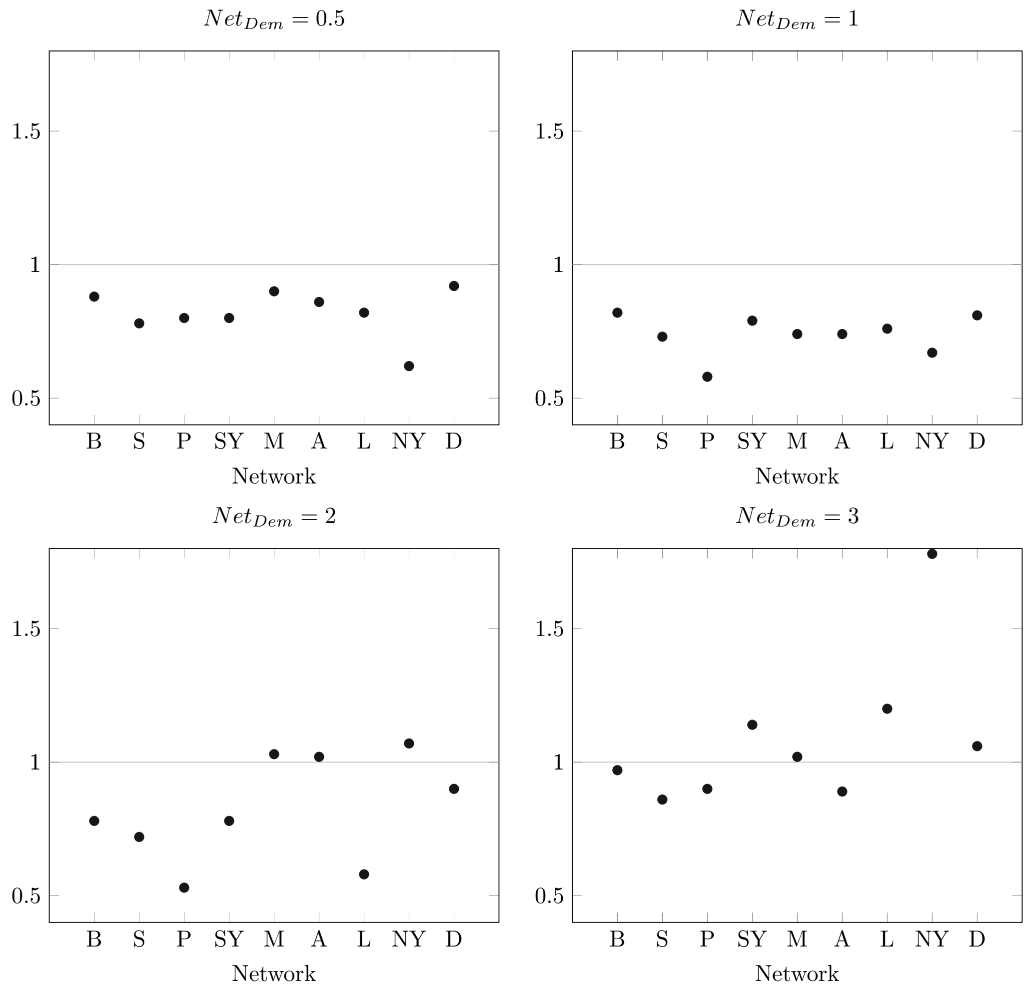



Figure 7.6: Computed Ratios between $NCT_{user}$ and $NCT_{block}$.

With regard to the tested instances, a medium demand level (1 or 2) will lead to the best results. Moreover, on average the induction of blockages works better for the instances with small-sized networks (B, S, P, SY) than for the instances with large-sized networks (M, A, L, NY, D). In summary the results correspond to the findings by Pas and Principio (1997) that the Braess paradox does not occur if the demand level is significantly low or high.

*Blockage-combination heuristic*

The blockage-combination heuristic, which is introduced in Section 7.3.3, is operated with all test instances. The $NCT^*$ is initialised with $NCT^1_{init}$, thus the heuristic is used to improve the solutions computed with the basic approach. Parameter $b = 5$ is fixed in the stop criterion as previous studies already showed that this value is best suited. As an additional stop criterion a maximum computation time of 1800 sec. is used. If the maximum computation time is reached in the test, the best solution computed within these 1800 sec. is chosen as a result. The results are summarised in Table 7.4:

| | B | S | P | SY | M | A | L | NY | D |
|---|---|---|---|---|---|---|---|---|---|
| 500 | | | | | | | | | |
| # | 7 | 5 | 5 | 4 | 6 | 2 | 0 | 0 | 0 |
| Average | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Min | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Max | 4 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| 700 | | | | | | | | | |
| # | 8 | 8 | 6 | 4 | 4 | 3 | 0 | 0 | 1 |
| Average | 3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Min | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Max | 10 | 3 | 6 | 1 | 2 | 2 | 0 | 0 | 1 |
| 800 | | | | | | | | | |
| # | 8 | 10 | 7 | 5 | 5 | 1 | 0 | 0 | 0 |
| Average | 4 | 5 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| Min | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Max | 11 | 9 | 8 | 9 | 10 | 2 | 0 | 0 | 0 |
| 900 | | | | | | | | | |
| # | 7 | 8 | 6 | 4 | 5 | 3 | 0 | 0 | 0 |
| Average | 3 | 2 | 2 | 2 | 2 | 1 | 0 | 0 | 0 |
| Min | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Max | 7 | 6 | 5 | 7 | 4 | 3 | 0 | 0 | 0 |
| #-Network | 30 | 31 | 24 | 17 | 20 | 9 | 0 | 0 | 1 |

Table 7.4: Improvements in the NCT with the Blockage-Combination Heuristic.

Table 7.4 depicts the number of improvements (#), the average improvement (Average) in periods as well as the minimum (Min) and maximum (Max) improvement in periods for each network and each set of evacuees. Moreover, the number of improvements per network (#-Network) is presented. The results show, that considerable improvements could be achieved in those instances with small- and medium-sized networks (B - A). In 131 out of 240 instances improvements could be observed. In the instances with large-sized networks (L - D) only one instance could be improved. For most of these instances the heuristic stopped, because the maximum computation time was exceeded. On average only small improvements of 1 or 2 periods could be achieved. But each data set contains instances where an improvement of more than 5 periods could be achieved. An superior improvement of more than 10 periods was achieved for the instances of network B with 700 and 800 evacuees.

*Sub-networks vs. specific blockages*

In the last part of the computational study the results computed with sub-networks (Chapter 6) and the best solutions computed with the specific blockages are compared. The NCTs of all instances resulting from both methods (sub-networks and specific blockage computing) are compared to the NCTs resulting from the system-optimal flows. The box plots in Figures 7.7 - 7.10 represent the NCTs resulting from the solution approach with sub-networks and specific blockages in relation to the NCTs resulting from system-optimal flows. The box plots with dotted lines represent the NCTs of the approach with specific blockages and the box plots with continuous lines depict the NCTs of the approach with sub-networks.

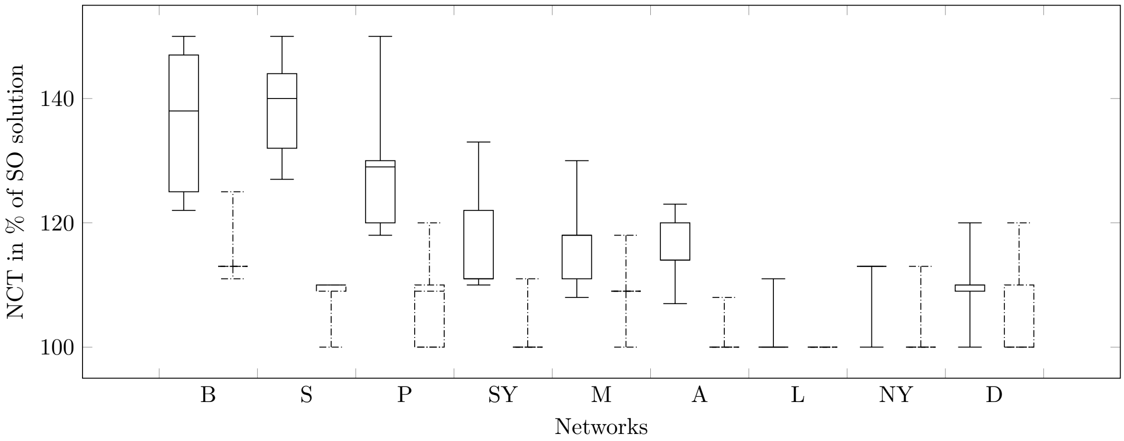



Figure 7.7: NCT of the Method with Sub-Networks and with Specific Blockages in Relation to the System-Optimal Solution (SO); Instances with 500 Evacuees.

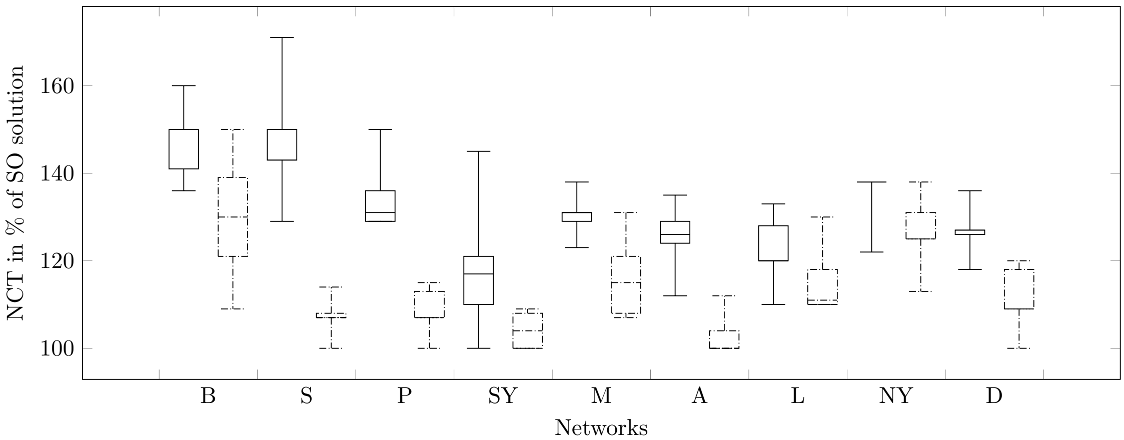



Figure 7.8: NCT of the Method with Sub-Networks and with Specific Blockages in Relation to the System-Optimal Solution (SO); Instances with 700 Evacuees.

Figure 7.7 depicts the results for all instances with 500 evacuees. For the instances with small- and medium-sized networks (B - A) the specific blockage approach leads to better results than the introduction of sub-networks. In case of the instances with large-sized networks (L - D) there is no significant difference between both approaches.
A similar picture can be seen with the instances of 700 evacuees in Figure 7.8. When considering 700 evacuees a significant difference between both solution approaches can be observed with the small- and medium-sized networks, here the approach of specific blockages works best. In contrast to that the results of both approaches are similar for the instances with large-sized networks as already seen with 500 evacuees.

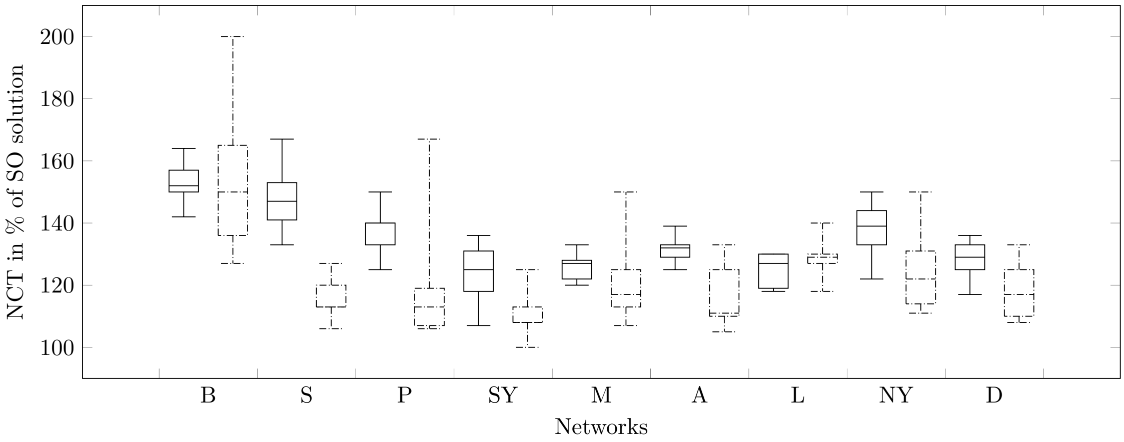



Figure 7.9: NCT of the Method with Sub-Networks and with Specific Blockages in Relation to the System-Optimal Solution (SO); Instances with 800 Evacuees.

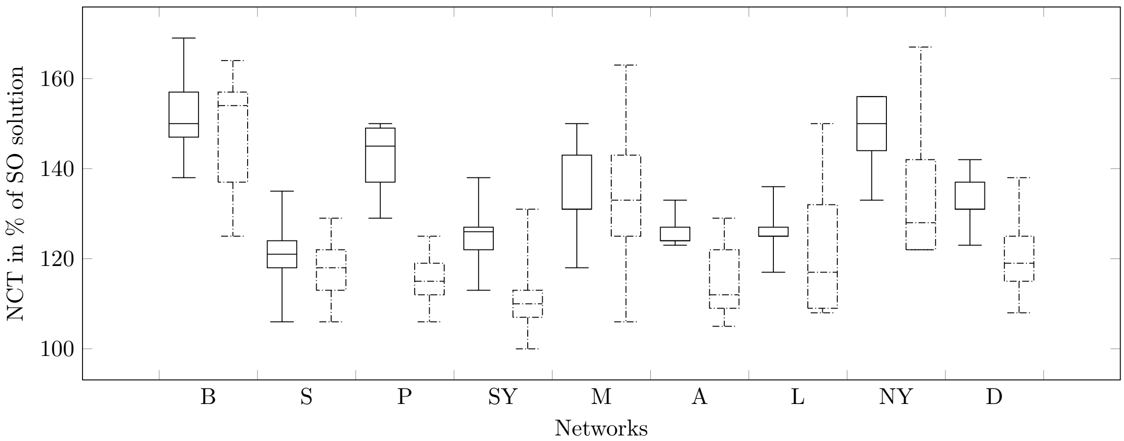



Figure 7.10: NCT of the Method with Sub-Networks and with Specific Blockages in Relation to the System-Optimal Solution (SO); Instances with 900 Evacuees.

