KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

Design of integrated scheduling and simulation models to optimise people flows and maximise safety

Dissertation presented to obtain the degree of Doctor in Business Economics

by

Hendrik VERMUYTEN

Number 646 2019

FACULTEIT ECONOMIE EN
BEDRIJFSWETENSCHAPPEN

KU LEUVEN

Design of integrated scheduling and simulation models to optimise people flows and maximise safety

Proefschrift voorgedragen tot het behalen van de graad van Doctor in de Toegepaste Economische Wetenschappen

door

Hendrik VERMUYTEN

Nummer 646

2019

Doctoral committee

Chairperson:	Prof. dr. Ingrid Van Keilegom *KU Leuven*
Advisor:	Prof. dr. Jeroen Beliën *KU Leuven - Campus Brussels*
Co-advisors:	Prof. dr. Liesje De Boeck *KU Leuven - Campus Brussels* Prof. dr. ir. Genserik Reniers *KU Leuven - Campus Brussels* Prof. dr. Tony Wauters *KU Leuven - Campus Ghent*
Members:	Prof. dr. Frits Spieksma *Eindhoven University of Technology* Prof. dr. Inneke Van Nieuwenhuyse *Hasselt University* Prof. dr. ir. Winnie Daamen *Delft University of Technology*

Acknowledgements

This thesis is the end result of a four-year journey. What seemed like an incredibly long time at the start has gone by so fast. I have learned a lot of new things and had the chance to meet many interesting people. Therefore, I would like to thank everyone involved.

Jeroen, I would like to thank you for giving me the opportunity to work on this research project. I could not have imagined a better supervisor. During my first doctoral course, 'Combinatorial Optimization and Local Search Techniques', I immediately got to know you as a knowledgeable and passionate teacher. I really enjoyed the lectures, especially trying to beat the algorithms of the professor. From the start of this project, you allowed me to to work on problems I found interesting and let me experiment freely. However, whenever I had questions, you immediately took the time to help me out. Also, whenever I would send a preliminary version of a paper, you always provided me with detailed feedback very quickly. You also offered me many learning opportunities: my teaching experiences for the course Applications of Operations Research and my research stay in Lisbon. Finally, I really enjoyed the many conferences with you, Dries, and Stefan. The road trip with the three of you to the Grand Canyon and Las Vegas is absolutely one of the highlights of the last four years.

Liesje, I would like to thank you for thoroughly reviewing every chapter in this thesis. With your attention to detail, you have made many improvements to the writing. You also questioned various modelling assumptions that we took for granted. I always enjoyed your constructive criticism and positive attitude.

Genserik, I would like to thank you for interesting questions and remarks from a safety point of view. It is unfortunate that in the end I did not have the chance to work on the Petri nets approach for the analysis of emergency situations. It would certainly be an interesting idea for future research.

Tony, you gave many helpful ideas and suggestions for the metaheuristic approaches in this thesis. You always ask important questions that are often ignored, such as why one approach is chosen above another approach or why the given values were selected for the parameters of a model. These questions force you to question things you took for granted and help improve the quality of the algorithms.

I would like to thank the other members of the doctoral committee. Frits, you provided many thought-provoking questions and constructive feedback. Inneke, your helpful suggestions and your experience with simulation-optimisation helped to improve this text. I also want to thank you for introducing me to Jeroen. Without you, I would not have had the chance to work on this doctoral thesis. Professor Winnie Daamen, I would like to thank you for agreeing to be a member of this doctoral committee. As most jury members and myself have an operations research background, we approached the problems in this thesis from that perspective. Therefore, your expertise on pedestrian and traffic simulation models has been very valuable to offer a different point of view. Based on your questions and suggestions, a large part of this thesis has been improved. Finally, I would like to thank professor Ingrid Van Keilegom for agreeing to be the chairperson of the jury.

I also want to thank professor Inês Marques and Stef Lemmens for contributing to my first publication. You already provided the ground work and gave much help during the publication process. Inês, I would like to thank you further for inviting me to work with you in Lisbon on an interesting real-life personnel scheduling problem. I really enjoyed my stay in Lisbon and working with you, professor Ana Barbosa-Póvoa, and Joana Namorado Rosa. You also showed sincere interest in the problems I was working on for this thesis.

I wish to thank Jens for being not only a good colleague but also a friend. We could always discuss our research and vent our frustrations to each other when our C++ codes would give yet another error. I always enjoyed it when you accompanied us on conferences, such as the yearly ORBel conference and the INFORMS Annual Meeting in Philadelphia.

A final word of thanks goes to my family and friends. To my parents, I am sure that you are as proud of this doctoral thesis as I am. I was lucky to grow up in a loving family that nurtured my interest in science since I was a young boy. You offered me the freedom to study whatever I wanted and always supported me over the course of this PhD. Lisbeth, I want to thank you for the many beautiful memories of the last three years. I hope there are many more years to come. Your daily support has helped me tremendously in writing this thesis. You always showed interest in the things I was working on. Furthermore, whenever I was working during evenings or weekends to meet a deadline, you would surprise me with small gifts, such as an encouraging message or a beautifully presented plate of fruit.

Hendrik Vermuyten
April 2019

A little learning is a dang'rous thing;
Drink deep, or taste not the Pierian spring:
There shallow draughts intoxicate the brain,
And drinking largely sobers us again.