For the data sets with 800 and 900 evacuees (Figure 7.9 and 7.10), both approaches lead to similar results for all tested instances. In summary, the approach with specific blockages resulted in average in better NCTs than the method of sub-networks.
The results show, traffic can be guided out of the affected area more efficiently when specific connections between sections are blocked. But it is supposed that many blockages are required to do so. The next chapter analyses how many blockages are invented by the computed solutions and to which extent this number can be minimised. Therefore, an additional objective function is proposed for the bi-level model.

# Chapter 8

# Guiding Selfish Evacuees with a Minimum Number of Street Blockages

The number of blockages which are used to guide the selfish evacuation traffic have a significant influence on the solution quality. With more blockages the traffic can be routed more precisely. But each blockage leads to high organisational effort, because relief units are necessary to block the connections between street sections. Moreover, the positions of blockages have to be communicated to the residents in the affected area. Hence the number of blockages should be as small as possible taking into account that the NCT is minimised. Therefore, this chapter induces an additional objective function that minimises the number of installed blockages. In Section 8.1 general aspects of multi-objective optimisation are illustrated and solution methods, which are relevant for this thesis, are presented. In Section 8.2 the impact of different blockage combinations on the solution is discussed and a second objective function is introduced. Section 8.3 presents a solution method that is based on lexicographic optimisation and Section 8.4 presents a heuristic that computes solution sets on the basis of the $\varepsilon$-constraint method. The chapter closes with a computational study in Section 8.5. The Sections 8.2 and 8.3 are based on the paper by Kimms and Seekircher (2017).

## 8.1 Multi-Objective Optimisation

Many real-world problems cannot be captured with a single objective function. Therefore, in the field of research in multi-objective optimisation a collection of objective functions is systematically and simultaneously optimised (Marler and Arora (2004)). In contrast to the single-objective optimisation for a majority of the cases it is not possible to find a generally accepted optimum. The best solution depends on the preferences of a (human) decision maker (Coello Coello, C. A. (1999)). A multi-objective problem (Marler and Arora (2004)) can formally be state as follows

$$min_{\mathbf{x}\in\mathbf{X}} \quad \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), ..., F_k(\mathbf{x})]^T \tag{8.1}$$

s.t.

$$g_j(\mathbf{x}) \leq 0, \qquad j = 1, 2, ..., m, \tag{8.2}$$
$$h_l(\mathbf{x}) = 0, \qquad l = 1, 2, ..., e. \tag{8.3}$$

Let $k$ be the number of objective functions, $m$ the number of inequality constraints, $e$ the number of equality constraints and $\mathbf{X}$ be the feasible decision space. $\mathbf{F}(\mathbf{x})$ is a vector of objective functions $F_i(\mathbf{x})$, $\mathbf{x}$ is a vector of decision variables, and $x_i^*$ is the point that minimises the objective function $F_i(\mathbf{x})$. In a bi-objective model, the vector $\mathbf{F}(\mathbf{x})$ includes two objective functions. For a multi-objective optimisation problem a single global solution can typically not be specified. Marler and Arora (2004) describe the solution as a set of points and the definition of an optimum depends on the decision makers preferences. In context of multi-objective optimisation the concept of the Pareto optimum is used to define a solution. A vector $\mathbf{x^*} \in \mathbf{X}$ is Pareto optimal, if there exist no feasible vector $\mathbf{x} \in \mathbf{X}$ which would decrease some objective functions without causing an increase by at least one other objective function. A multitude of Pareto optimal solutions exists for one problem, the Pareto optimal set (Coello Coello, C. A. (1999)).
The following paragraph defines solution concepts which can be used to determine solution sets depending on the preferences of the decision makers. Here, the most common methods and those that are relevant to this thesis are briefly summarised (for a comprehensive survey of various methods the reader is referred to Coello Coello, C. A. (1999) and Marler and Arora (2004)). Marler and Arora (2004) group the concepts into methods with a priori articulation of preferences, a posteriori articulation of preferences and without any articulation of preferences. In the following the methods with preferences articulation are presented. The methods without preference articulation are not relevant for the problem considered in this thesis.
For many optimisation problems the decision maker can specify preferences for the considered objective functions. In this case, the preference order can be integrated into the optimisation. One class of methods are scalarisation methods, where all objective functions are aggregated to one function. The most common method in this class is the weighted sum approach. Here, all objective functions are aggregated to one, each weighted with a coefficient that expresses the relative importance of the objective (Coello Coello, C. A. (1999)). Formally the objective function (8.1) can be aggregated to

$$min \sum_{i=1}^{k} w_i F_i(\mathbf{x}). \tag{8.4}$$

Let $w_i \geq 0$ ($\sum_{i=1}^{k} w_i = 1$) be the weighting coefficient representing the relative importance of objective $i$.
Another method is the lexicographic approach (Marler and Arora (2004)). In this approach the objective functions are ordered by their importance and the optimisation problem is solved separately with each objective function. The objective functions and values of the more important objectives are considered as constraints. Formally the problem is defined as

$$min_{\mathbf{x} \in \mathbf{X}} \quad F_i(\mathbf{x}) \tag{8.5}$$
s.t.
$$F_j(\mathbf{x}) \leq F_j(\mathbf{x}_j^*), \qquad j = 1, 2, .., i-1, i > 1, i = 1, 2, .., k. \tag{8.6}$$
$$g_j(\mathbf{x}) \leq 0, \qquad j = 1, 2, ..., m, \tag{8.7}$$
$$h_l(\mathbf{x}) = 0, \qquad l = 1, 2, ..., e. \tag{8.8}$$

Let $i$ be the function's position in the preference order and $F_j(\mathbf{x}_j^*)$ is the optimal value of the $j$-th objective function. Other methods, which are applicable to a given preference order, are for example the weighted min-max method or goal programming (Marler and Arora (2004)). For some optimisation problems it is difficult to determine a preference ranking a priori. In this cases solution methods based on a posteriori preference articulation can be used. In these methods a representation of the Pareto optimal set is computed and the decision maker can choose from a wide range of solutions (Marler and Arora (2004)). The $\varepsilon$-constraint method, is such a method (Coello Coello, C. A. (1999)). In this method one objective function is minimised, and the other objective functions are considered as constraints. These objective functions are restricted by parameter $\varepsilon_r$, which is a predefined goal that the decision maker wants to comply with. The method can be formally expressed as follows

$$min_{x \in X} \quad F_i(\mathbf{x}) \tag{8.9}$$

s.t.

$$F_r(\mathbf{x}) \leq \varepsilon_r \qquad r = 1, 2, ..., k \text{ and } r \neq i. \tag{8.10}$$

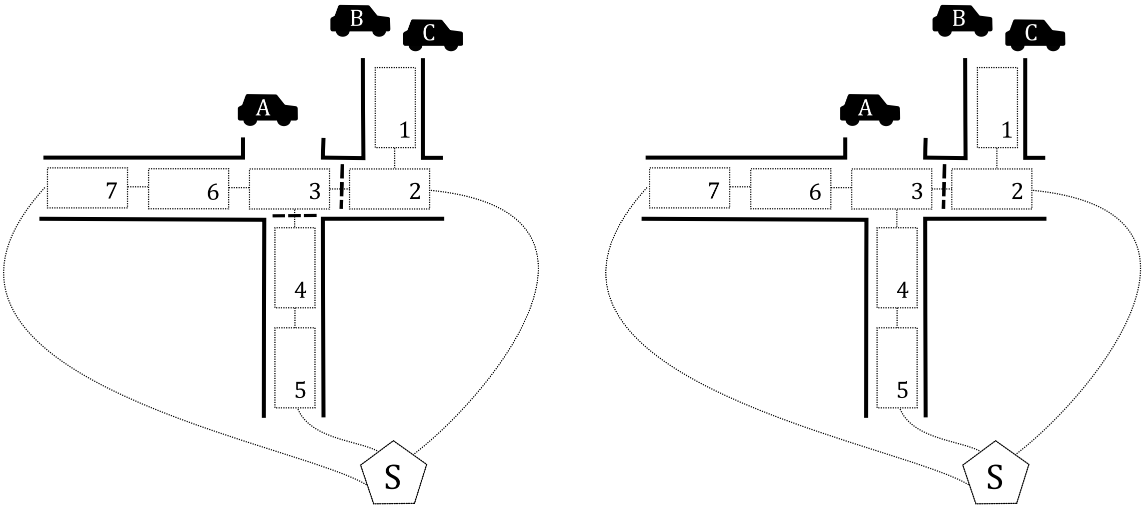
$$g_j(\mathbf{x}) \leq 0, \qquad j = 1, 2, ..., m, \tag{8.11}$$

$$h_l(\mathbf{x}) = 0, \qquad l = 1, 2, ..., e. \tag{8.12}$$

Let $r$ be all objectives without objective $i$, which is determined as the most important objective. For computing the Pareto optimal (sub)set the problem is solved with different values for parameter $\varepsilon_r$. The decision maker can choose a solution from this (sub)set that fits best for him. By using a posteriori methods, the decision maker can choose alternatives, on the basis of the stated results. Other a posteriori methods for example are physical programming or the normal constraint method (Marler and Arora (2004)). In addition to these methods, the methods with a priori preference articulation can be used to compute a set of solutions by using different parameter combinations.

## 8.2 Problem Description

In this thesis, evacuation plans are developed with the aim to evacuate the affected area as fast as possible and to minimise the NCT. To reach this goal, connections between street sections are blocked which in turn to guide the evacuation traffic. The number of blockages, which are necessary to lead the selfish evacuation traffic, was not yet considered for the optimisation. But in a real-life situation it is useful to compute solutions that have a minimum number of these blockages installed; especially when different numbers of blockages result in the same NCT. Moreover, in an emergency evacuation it is assumed that limited relief units are available that can block these connections. This limitation must be considered when a solution is developed. An additional objective function is defined that minimises the number of installed blockages. Before the second objective function is introduced, the problem that arises from different blockage combinations is illustrated in Figure 8.1. Figure 8.1a shows a network with three evacuees (A, B, and C) and two blockages. The first blockage is installed between sections 3 and 2 and the second one between sections 3 and 4. In the network without blockages all evacuees would take routes that include section 2 (1 - 2 - S and 3 - 2 - S) if it is assumed that all evacuees prefer the fastest way. These routes would lead to a NCT of 5 periods. When the blockages as depicted in Figure 8.1a are installed evacuee A is forced to take the alternative route 3 - 6 - 7 - S. In this case there is no need for evacuee A to wait until the evacuation starts and thus the NCT is reduced to 4 periods. As depicted in Figure 8.1b the same network is shown but only one connection between sections 3 and 2 is blocked. In this network evacuee A can choose between routes 3 - 4 - 5 - S and 3 - 6 - 7 - S. Both route choices will lead to the same NCT, because the routes have the same length and there are no other evacuees that take one of these routes. The solutions with one or two blockages lead to the same NCT.




(a) Network with Two Blockages. (b) Network with One Blockage.

Figure 8.1: Example to Illustrate the Impact of Different Numbers of Blockages on the NCT.

The example points out that different blockage combinations can lead to the same NCT, because the number of used blockages is not taken into account when minimising the NCT. But as stated above in a real-life scenario each blockage leads to organisational costs; e.g. communication efforts or relief units that block the street sections. To consider the number of blockages in the optimisation problem, a second objective function is formulated for the bi-level model, which was presented in Section 7.2. The blockages are determined in the lower-level problem. Thus, the second objective function is added to the mathematical model (7.6) - (7.28)

$$\min \sum_{i \in I} \sum_{j \in I} (1 - m_{ij}) \beta_{ij}. \tag{8.13}$$

The blockage of connections is determined with $m_{ij}$, and for each determined blockage is $m_{ij} = 0$. As a reminder: the parameter $\beta_{ij}$ indicates the connections between sections, thus this parameter determines the network structure. In the objective function (8.13) the number of blocked sections is counted and minimised. The additionally formulated objective function is contrary to the objective function (7.6). With more blockages the traffic can be guided more precisely but these blockages lead to a higher organisational effort and higher costs. By computing a solution for this problem the contradiction of the objectives must be considered.

## 8.3 A Solution Procedure with Lexicographic Optimisation

In order to precisely guide the traffic it is recommended to use blockages. However, it is discussed in Section 8.2 that as little blockages as possible should be invented in a network to minimise the organisational effort. Contrary the invention of a limited number of blockages will result in a more disordered traffic guidance. Thus the two objectives, which are considered in this chapter, are contrary to each other. Therefore a solution method, which considers the compromise between both objective functions is used. This thesis is motivated by the main objective of an evacuation plan: the affected area should be cleared as fast as possible. Based on this motivation an obvious preference ranking between both objective function arises. In Section 8.1 different methods, which deal with bi-objective models with a priori preference articulation, are presented. To capture the clear preference ranking and to counteract the problem that the same NCT can be reached with different numbers of blockages, the method of lexicographical optimisation is used. The problem is solved for the first objective function without considering the second objective function. If the problem with the first objective function will lead to multiple solutions, then a solution is determined that is in minimum for the second objective. First, a solution with a minimum NCT is identified and if it is possible to reach this NCT by applying different blockage combinations, a solution with a minimum number of blocked connections will be computed. To do so, the bi-level problem (7.6) - (7.28) is solved first. The resulting NCT determines the threshold that has to be considered when the number of blockages is minimised. Therefore, the model (7.6) - (7.28) is extended by constraint (8.14)

$$\sum_{t \in T} \nu_t \leq NCT^*. \tag{8.14}$$

The additional constraint states that the number of periods where evacuees are still in the network has to be smaller or equal to $NCT^*$. Herein, $NCT^*$ is the best NCT that is computed with the first objective function model. The solution approach for the described problem can be summarised with the following steps:

**Step 1:** Compute a solution for the bi-level problem (5.1), (5.3) - (5.6), (7.5), and (7.6) - (7.28). Get as a result a (minimum) $NCT^*$.

**Step 2:** Compute a solution for the bi-level problem (5.1), (5.3) - (5.6), (7.5), and (7.7) - (7.28), (8.13), (8.14). The (minimum) $NCT^*$ is considered in constraint (8.14).

The solutions for the bi-level model in step 1 and step 2 can be computed with one of the solution approaches, which are presented in Section 7.3.
The NCT that is used as a threshold in the model formulation in step 2 is the objective value of a feasible solution for the network design problem, computed in step 1. Thus it is generally possible to find a solution for the model formulation in step 2 which complies with this NCT. But, as a cause of the iterative solution approach, it is possible that the blockage combination computed in step 1 cannot be found. The model in step 2 is solved iteratively and in each iteration a solution with a minimum number of blockages is computed. This can lead to a blockage combination, which does not comply with the NCT computed in the first step and a feasible solution cannot be found.

To deal with this problem, the solution approach is extended by the possibility to reverse a part of the computed blockages. The extension ensures that the solution approach always leads to a feasible solution. The variable $\eta \geq 0$ is introduced to increases the NCT computed in step 1, if constraint (8.14) cannot be met. Constraint (8.14) is extended by $\eta \geq 0$

$$\sum_{t \in T} \nu_t \leq NCT^* + \eta. \tag{8.15}$$

With $\eta$ the constraint will be relaxed, if it is not possible to compute a solution with the given $NCT^*$ and the blockages computed in the previous iterations (for reminder, in each iteration additional blockages are computed, see Section 7.3). To prevent that a solution with a lower number of blockages and with a higher NCT is computed, the use of $\eta$ is punished in the objective function. Thus the $NCT^*$ is only relaxed ($\eta > 0$) when no feasible solution with the given $NCT^*$ and in the previous iterations computed blockages can be computed. The parameter $\lambda\eta$ is added to objective function (8.13)

$$\min \sum_{i \in I} \sum_{j \in I} (1 - m_{ij})\beta_{ij} + \lambda\eta. \tag{8.16}$$

To ensure that $\eta$ is only used to prevent infeasible solutions the parameter $\lambda$ is set to a large number. $\eta > 0$ indicates that with the blockages made in previous iterations a solution with $NCT^*$ cannot be found. Thus in the solution approach some of the determined blockages have to be reversed to get a feasible solution that yields $NCT^*$. The extended solution approach consists of the following steps:

**Step 1:** Compute a solution for the bi-level problem (5.1), (5.3) - (5.6), (7.5), and (7.6) - (7.28). Get as a result a (minimum) $NCT^*$.

**Step 2:** Compute a solution for the bi-level problem (5.1), (5.3) - (5.6), (7.5), and (7.7) - (7.28), (8.15), (8.16). The objective value from step 1 is considered in constraint (8.15). To get a solution the upper- and the lower-level problems are solved iteratively. The following decisions are made in dependence on $\eta$:

(a) There is a solution with $\eta = 0$:
   - The solution is a complete solution: go to step 4.
   - The solution is an incomplete solution: a feasible solution is still possible, start again with step 2.

(b) There is a solution with $\eta > 0$. The determined blockage combination cannot lead to a feasible solution with the given $NCT^*$: go to step 3.

**Step 3:** Identify all blockages, $Fix^0$. Choose a random percentage $h$ ($h \in [0, 1]$) of the determined blockages for example $|Fix^0| \times h$ blockages and reverse them. Go to step 2 and compute the next step with the remaining blockages.

**Step 4:** The algorithm stops, if a complete feasible solution is found.

The complete and incomplete solutions (step 2) have the same meaning as the complete and incomplete blockage combinations defined in Section 7.3.3.

## 8.4 An $\varepsilon$-Constraint Based Solution Method

In Section 8.1 different methods for multi-objective models are described. One part of the methods deals with problems without an clear preference ranking. Hence, in these methods a set of solutions are computed and the decision maker can choose one solution that fits best. A solution approach that is based on the $\varepsilon$-constraint method is introduced to compute such a set of solutions. As discussed in Section 8.2 the possibilities to install blockages between street sections could be limited for example by the available number of relief units. Hence, these limitations have to be considered when developing evacuation plans. With the $\varepsilon$-constraint method the best possible NCT is computed for a maximum number of predetermined street blockages. The second objective function (8.13), which is introduced in Section 8.2, is added as a constraint to model (7.6) - (7.28). The additional constraint is

$$\sum_{i \in I} \sum_{j \in I} (1 - m_{ij}) \beta_{ij} \leq \varepsilon. \tag{8.17}$$

With various values for parameter $\varepsilon$ a set of solutions is computed and the decision maker can chose one of these solutions. With this solution approach the effect of additional blockages on the NCT, can be considered by the evacuation planner. While minimising the NCT is the main objective, other solutions can be taken into account to get a more detailed view on the problem. The problem can be solved with one of the heuristics introduced in Section 7.3.
Useful values for $\varepsilon$ can be determined in the range between 0 (user-optimal solution) and the number of blockages, which are determined with the solution approach in Section 8.3. More than these blockages are not needed because they would not lead to a reduction in NCT. To get the complete set of solutions for each number of blockages in the determined range, the bi-level model (5.1), (5.3) -(5.6), and (7.6) - (7.28), (8.17) has to be solved. Depending on the particular problem it could be reasonable to vary the values for $\varepsilon$ in larger steps and to compute only a subset of all possible solutions.

## 8.5 Computational Study

This chapter focuses the number of blockages that are needed to guide the traffic. As illustrated in Section 5.1.2 the proposed method from Chapter 7 enables a very detailed blockage of connections between street sections because the blockage of individual lanes is considered. It is hypothesised that a lot of blockages are necessary to guide the traffic in an ordered way and to achieve low NCTs. Table 8.1 presents the number of connections in each network ($\beta_{ij}$ and $\beta_{ji}$ are counted as one connection) and the computed blockages that results from minimising the NCT. The values are computed with the basic solution approach for the bi-level model (see Section 7.3.1). For each network and each set of evacuees the average, the minimum (Min) and the maximum (Max) number of used blockages is presented. The results show, that the number of evacuees does not have a significant influence on the number of blockages. For each network the number of blockages are close to each other irrespective of the number of evacuees. In contrast the number of connections in the network has an influence on the number of necessary blockages. In the networks with less connections (B - A) also less blockages are necessary in comparison to the network with more connections (L - D). But the number of connections is not the only factor that affects the number of blockages. When comparing the blockages and connections in network B with those in network SY it is obvious that both networks have 126 connections, but in network B an average of abound 50 blockages and in network SY an average of about 22 blockages are used to guide the traffic. Thus further factors like the number of alternative routes or exits influence the number of used blockages. At first glance it seems that a lot of blockages are necessary for the traffic guidance. But the number of blocked connections presented in Table 8.1 does not necessarily correlate to the number of blockages that have to be positioned in a real street network. A lot of blocked connections could be summed up to one blockage. Reminder: when it is forbidden to leave street section A in Figure 5.3 then three

| #-Connections | B 126 | S 131 | P 133 | SY 126 | M 164 | A 168 | L 250 | NY 303 | D 254 |
|---|---|---|---|---|---|---|---|---|---|
| 500 | | | | | | | | | |
| Average | 45 | 46 | 40 | 22 | 47 | 29 | 51 | 79 | 56 |
| Min | 37 | 37 | 33 | 16 | 42 | 23 | 39 | 47 | 42 |
| Max | 55 | 52 | 50 | 30 | 52 | 35 | 65 | 109 | 71 |
| 700 | | | | | | | | | |
| Average | 50 | 45 | 40 | 21 | 47 | 31 | 81 | 122 | 69 |
| Min | 41 | 39 | 32 | 19 | 39 | 24 | 70 | 87 | 55 |
| Max | 55 | 49 | 44 | 26 | 57 | 36 | 89 | 140 | 83 |
| 800 | | | | | | | | | |
| Average | 50 | 48 | 45 | 24 | 24 | 36 | 88 | 126 | 73 |
| Min | 39 | 45 | 40 | 13 | 40 | 30 | 80 | 114 | 61 |
| Max | 57 | 51 | 53 | 30 | 60 | 42 | 97 | 134 | 81 |
| 900 | | | | | | | | | |
| Average | 52 | 47 | 44 | 22 | 57 | 37 | 86 | 128 | 77 |
| Min | 47 | 43 | 32 | 10 | 53 | 34 | 73 | 121 | 63 |
| Max | 58 | 50 | 48 | 28 | 59 | 41 | 97 | 135 | 90 |

Table 8.1: Number of Blocked Connections by Minimising the NCT.

blockages are determined in the model. But in a real street network just one blockage has to be positioned. Nonetheless for some networks a lot of blockages are required. In the instances of network NY more than hundred blockages are computed. In such networks it could be useful to follow a different strategy to guide the traffic. In contrast to that, in the instances of network SY, an average of about 20 blocked connections is determined. In such networks the blockage of street sections is a good way to cope with selfish evacuation traffic.

*Blockage minimisation with lexicographic optimisation*

With the methods presented in this chapter the number of blockages that are necessary to guide the traffic can be significantly reduced. Due to high computational effort the solution approach introduced in Section 8.3 is tested only for selected instances. In this test the maximum computation time is set to 3 hours. The percentage of blockages that are reversed in the algorithm in step 3 is fixed to $h = \frac{1}{3}$. If the determined number of reversed sections is not an integer number, this number will be rounded. The approach has been run for each network and each set of evacuees with instance $*\_*\_a$. The $NCT^*$ is fixed to the results computed with the basic solution approach presented in Section 7.3.1. In 3 hours for 23 out of 36 instances a result could be computed. In Table 8.2 the number of blockages for instances $*\_500\_a$ are presented exemplary. The second column depicts the number of blocked connections computed with the iterative approach by minimising the NCT, and the third column depicts the blockages with the solution approach which is based on lexicographical optimisation.
The results point out, that the number of blockages that are used to guide the traffic can be significantly reduced. In the lower-level problem (Section 7.2) the number of used blockages

| Instance | NCT Minimisation | Blockage Minimisation |
|---|---|---|
| B_500_a | 40 | 26 |
| S_500_a | 52 | (-) |
| P_500_a | 36 | 32 |
| SY_500_a | 19 | 8 |
| M_500_a | 47 | 25 |
| A_500_a | 35 | 23 |
| L_500_a | 43 | 18 |
| NY_500_a | 63 | 25 |
| D_500_a | 55 | 6 |

Table 8.2: Number of Blockages by Minimising the NCT and the Number of Blockages.

is not relevant. Thus a lot of unnecessary blockages are determined. This effect is intensified by the iterative approach. By minimising the number of blockages for a predefined NCT, a solution with a significantly lower number of blockages can be computed. Hence, it is useful to consider the additional objective function to minimise the used blockages. A drawback that results from the second objective function is the increase in computation time. For example the computation time for instance $S\_500\_a$ solved with the iterative approach, was round about 96.1 CPU seconds. By considering the second objective function and using the proposed solution method it was not possible to compute a result for this instance within 3 hours. For nearly all the instances where a solution could be computed in the given time span, the number of blocked connections could be halved. A different way to reduce the number of blocked connections is introduced with the $\varepsilon$-constrained based method.

*Solution sets computed with the $\varepsilon$-constraint based method*

Additionally to the results computed with the solution procedure based on lexicographical optimisation, where the minimal number of blockages is focused, now a set of solutions is computed. Therefore, the $\varepsilon$-constraint based heuristic proposed in Section 8.4 is tested. To compute a set of solutions seven different values for parameter $\varepsilon$ are used. The method is computed exemplarily for each network with the instance $^*\_800\_a$. For the $\varepsilon$-values the number of blockages computed by the minimisation of the NCT are used as thresholds. The resulting numbers are reduced by 20 % to 80 % (rounded up to the next integer) in steps of 10 percent points. The seven data-points (1 - 7) for parameter $\varepsilon$, and the initial number of blockages for the used instances are listed in Table 8.3. Several values of $\varepsilon$ lead to the same NCT with different numbers of blockages or a higher NCT is computed with more blockages than in other instances. Thus, for each instance at most 5 solutions are considered, as can be seen in Figures 8.2 - 8.4. In these figures, the NCT and the used number of blockages are illustrated. The NCT with 0 blockages is the solution with user-optimal flows. With these results a decision maker can choose between different numbers of blocked connections and the resulting NCT.

| Instance | #-Blockages | ε-values 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| B_800_a | 48 | 39 | 34 | 29 | 24 | 20 | 15 | 10 |
| S_800_a | 45 | 36 | 32 | 27 | 23 | 18 | 14 | 9 |
| P _800_a | 43 | 35 | 31 | 26 | 22 | 18 | 13 | 9 |
| SY_800_a | 13 | 11 | 10 | 8 | 7 | 6 | 4 | 3 |
| M_800_a | 54 | 44 | 38 | 33 | 27 | 22 | 17 | 11 |
| A_800_a | 34 | 28 | 24 | 21 | 17 | 14 | 11 | 7 |
| L_800_a | 96 | 77 | 68 | 58 | 48 | 39 | 29 | 20 |
| NY_800_a | 134 | 108 | 94 | 81 | 67 | 54 | 40 | 27 |
| D_800_a | 77 | 62 | 54 | 47 | 39 | 31 | 24 | 16 |

Table 8.3: Dataset for Parameter $\varepsilon$.

Figure 8.2 depicts the results of the small networks. For instance B_800_a five combinations of blockages and NCTs are computed. Without blocking the NCT is 22 periods, with 24 blockages the NCT could be reduced to 18 periods and with 33 blockages to 15 periods. A decision maker can choose which blockages should be positioned in the network, taking the available number of relief units into account.

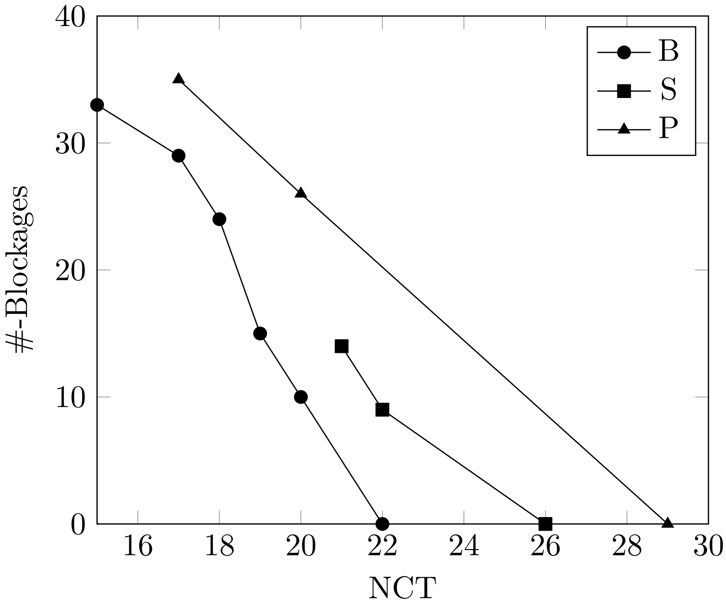



Figure 8.2: Illustration of the Solution Sets of the Instances with Small-Sized Networks.

In Figure 8.3 the illustration of the results from instance M_800_a points out, that only a small decrease of NCT can be achieved by blocking sections: 22 blockages lead to a NCT of 21 periods and further 11 blockages lead to a NCT of 20 periods. The evacuation planner can decide whether the benefits (minimal savings in NCT) justify the costs (additional efforts to block sections). The results show that it is advantageous to consider the number of blockages when making evacuations plans. The results computed with the solution approach based on lexicographical optimisation make clear, that the same NCT can be reached with various blockage combinations and numbers of blocked connections. To reduce the organisational effort the number of blockages should be as small as possible. Furthermore, the combinations of different numbers of blockages and resulting NCTs show, that it could be useful to take a closer look at these combinations. If a high number of additional blockages is needed to achieve a minimal reduction of the NCT, it might be reasonable to chose an unfavourable NCT that goes along with a reduced number of blockages.

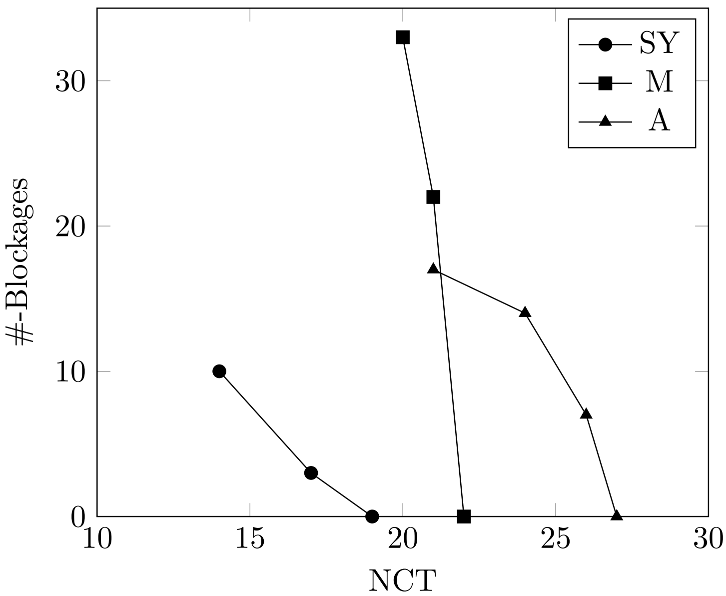



Figure 8.3: Illustration of the Solution Sets of the Instances with Medium-Sized Networks.

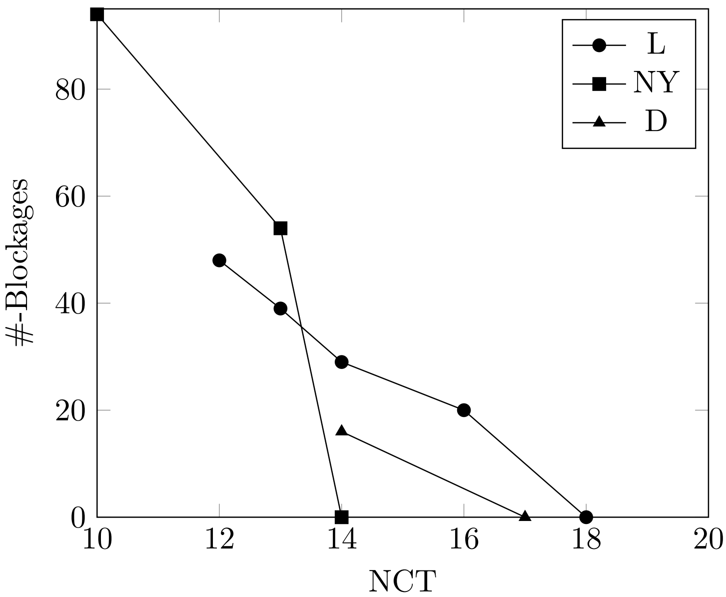



Figure 8.4: Illustration of the Solution Sets of the Instances with Large-Sized Networks.

For example in Figure 8.4, in instance NY_800_a 54 blockages are necessary to reduce the NCT from 14 to 13 periods. The opposite is given in Figure 8.3 for instance A_800_a: three additional blockages (from 14 to 17 blockages) can reduce the NCT from 21 to 24 periods - a small additional effort of installing blockages leads to a significant reduction in NCT.

# Chapter 9

# Conclusions

The route selection, which people make to reach a destination, has a significant impact on the traffic flow in a network. If all the network users choose the same routes, these routes will be congested and the traffic flow in the network decreases. These findings are relevant for daily traffic routing as well as for the traffic flow in extraordinary situations like in evacuation scenarios. However, just a few studies in the literature of evacuation traffic management consider this relevant aspect.

This thesis studies the evacuation of urban areas taking into account selfishly acting evacuees. In case of an evacuation the traffic must be routed out of the affected area to protect the lives of the residents. In such a situation, the capacity demand most certainly exceeds the network capacity planned for daily traffic. Hence, evacuation plans which guide the evacuees are necessary to make the most of the scarce capacities. In most of these plans optimal routing or timing-strategies are developed to clear the affected area. But various studies show that people do not follow the instructions of authorities in evacuations. Accordingly, those evacuation plans that require the evacuees to adhere to optimal strategies may fail. Therefore, this thesis proposes an evacuation strategy that considers the behaviour of evacuees. With this strategy the street network is adjusted to guide the evacuees out of the affected area and the evacuation routing is optimised. These plans are applicable even when the evacuees choose their routes selfishly and do not follow the instructions from authorities. The proposed method is based on Braess's paradox. Contrary to the findings from Braess street sections were blocked to guide the traffic flow. The position of the blockages was determined in response to the route choice preferences of the evacuees. Thus, at first various behaviour patterns as described in the literature were investigated and in a subsequent step the evacuee's route selection was modelled according to these patterns. The first method that was proposed in this thesis followed the idea of sub-networks to restrict the possibilities of the evacuees. Thus, the network was divided into sub-networks with one exit in such a way that the NCT was minimised. As a drawback of this method it was found that the traffic cannot be guided within these sub-networks. Hence, another idea had been introduced, which blocked specific connections between sections to achieve a more precise traffic routing. The main assumption of this thesis is, that the routes of the evacuees cannot be determined by a central decision maker. Thus, the route choice and the network optimisation were considered separately and the problems were formulated as a bi-level model. The optimal routes of the evacuees were computed by the upper-level problem and the network was optimised by the lower-level problem. Taking into account that each blockade leads to organisational effort a second objective function, which minimises the number of required blockages was introduced.

With an additional objective function in the bi-level model a solution with a minimum number of blockages, was determined. Moreover, a set of different combinations of blockage numbers and resulting NCT was computed to support the decision making process of the evacuation planner. Furthermore, a multitude of heuristic solution approaches was developed to find solutions for the proposed strategies and comprehensive computational studies were conducted to test them.

Chapter 6 introduced the concept of sub-network formation to guide the traffic flow. As stated above in this concept the network is divided into sub-networks with just one exit. The sub-networks are computed on the basis of the weights $g_{is}$ that indicate the worst possible evacuation time necessary to leave the affected area, when section $i$ is assigned to the sub-network with exit $s$. Each section of the network is assigned to exactly one sub-network with exit $s$ and considering the minimisation of the sum over the related $g_{is}$ values. To implement these sub-networks all connections between sections that belong to different networks are blocked. To determine the weights $g_{is}$ an iterative solution procedure was presented and for the assignment from sections to sub-networks, a mathematical model as well as a heuristic approach were introduced. Furthermore, an iterative solution approach was proposed that combines the computation of the weights $g_{is}$ with the assignment of sections to sub-networks. In a computational study, first different parameters of the solution approach were tested to fine-tune the heuristic's parameters and afterwards the influences of network modifications on the NCT were investigated. The results of the computational study show that the computation of sub-networks leads to a reduction in NCT when comparing it to the user-optimal solution. Moreover, the spread in the NCT for various distributions of evacuees in the network is reduced with these sub-networks. A test with different numbers of exits in the network was executed and the results show, that besides the number of exits also the position of the exits has an impact on the NCT.

Chapter 7 presented the method of blocking specific connections between sections. This method is a countermeasure to overcome the drawback from the method that was introduced in Chapter 6 which stated the traffic cannot be routed within the sub-networks. To consider that the route choice of evacuees and the determination of blockages was processed by independent decision makers, a bi-level optimisation approach was applied. Several heuristics were presented to solve the bi-level model and to determine the position of blockages that guide the selfish evacuation traffic. In a basic heuristic the upper- and lower-level problems were solved iteratively until no additional blockages were needed anymore to route the traffic. Additionally, a preprocessing algorithm was developed to identify connections that must not be blocked. To reduce the computational effort the decision variables were fixed for these connections. In the basic solution approach the upper- and lower-level problems are solved iteratively, and in each iteration the optimal solution of the upper- and lower-level problems is considered. Tests show, that the optimal solution of each problem does not always lead to the best evacuation plan. Therefore, another heuristic was developed that considers different combinations of blockages. The results of the computational study show that the blockage of specific connections leads to more favourable results than the sub-network method. Moreover, it was shown that the preprocessing procedure reduces the computation time for the majority of the tested instances. Only for some instances related to the large-sized networks preprocessing leads to an increase of computation time. With the blockage-combination heuristic for instances of the small- and medium-sized networks an improvement of the NCT was achieved. For the instances of large networks, only for one network an improvement was computed. In this test the computation time was fixed to 1800 sec. For the instances with large-sized networks, the maximum computation time was not high enough to achieve improvements with the blockage-combination heuristic.

In addition to the objective function which was considered in Chapter 6 and Chapter 7 in Chapter 8 the number of street blockages was minimised. Every installed blockage leads to organisational cost, hence a solution with a minimum of street blockages is preferred considering the minimisation of the NCT. According to the problem definition there is a clear preference ranking between both objective functions which is best suited for the application of a lexicographical optimisation. Therefore, a heuristic based on this method was developed. With this method a solution with a minimum NCT was computed. If different blockage combinations lead to the same NCT, the solution with the minimum number of blockages was chosen. Additionally, a solution procedure that is based on the $\varepsilon$-constraint method was presented. This method computes a set of solutions with different numbers of blockages and their corresponding NCTs. This sub-set of solutions represents an instrument that supports the decision-making process. It provides a closer view on the computed solutions and also considers aspects like the limited numbers of relief units. The computational study compared the number of blockages that was computed by a) minimising the NCT and by b) minimising the number of blockages. When the additional objective function is considered the number of used blockages could be halved in most test instances. The computed solution set shows, that for some instances a lot of blockages are necessary to achieve a small reduction in the NCT. When not all blockages can be positioned in the networks that are needed to achieve the minimum NCT it can be useful for an evacuation planner to compare different numbers of blockages and the related NCT.

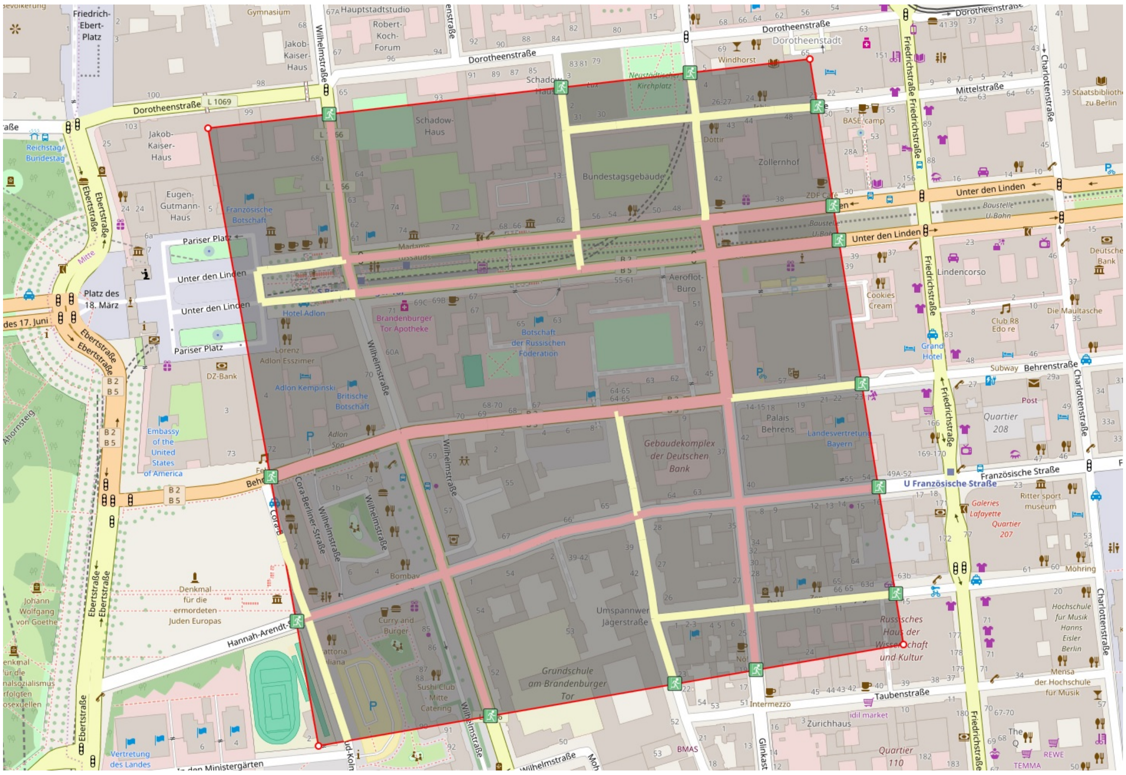
This thesis makes clear how important it is to consider the selfish behaviour of the evacuees in the evacuation planning. Only a few studies exist that combine both topics, thus it is obvious that further research is necessary in this area. Most of the methods that are applied in case of selfish routing in general traffic scenarios cannot be adopted to selfish routing in evacuation scenarios. Due to extensive modifications of the street network these methods cannot be realised in case of an evacuation. Therefore, strategies are developed which can be quickly implemented. The conducted computational studies show that the blockage strategy to guide the traffic out of the affected area is a proper way to tackle the problem of selfishly acting evacuees. These network modifications can be quickly implemented and the effort to communicate the adjustments to the evacuees is small. The positions of the blockages can be illustrated for example on a map and the network users can choose their routes considering these blockages. Although good results can be achieved by blocking street sections the implementation of such blockades can be very costly for certain network structures. Future research should address the development of other strategies for such networks to be able to cope with selfish routing in evacuation scenarios. In this thesis it is assumed, that all networks users behave selfishly. This assumption is realistic, because various studies showed, that most of the evacuees choose their routes according to their preferences and do not follow the instructions of authorities. But it is also realistic to assume that a part of the evacuees follow the instructions of authorities. Thus, the problem could be extended by considering two types of evacuees. With this additional evacuee type, the concept of Stackelberg routing can be applied. For the evacuees that follow the instructions of the evacuation planner a (Stackelberg) routing strategy can be implemented and the remaining network users route selfishly (see for example Roughgarden (2004), Bonifaci et al. (2010)). As known from the literature this strategy reduces the consequences of selfish routing.
In this thesis multiple heuristics were developed to cope with the concept of sub-networks and the bi-level model for network optimisation with selfishly acting evacuation traffic. In future

research a question of efficient optimal procedures for the described problem may be investigated. To develop an optimal solution method for the bi-level model presented in Chapter 7 in a first consideration the bi-level model can be transformed into an integrated form. Furthermore, criteria can be developed to identify street sections that lead to Braess's paradox. By blocking these sections the occurrence of the Braess paradox can be prevented.
The optimisation of evacuation plans is an important research area in the field of operations research. With these optimal plans evacuations can be performed more successfully and the consequences of disasters can be reduced. In this thesis a further aspect of the research area of evacuation planning using methods of operations research was investigated. It was demonstrated that it is important and reasonable to consider the people's behaviour when planning evacuation scenarios. To the best of our knowledge, this thesis belongs to one of the few studies that combine evacuation planning and selfish routing using optimisation methods. Most studies that combine evacuation planning with selfish acting evacuees, focus on the simulation of the traffic flow and not on its optimisation. Therefore, further research is needed to compute realistic evacuation plans which can be implemented in real-life scenarios.

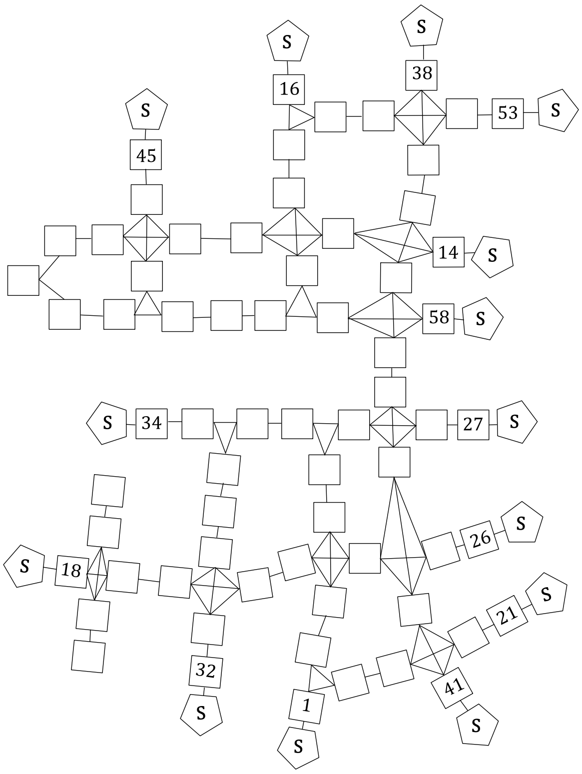
# Appendix A

# Test Networks

The Figures A.1 to A.9 illustrate the networks, which are used in the computational studies. In each figure first the original networks generated with the software tool SPSE (SPSE (2016)) are presented and second the networks transformed into sections are depicted. With the exits signs / pentagons (S) the sections, which are connected with the super sink S, are marked.

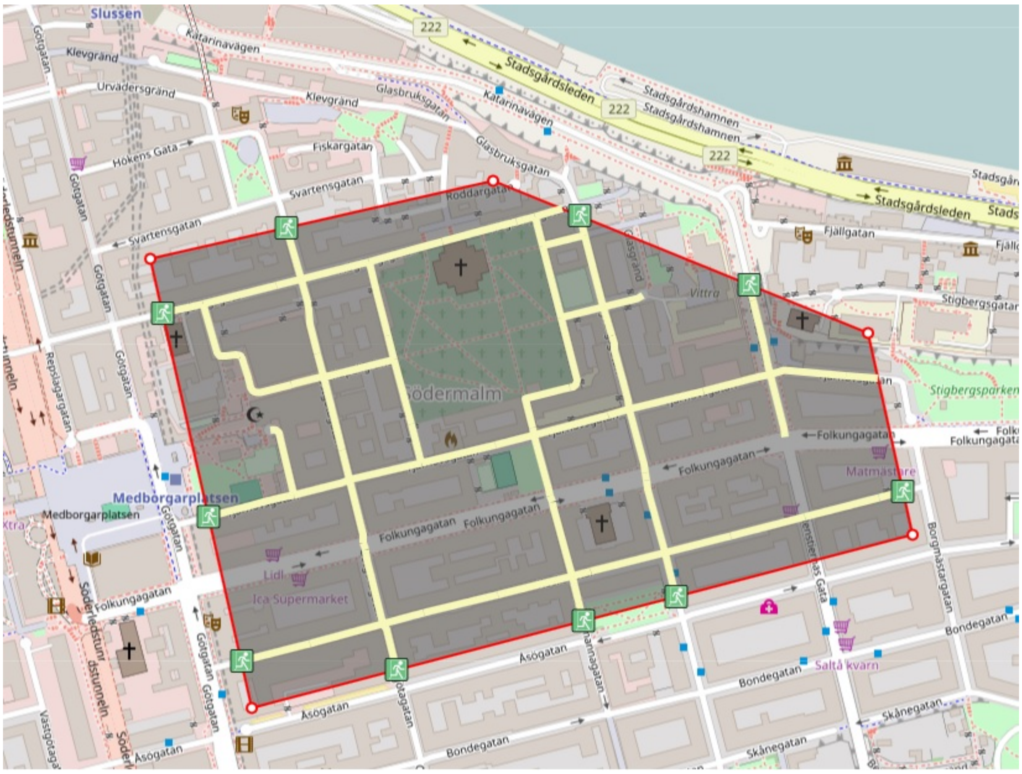



(a) Original Network.

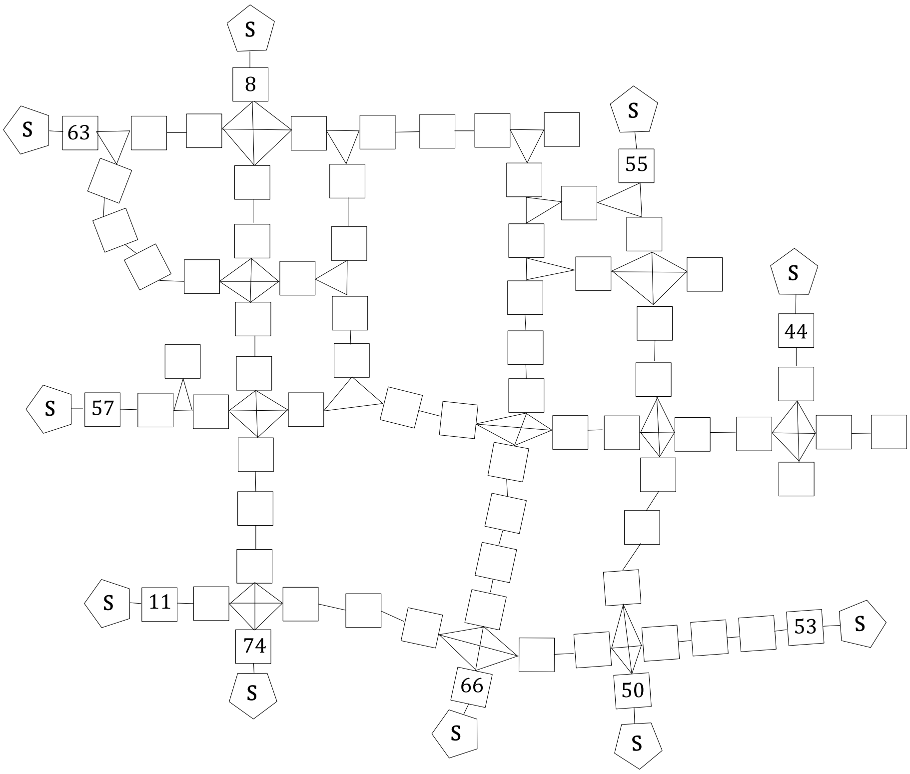



(b) Network Transformed into Sections.

Figure A.1: Network Berlin.



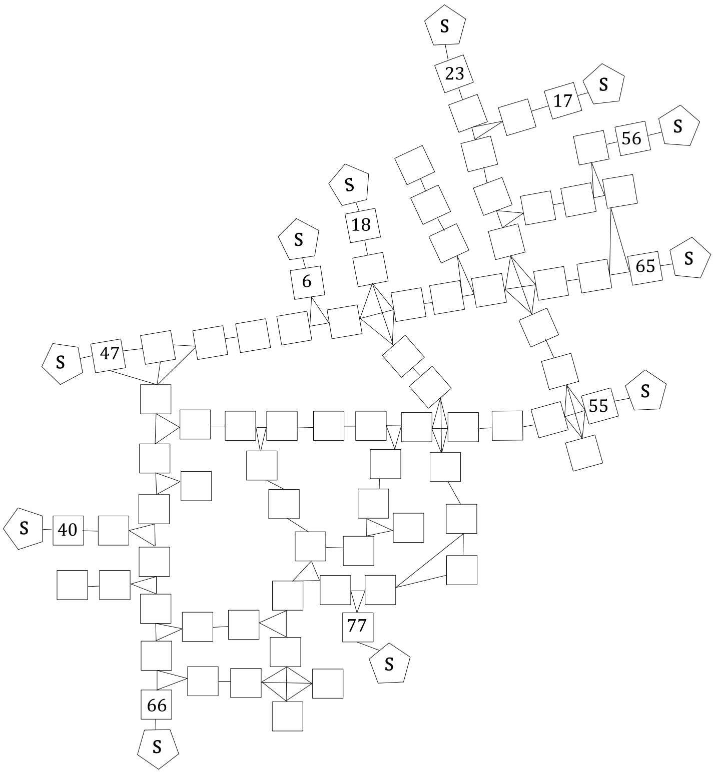

(a) Original Network.




(b) Network Transformed into Sections.

Figure A.2: Network Stockholm.

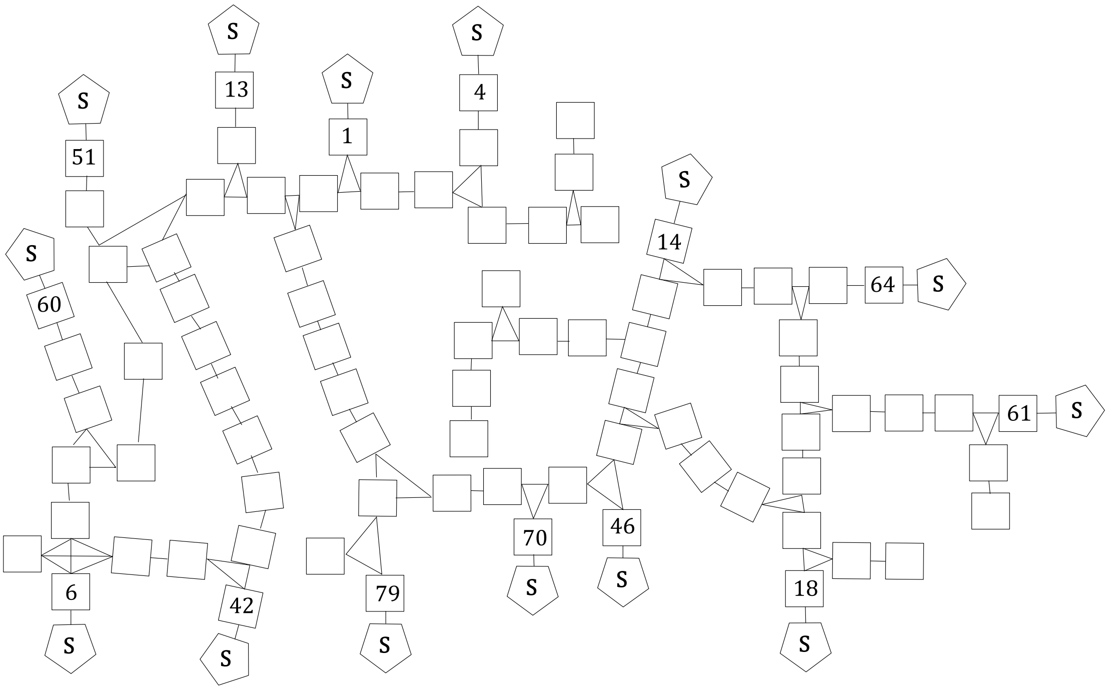

(a) Original Network.




(b) Network Transformed into Sections.

Figure A.3: Network Paris.

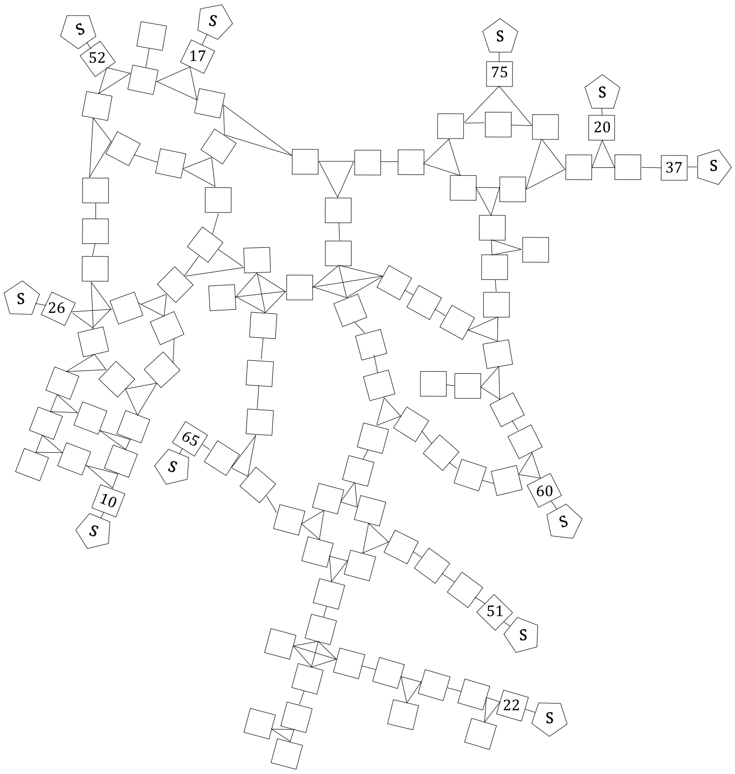

(a) Original Network.




(b) Network Transformed into Sections.

Figure A.4: Network Sydney.

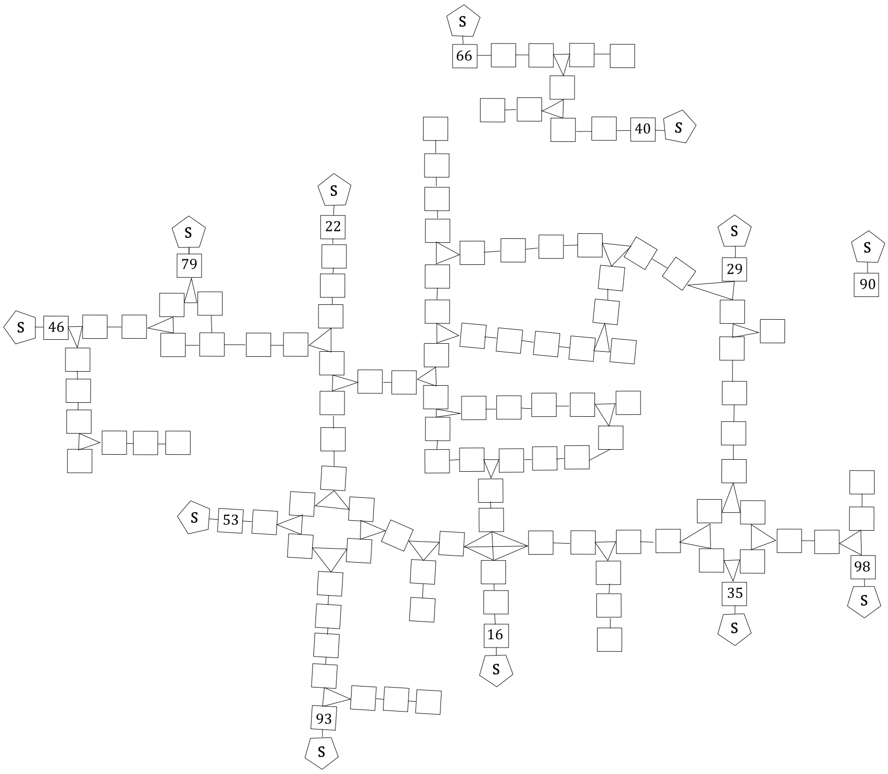

(a) Original Network.




(b) Network Transformed into Sections.

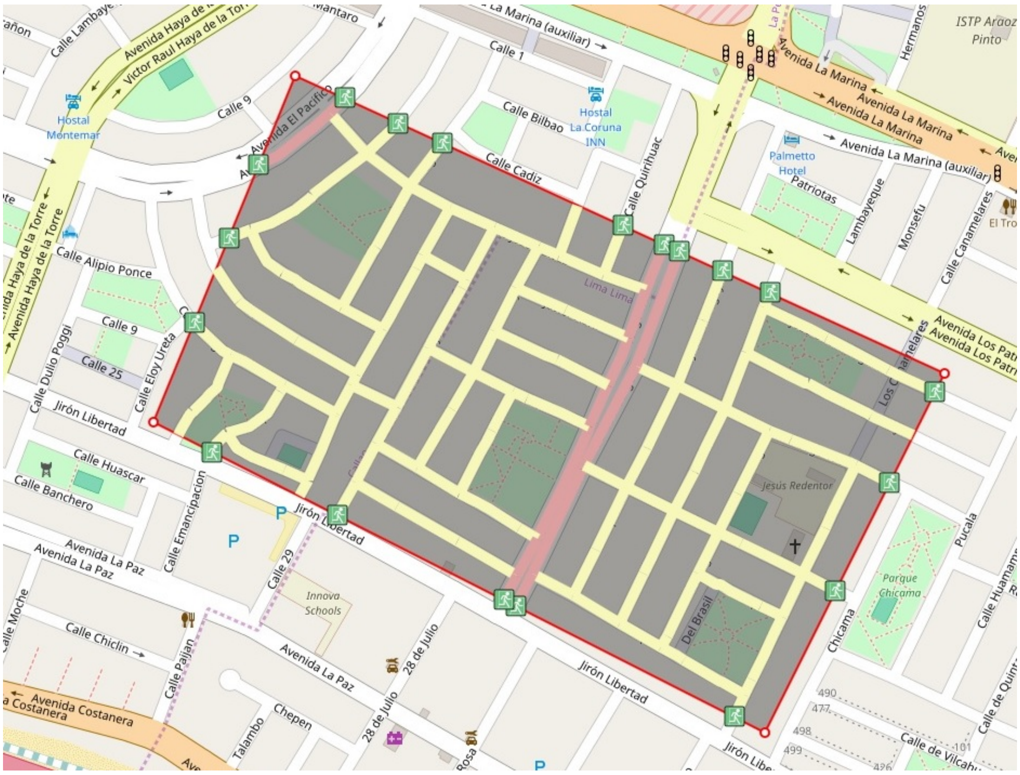
Figure A.5: Network Melbourne.


(a) Original Network.

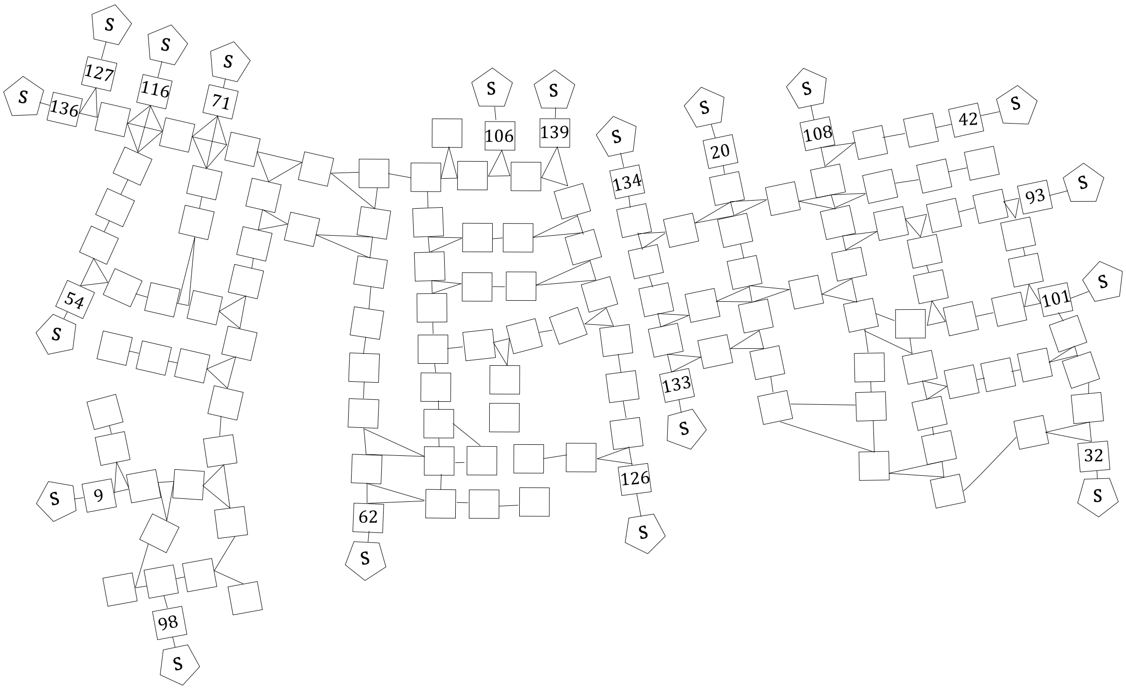



(b) Network Transformed into Sections.

Figure A.6: Network Auckland.



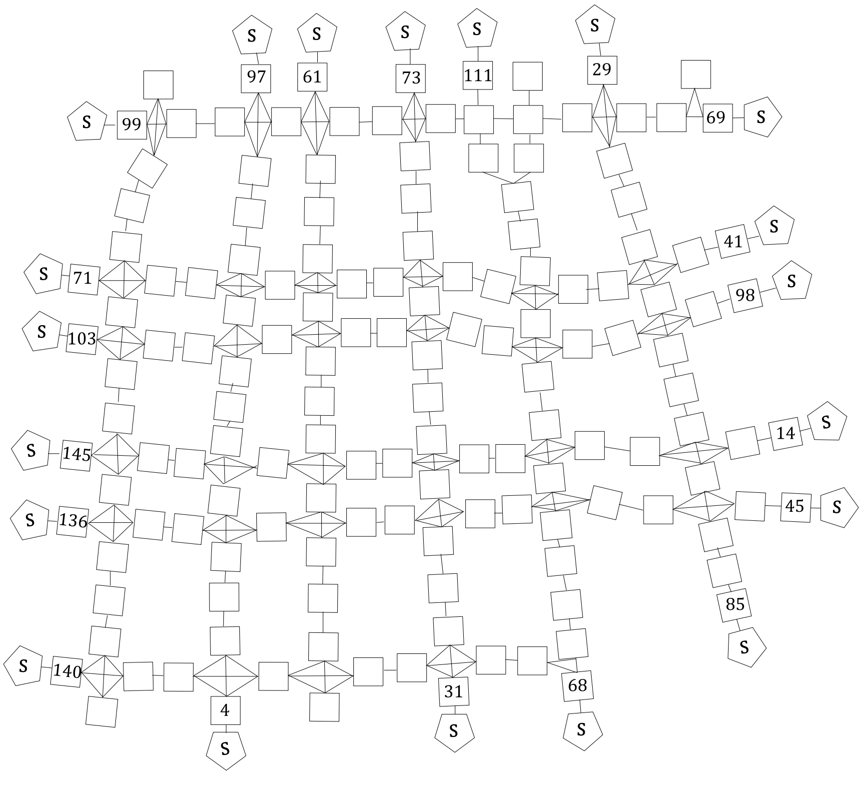

(a) Original Network.




(b) Network Transformed into Sections.

Figure A.7: Network Lima.

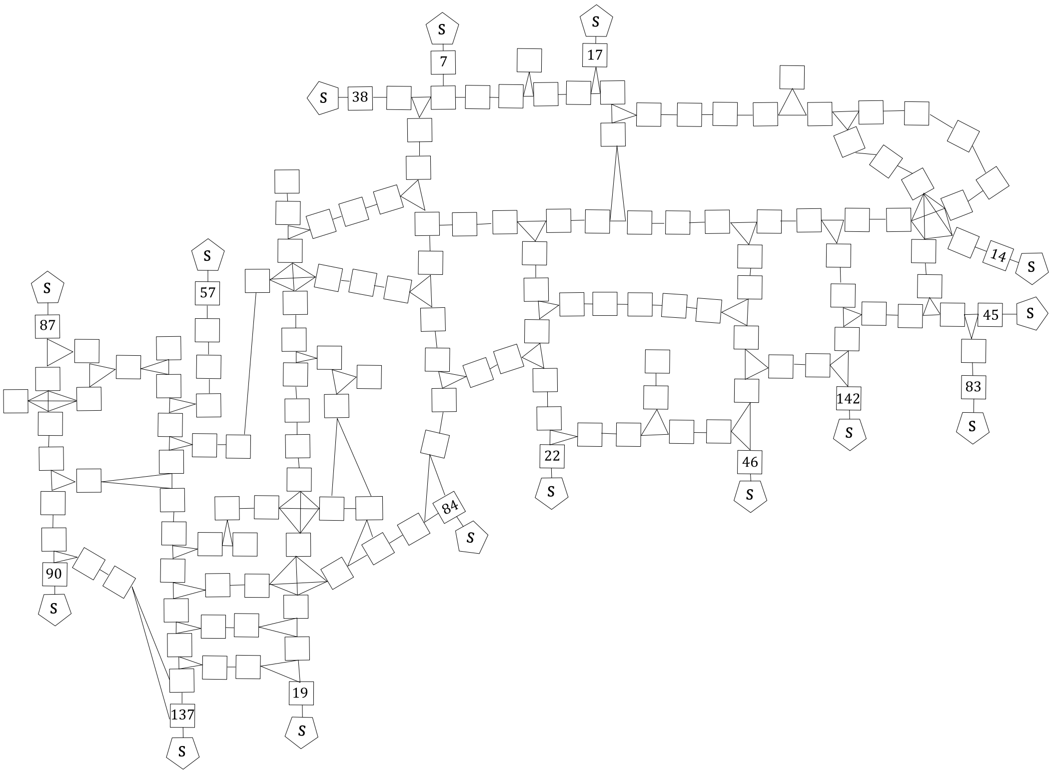
(a) Original Network.




(b) Network Transformed into Sections.

Figure A.8: Network New York.


(a) Original Network.




(b) Network Transformed into Sections.

Figure A.9: Network Dubai.

# Bibliography

Abdelgawad, H. and Abdulhai, B., (2013), Emergency evacuation planning as a network design problem: a critical review, *Transportation Letters*, **1** (1), pp. 41–58.

Altay, N. and Green, W. G., (2006), OR/MS research in disaster operations management, *European Journal of Operational Research*, **175** (1), pp. 475–493.

Askoura, Y., Lebacque, J. P., and Haj-Salem, H., (2011), Optimal sub-networks in traffic assignment problem and the Braess paradox, *Computers & Industrial Engineering*, **61** (2), pp. 382–390.

Bagloee, S. A., Ceder, A., Tavana, M., and Bozic, C., (2013), A heuristic methodology to tackle the Braess Paradox detecting problem tailored for real road networks, *Transportmetrica A: Transport Science*, **10** (5), pp. 437–456.

Bard, J. F., (2011), *Practical bilevel optimization: Algorithms and applications*, *Nonconvex optimization and its applications*, vol. 30, Springer, New York and London.

Bazzan, A. L. and Klügl, F., (2005), Case studies on the Braess Paradox: Simulating route recommendation and learning in abstract and microscopic models, *Transportation Research Part C: Emerging Technologies*, **13** (4), pp. 299–319.

Bish, D. R., Sherali, H. D., and Hobeika, A. G., (2014), Optimal evacuation planning using staging and routing, *Journal of the Operational Research Society*, **65** (1), pp. 124–140.

Bonifaci, V., Harks, T., and Schäfer, G., (2010), Stackelberg Routing in Arbitrary Networks, *Mathematics of Operations Research*, **35** (2), pp. 330–346.

Bracken, J. and McGill, J. T., (1973), Mathematical programs with optimization problems in the constraints, *Operations Research*, **21** (1), pp. 37–44.

Braess, D., (1968), Über ein Paradoxon aus der Verkehrsplanung, *Unternehmensforschung*, **12** (1), pp. 258–268.

Bretschneider, S. and Kimms, A., (2011), A basic mathematical model for evacuation problems in urban areas, *Transportation Research Part A: Policy and Practice*, **45** (6), pp. 523–539.

Bretschneider, S. and Kimms, A., (2012), Pattern-based evacuation planning for urban areas, *European Journal of Operational Research*, **216** (1), pp. 57–69.

Candler, W. and Norton, R. D., (1977), Multilevel Programming and Development Policy, World Bank Staff.

Chalmet, L. G., Francis, R. L., and Saunders, P. B., (1982), Network models for building evacuation, *Fire Technology*, **18** (1), pp. 90–113.

Chen, X., Meaker, J. W., and Zhan, F. B., (2006), Agent-Based Modeling and Analysis of Hurricane Evacuation Procedures for the Florida Keys, *Natural Hazards*, **38** (3), pp. 321–338.

Chen, X. and Zhan, F. B., (2008), Agent-based modelling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies, *Journal of the Operational Research Society*, **59** (1), pp. 25–33.

Chiu, Y.-C. and Zheng, H., (2007), Real-time mobilization decisions for multi-priority emergency response resources and evacuation groups: Model formulation and solution, *Transportation Research Part E: Logistics and Transportation Review*, **43** (6), pp. 710–736.

Choi, W., Hamacher, H. W., and Tufekci, S., (1988), Modeling of building evacuation problems by network flows with side constraints, *European Journal of Operational Research*, **35** (1), pp. 98–110.

Christodoulou, G., Mehlhorn, K., and Pyrga, E., (2014), Improving the Price of Anarchy for Selfish Routing via Coordination Mechanisms, *Algorithmica*, **69** (3), pp. 619–640.

Coello Coello, C. A., (1999), A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information systems*, **1** (3), pp. 129–156.

Cole, R., Dodis, Y., and Roughgarden, T., (2003), How much can taxes help selfish routing?, in: Menasce, D. and Nisan, N., eds., *the 4th ACM conference*, pp. 98–107.

Colson, B., Marcotte, P., and Savard, G., (2007), An overview of bilevel optimization, *Annals of Operations Research*, **153** (1), pp. 235–256.

Correa, J. R., Schulz, A. S., and Stier-Moses, N. E., (2004), Selfish Routing in Capacitated Networks, *Mathematics of Operations Research*, **29** (4), pp. 961–976.

Coutinho-Rodrigues, J., Tralhão, L., and Alçada-Almeida, L., (2012), Solving a location-routing problem with a multiobjective approach: the design of urban evacuation plans, *Journal of Transport Geography*, **22**, pp. 206–218.

Cova, T. J. and Johnson, J. P., (2002), Microsimulation of neighborhood evacuations in the urban - wildland interface, *Environment and Planning A*, **34** (12), pp. 2211–2229.

Cova, T. J. and Johnson, J. P., (2003), A network flow model for lane-based evacuation routing, *Transportation Research Part A: Policy and Practice*, **37** (7), pp. 579–604.

Dafermos, S. and Nagurney, A., (1984), On some traffic equilibrium theory paradoxes, *Transportation Research Part B: Methodological*, **18** (2), pp. 101–110.

Daganzo, C. F., (1994), The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory, *Transportation Research Part B: Methodological*, **28** (4), pp. 269–287.

Daganzo, C. F., (1995), The cell transmission model, part II: network traffic, *Transportation Research Part B: Methodological*, **29** (2), pp. 79–93.

Dash, N. and Gladwin, H., (2007), Evacuation decision making and behavioral responses: Individual and household, *Natural Hazards Review*, **8** (3), pp. 69–77.

Dijkstra, E. W., (1959), A note on two problems in connexion with graphs, *Numerische mathematik*, **1** (1), pp. 269–271.

Drabek, T. E., (1999), Understanding disaster warning responses, *The Social Science Journal*, **36** (3), pp. 515–523.

EM-DAT, (2017), Disaster Trends. URL: http://www.emdat.be/disaster_trends/index.html. Access: 16.01.2017.

Ferrante, A. and Parente, M., (2008), Mixed Nash equilibria in selfish routing problems with dynamic constraints, *Theoretical Computer Science*, **393** (1-3), pp. 37–53.

Frank, M., (1981), The Braess paradox, *Mathematical Programming*, **20** (1), pp. 283–302.

Friedman, E. J., (2004), Genericity and congestion control in selfish routing, in: *43rd IEEE Conference on Decision and Control (CDC)*, pp. 4667–4672 Vol.5.

Georgiou, C., Pavlides, T., and Philippou, A., (2006), Network uncertainty in selfish routing, in: *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, p. 10 pp.

Goerigk, M., Deghdak, K., and Heßler, P., (2014), A comprehensive evacuation planning model and genetic solution algorithm, *Transportation Research Part E: Logistics and Transportation Review*, **71**, pp. 82–97.

Gwynne, S., Galea, E. R., Owen, M., Lawrence, P. J., and Filippidis, L., (1999), A review of the methodologies used in the computer simulation of evacuation from the built environment, *Building and Environment*, **34** (6), pp. 741–749.

Hadas, Y. and Laor, A., (2013), Network design model with evacuation constraints, *Transportation Research Part A: Policy and Practice*, **47**, pp. 1–9.

Hamacher, H. W., Heller, S., and Rupp, B., (2013), Flow location (FlowLoc) problems: dynamic network flows and location models for evacuation planning, *Annals of Operations Research*, **207** (1), pp. 161–180.

Hamacher, H. W. and Tjandra, S. A., (2001), *Mathematical modelling of evacuation problems: A state of art*, Fraunhofer-Institut für Techno-und Wirtschaftsmathematik, Fraunhofer (ITWM).

Handford, D. and Rogers, A., (2012), An agent-based social forces model for driver evacuation behaviours, *Progress in Artificial Intelligence*, **1** (2), pp. 173–181.

He, X. and Peeta, S., (2014), Dynamic Resource Allocation Problem for Transportation Network Evacuation, *Networks and Spatial Economics*, **14** (3-4), pp. 505–530.

Hoefer, M. and Souza, A., (2010), Tradeoffs and Average-Case Equilibria in Selfish Routing, *ACM Transactions on Computation Theory*, **2** (1), pp. 1–25.

Hsu, F.-T. and Su, H.-J., (2011), Random Access Game in Fading Channels With Capture: Equilibria and Braess-like Paradoxes, *IEEE Transactions on Signal Processing*, **59** (3), pp. 1158–1169.

Hsu, Y.-T. and Peeta, S., (2014), Risk-based spatial zone determination problem for stage-based evacuation operations, *Transportation Research Part C: Emerging Technologies*, **41**, pp. 73–89.

Huang, X., Ozdaglar, A. E., and Acemoglu, D., (2006), Efficiency and Braess' Paradox under pricing in general networks, *IEEE Journal on Selected Areas in Communications*, **24** (5), pp. 977–991.

Huibregtse, O., Hegyi, A., and Hoogendoorn, S., (2012), Blocking roads to increase the evacuation efficiency, *Journal of Advanced Transportation*, **46** (3), pp. 282–289.

ISO22315, (2014), Societal security - Mass evacuation - Guidelines for planning.

Kalafatas, G. and Peeta, S., (2006), Network Design for Evacuation Planning, *IFAC Proceedings Volumes*, **39** (12), pp. 199–203.

Kalafatas, G. and Peeta, S., (2009), Planning for evacuation: insights from an efficient network design model, *Journal of infrastructure systems*, **15** (1), pp. 21–30.

Karakostas, G., Kim, T., Viglas, A., and Xia, H., (2007), Selfish Routing with Oblivious Users, in: Prencipe, G. and Zaks, S., eds., *Structural Information and Communication Complexity*, *Lecture Notes in Computer Science*, vol. 4474, Springer Berlin Heidelberg, Berlin and Heidelberg, pp. 318–327.

Karakostas, G. and Kolliopoulos, S. G., (2009), Stackelberg Strategies for Selfish Routing in General Multicommodity Networks, *Algorithmica*, **53** (1), pp. 132–153.

Kimms, A. and Maassen, K.-C., (2011a), Extended cell-transmission-based evacuation planning in urban areas, *Pesquisa Operacional*, **31** (3), pp. 405–441.

Kimms, A. and Maassen, K.-C., (2011b), Optimization and simulation of traffic flows in the case of evacuating urban areas, *OR Spectrum*, **33** (3), pp. 571–593.

Kimms, A. and Maassen, K.-C., (2012a), Cell-transmission-based evacuation planning with rescue teams, *Journal of Heuristics*, **18** (3), pp. 435–471.

Kimms, A. and Maassen, K.-C., (2012b), A fast heuristic approach for large-scale cell-transmission-based evacuation route planning, *Networks*, **60** (3), pp. 179–193.

Kimms, A. and Maiwald, M., (2015a), Bi-objective safe and resilient urban evacuation planning, *Working Paper MSM, University of Duisburg-Essen.*

Kimms, A. and Maiwald, M., (2015b), An exact network flow formulation for cell-based evacuation in urban areas, *Working Paper MSM, University of Duisburg-Essen.*

Kimms, A. and Seekircher, K., (2015), Network design to anticipate selfish evacuation routing, *EURO Journal on Computational Optimization*, (4), pp. 1–28.

Kimms, A. and Seekircher, K., (2016), A Bi-level Formulation of a Network Design Problem to Anticipate Selfish Evacuation Routing, *Working Paper MSM, University of Duisburg-Essen.*

Kimms, A. and Seekircher, K., (2017), Guiding Selfish Evacuees with a Minimum Number of Street Blockages, *Working Paper MSM, University of Duisburg-Essen.*

Knight, V. A. and Harper, P. R., (2013), Selfish routing in public services, *European Journal of Operational Research*, **230** (1), pp. 122–132.

Kongsomsaksakul, S., Chao, Y., and Anthony, C., (2005), Shelter location-allocation model for flood evacuation planning, *Journal of the Eastern Asia Society for Transportation Studies*, **6**, pp. 4237–4252.

Korilis, Y. A., Lazar, A. A., and Orda, A., (1999), Avoiding the Braess paradox in non-cooperative networks, *Journal of Applied Probability*, **36** (1), pp. 211–222.

Koutsoupias, E. and Papadimitriou, C., (1999), Worst-Case Equilibria, in: Goos, G., Hartmanis, J., van Leeuwen, J., Meinel, C., and Tison, S., eds., *STACS 99*, *Lecture Notes in Computer Science*, vol. 1563, Springer Berlin Heidelberg, Berlin and Heidelberg, pp. 404–413.

Kulshrestha, A., Di Wu, Lou, Y., and Yin, Y., (2011), Robust Shelter Locations for Evacuation Planning with Demand Uncertainty, *Journal of Transportation Safety & Security*, **3** (4), pp. 272–288.

Li, L., Jin, M., and Zhang, L., (2011), Sheltering network planning and management with a case in the Gulf Coast region, *International Journal of Production Economics*, **131** (2), pp. 431–440.

Lighthill, M. J. and Whitham, G. B., (1955), On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **229** (1178), pp. 317–345.

Lim, G. J., Rungta, M., and Baharnemati, M. R., (2014), Reliability analysis of evacuation routes under capacity uncertainty of road links, *IIE Transactions*, **47** (1), pp. 50–63.

Lim, G. J., Zangeneh, S., Reza Baharnemati, M., and Assavapokee, T., (2012), A capacitated network flow optimization approach for short notice evacuation planning, *European Journal of Operational Research*, **223** (1), pp. 234–245.

Lindell, M. K. and Perry, R. W., (2012), The Protective Action Decision Model: Theoretical Modifications and Additional Evidence, *Risk Analysis*, **32** (4), pp. 616–632.

Lindell, M. K. and Prater, C. S., (2007), Critical Behavioral Assumptions in Evacuation Time Estimate Analysis for Private Vehicles: Examples from Hurricane Research and Planning, *Journal of Urban Planning and Development*, **133** (1), pp. 18–29.

Liu, Y., Lai, X., and Chang, G.-L., (2006), Cell-Based Network Optimization Model for Staged Evacuation Planning Under Emergencies, *Transportation Research Record: Journal of the Transportation Research Board*, **1964**, pp. 127–135.

Madireddy, M., Medeiros, D. J., and Kumara, S., (2011), An agent based model for evacuation traffic management, in: *2011 Winter Simulation Conference - (WSC 2011)*, pp. 222–233.

Marler, R. T. and Arora, J. S., (2004), Survey of multi-objective optimization methods for engineering, *Structural and Multidisciplinary Optimization*, **26** (6), pp. 369–395.

Miller-Hooks, E. and Sorrel, G., (2008), Maximal Dynamic Expected Flows Problem for Emergency Evacuation Planning, *Transportation Research Record: Journal of the Transportation Research Board*, **2089**, pp. 26–34.

Müller, G., (1998), Zivielschutzforschung: Kriterien für Evakuierungsempfehlungen bei Chemikalienfreisetzungen, *Herausgegeben vom Bundesamt für Zivilschutz*, (32).

Murchland, J. D., (1970), Braess's paradox of traffic flow, *Transportation Research*, **4** (4), pp. 391–394.

Murray-Tuite, P. and Mahmassani, H., (2004), Transportation network evacuation planning with household activity interactions, *Transportation Research Record: Journal of the Transportation Research Board*, (1894), pp. 150–159.

Murray-Tuite, P. and Wolshon, B., (2013), Evacuation transportation modeling: An overview of research, development, and practice, *Transportation Research Part C: Emerging Technologies*, **27**, pp. 25–45.

Murray-Tuite, P., Yin, W., Ukkusuri, S., and Gladwin, H., (2012), Changes in Evacuation Decisions Between Hurricanes Ivan and Katrina, *Transportation Research Record: Journal of the Transportation Research Board*, **2312**, pp. 98–107.

Nagurney, A., (2010), The negation of the Braess paradox as demand increases: The wisdom of crowds in transportation networks, *EPL (Europhysics Letters)*, **91** (4), p. 48002.

Nagurney, A., Parkes, D., and Daniele, P., (2007), The Internet, evolutionary variational inequalities, and the time-dependent Braess paradox, *Computational Management Science*, **4** (4), pp. 355–375.

Ng, M. and Waller, S. T., (2010), Reliable evacuation planning via demand inflation and supply deflation, *Transportation Research Part E: Logistics and Transportation Review*, **46** (6), pp. 1086–1094.

Özdamar, L. and Ertem, M. A., (2015), Models, solutions and enabling technologies in humanitarian logistics, *European Journal of Operational Research*, **244** (1), pp. 55–65.

Pan, X., Han, C. S., Dauber, K., and Law, K. H., (2007), A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations, *Ai & Society*, **22** (2), pp. 113–132.

Papadimitriou, C., (2001), Algorithms, games, and the internet, in: Vitter, J. S., Spirakis, P., and Yannakakis, M., eds., *the thirty-third annual ACM symposium*, pp. 749–753.

Pas, E. I. and Principio, S. L., (1997), Braess' paradox: Some new insights, *Transportation Research Part B: Methodological*, **31** (3), pp. 265–276.

Pel, A. J., Bliemer, Michiel C. J., and Hoogendoorn, S. P., (2012), A review on travel behaviour modelling in dynamic traffic simulation models for evacuations, *Transportation*, **39** (1), pp. 97–123.

Pidd, M., Silva, F. N. d., and Eglese, R. W., (1996), A simulation model for emergency evacuation, *European Journal of Operational Research*, **90** (3), pp. 413–419.

Pigou, A. C., (1920), The Economics of Welfare, 4th, *London: Macnillam*.

Polimeni, A. and Vitetta, A., (2011a), Dynamic vehicle routing in road evacuation: a model for route design, *WIT Transactions on the Built Environment*, **116**, pp. 627–638.

Polimeni, A. and Vitetta, A., (2011b), Dynamic vehicle routing in road evacuation: route design experimentation, in: Guarascio, M., Reniers, G., Brebbia, C. A., and Garzia, F., eds., *SAFE 2011*, WIT Transactions on the Built Environment, WIT PressSouthampton, UK, pp. 391–402.

Rapoport, A., Gisches, E. J., and Mak, V., (2014), Distributed Decisions in Networks: Laboratory Study of Routing Splittable Flow, *Production and Operations Management*, **23** (2), pp. 314–331.

Rapoport, A., Kugler, T., Dugar, S., and Gisches, E. J., (2009), Choice of routes in congested traffic networks: Experimental tests of the Braess Paradox, *Games and Economic Behavior*, **65** (2), pp. 538–571.

Richards, P. I., (1956), Shock Waves on the Highway, *Operations Research*, **4** (1), pp. 42–51.

Roughgarden, T., (2003), The price of anarchy is independent of the network topology, *Journal of Computer and System Sciences*, **67** (2), pp. 341–364.

Roughgarden, T., (2004), Stackelberg Scheduling Strategies, *SIAM Journal on Computing*, **33** (2), pp. 332–350.

Roughgarden, T., (2005), *Selfish routing and the price of anarchy*, MIT Press, Cambridge and Mass.

Roughgarden, T., (2006), On the severity of Braess's Paradox: Designing networks for selfish users is hard, *Journal of Computer and System Sciences*, **72** (5), pp. 922–953.

Roughgarden, T. and Tardos, É., (2002), How bad is selfish routing?, *Journal of the ACM (JACM)*, **49** (2), pp. 236–259.

Saadatseresht, M., Mansourian, A., and Taleai, M., (2009), Evacuation planning using multiobjective evolutionary optimization approach, *European Journal of Operational Research*, **198** (1), pp. 305–314.

Sadri, A. M., Ukkusuri, S. V., Murray-Tuite, P., and Gladwin, H., (2014), How to Evacuate: Model for Understanding the Routing Strategies during Hurricane Evacuation, *Journal of Transportation Engineering*, **140** (1), pp. 61–69.

Sangho, K., Shekhar, S., and Min, M., (2008), Contraflow Transportation Network Reconfiguration for Evacuation Route Planning, *IEEE Transactions on Knowledge and Data Engineering*, **20** (8), pp. 1115–1129.

Sheffi, Y., Mahmassani, H., and Powell, W. B., (1982), A transportation network evacuation model, *Transportation research part A: general*, **16** (3), pp. 209–218.

Shen, T.-S., (2005), ESM: a building evacuation simulation model, *Building and Environment*, **40** (5), pp. 671–680.

Shen, W., Nie, Y., and Zhang, H., (2007), Dynamic Network Simplex Method for Designing Emergency Evacuation Plans, *Transportation Research Record: Journal of the Transportation Research Board*, **2022**, pp. 83–93.

Sherali, H. D., Carter, T. B., and Hobeika, A. G., (1991), A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions, *Transportation Research Part B: Methodological*, **25** (6), pp. 439–452.

Sinuany-Stern, Z. and Stern, E., (1993), Simulating the evacuation of a small city: the effects of traffic factors, *Socio-Economic Planning Sciences*, **27** (2), pp. 97–108.

Sorensen, J. H., (2000), Hazard warning systems: Review of 20 years of progress, *Natural Hazards Review*, **1** (2), pp. 119–125.

SPSE, (2016), SPSE-Software zur Planung von städtischen Evakuierungen. University of Duisburg-Essen. URL: https://www.log.msm.uni-due.de/forschung/software-zur-planung-von-staedtischen-evakuierungen/. Access: 02.12.2016.

Stackelberg, H. v., (1952), *The theory of the market economy*, Oxford University Press.

Steinberg, R. and Zangwill, W. I., (1983), The prevalence of Braess' paradox, *Transportation Science*, **17** (3), pp. 301–318.

Stepanov, A. and Smith, J. M., (2009), Multi-objective evacuation routing in transportation networks, *European Journal of Operational Research*, **198** (2), pp. 435–446.

Talbi, E.-G., (2013), *Metaheuristics for bi-level optimization*, *Studies in computational intelligence*, vol. 482, Springer, Berlin and New York.

Tu, H., Tamminga, G., Drolenga, H., Wit, J. d., and van der Berg, Wouter, (2010), Evacuation plan of the city of almere: simulating the impact of driving behavior on evacuation clearance time, *Procedia Engineering*, **3**, pp. 67–75.

Tuydes, H. and Ziliaskopoulos, A., (2006), Tabu-Based Heuristic Approach for Optimization of Network Evacuation Contraflow, *Transportation Research Record: Journal of the Transportation Research Board*, **1964**, pp. 157–168.

Tuydes-Yaman, H. and Ziliaskopoulos, A., (2014), Modeling demand management strategies for evacuations, *Annals of Operations Research*, **217** (1), pp. 491–512.

Valiant, G. and Roughgarden, T., (2010), Braess's Paradox in large random graphs, *Random Structures & Algorithms*, **37** (4), pp. 495–515.

Van Wassenhove, L. N. and Pedraza Martinez, A. J., (2012), Using OR to adapt supply chain management best practices to humanitarian logistics, *International Transactions in Operational Research*, **19** (1-2), pp. 307–322.

Wardrop, J. G., (1952), Road Paper. Some Theoretical Aspects of Road Traffic Research, *Proceedings of the institution of civil engineers*, **1** (3), pp. 325–362.

Wilmot, C. and Meduri, N., (2005), Methodology to establish hurricane evacuation zones, *Transportation Research Record: Journal of the Transportation Research Board*, (1922), pp. 129–137.

Wolshon, B., (2001), “one-way-out”: contraflow freeway operation for hurricane evacuation, *Natural Hazards Review*, **2** (3), pp. 105–112.

Xiao, J., Xiong, N., Yang, L. T., and He, Y., (2008), A joint selfish routing and channel assignment game in wireless mesh networks, *Computer Communications*, **31** (7), pp. 1447–1459.

Xie, C., Lin, D.-Y., and Travis Waller, S., (2010), A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies, *Transportation Research Part E: Logistics and Transportation Review*, **46** (3), pp. 295–316.

Xie, C. and Turnquist, M. A., (2011), Lane-based evacuation network optimization: An integrated Lagrangian relaxation and tabu search approach, *Transportation Research Part C: Emerging Technologies*, **19** (1), pp. 40–63.

Yamada, T., (1996), A network flow approach to a city emergency evacuation planning, *International Journal of Systems Science*, **27** (10), pp. 931–936.

Yazici, A. and Ozbay, K., (2010), Evacuation Network Modeling via Dynamic Traffic Assignment with Probabilistic Demand and Capacity Constraints, *Transportation Research Record: Journal of the Transportation Research Board*, **2196**, pp. 11–20.

Yuan, F., Han, L., Chin, S.-M., and Hwang, H., (2006), Proposed framework for simultaneous optimization of evacuation traffic destination and route assignment, *Transportation Research Record: Journal of the Transportation Research Board*, (1964), pp. 50–58.

Ziliaskopoulos, A. K., (2000), A linear programming model for the single destination system optimum dynamic traffic assignment problem, *Transportation Science*, **34** (1), pp. 37–49.

Zou, N., Yeh, S.-T., Chang, G.-L., Marquess, A., and Zezeski, M., (2005), Simulation-based emergency evacuation system for Ocean City, Maryland, during hurricanes, *Transportation Research Record: Journal of the Transportation Research Board*, (1922), pp. 138–148.