Alexander Pope, *An Essay on Criticism*

Abstract

The study of pedestrian walking behaviour and crowd dynamics is an important topic with many applications. Many computationally efficient but less accurate macroscopic models (where the crowd is described as a whole using average quantities such as density and velocity at a certain location and time) and computationally less efficient but more accurate microscopic models (where each pedestrian is described as a separate entity) have been developed to describe the walking behaviour of individual pedestrians and large crowds. Moreover, many optimisation models have been proposed to solve problems involving pedestrians. Most of these models have hitherto focused on evacuation problems, where for a given building and an initial distribution of pedestrians, the optimal evacuation plan for each pedestrian is computed. A few researchers have also focused on design problems to find the optimal layout of a pedestrian facility or bottleneck passages.

In this thesis, we consider the link between timetabling problems and crowd flow optimisation. Indeed, the assignment of events to timeslots and rooms in a timetable has an impact on the resulting people flows. For example, in a university course timetable, at the end of each timeslot, students have to switch rooms to go to their next class. This can cause congestion problems in the halls and stairwells in universities where the class rooms are concentrated in one or a few buildings. Furthermore, if the building needs to be evacuated at a certain time, this evacuation process is also influenced by the university course timetable. After all, the timetable determines how many people are present in the building in each timeslot and in which rooms. University course timetables are not the only example. The timetable at large conferences, music festivals, cultural events, or sports events also determines the people flows during the event.

This thesis consists of four main parts. Chapter 2 presents a review of optimisation models for pedestrian evacuation and design problems. Relevant empirical research and descriptive (mathematical) models of pedestrian walking behaviour are also discussed. This review shows that most models include the inverse relationship between density and walking speed, but that calibration and implementation of the proposed models are lacking.

Chapter 3 focuses on the university course timetabling problem (UCTP) at KU Leuven Campus Brussels. A two-stage mixed-integer programming (MIP) approach is developed to build a timetable that maximises the scheduling preferences and minimises the travel times of students between lectures in consecutive timeslots. Pedestrians are modelled using a macroscopic network model with a density-dependent arc traverse time. The model is extended to optimise evacuations time of students in the event of an emergency. Computational results show that the model succeeds in constructing timetables with reduced travel or egress times. However, the model fails to solve large real-life instances, such as the KU Leuven instance. Therefore, a heuristic approach is developed to solve the problem. In contrast to the two-stage MIP approach, the heuristic is able to find good quality solutions for the KU Leuven instance. Moreover, it succeeds in improving upon the solutions found by the two-stage MIP approach for all other test instances.

In Chapter 4, we develop a generic, flexible model for timetabling incorporating resulting people flows. To keep the model generic and tractable, only the assignment of events to rooms is optimised, while the assignment of events to timeslots is considered given. The pedestrian walking behaviour and crowd dynamics are described using the microscopic pedestrian simulator Menge developed by Curtis et al. (2016). A surrogate-based tabu search heuristic is proposed to solve the problem. The surrogate model is used to speed up the search by filtering the number of candidate solutions that are evaluated using the expensive Menge simulator. The performance of different surrogate models is evaluated. The model is used to solve two applications, one where we minimise evacuation times and one where we minimise travel times between events in consecutive timeslots. The results show that for both applications the model succeeds in building timetables with significantly reduced travel or egress times. Finally, the model is implemented in a scheduling tool with graphical user interface and applied to the room assignment problem at KU Leuven Campus Brussels. It shows that our model is able to tackle real-life problem instances with large numbers of pedestrians.

Finally, Chapter 5 compares the network model of Chapter 3 and the microscopic Menge simulator of Chapter 4. Using exhaustive search on small problem instances, the quality assigned to each solution in the solution space by the different models is compared in detail. Moreover, the modelling power and the robustness of the models with respect to the calibration of their parameters is discussed.

Contents

Chapter 1

Introduction

1.1 Motivation

On 22 May 1967, a fire took place at the Innovation store in Brussels, resulting in the deaths of 251 people. Contributing to the high death toll were inadequate fire extinguishers, a complex building layout, and a lack of emergency exists. On 29 May 1985, one of the largest disasters in football history occurred at the Heysel stadium in Brussels during the European Cup Final between Liverpool and Juventus. Liverpool fans charged at Juventus fans who were standing in an adjacent 'neutral section'. In the resulting turmoil, people were pressed against a collapsing wall and trampled. 39 people were killed and hundreds were injured in the confrontation. More recently, 21 people died at the 2010 Love Parade dance festival in Duisburg. The location of the festival had a capacity of only 250,000 people, while around 400,000 people had arrived in Duisburg. This resulted in large flows of people pushing into each other, effectively crushing people to death.

Events such as these have inspired a great deal of research into pedestrian walking behaviour and crowd dynamics (see, e.g., Helbing and Johansson, 2010; Schadschneider et al., 2009). Most of this research focuses on empirical investigations of pedestrian dynamics and developing mathematical models that describe these dynamics truthfully (Vermuyten et al., 2016a). By contrast, optimisation models for pedestrian evacuation and design problems are less well-researched. More specifically, most optimisation models are concerned with the derivation of optimal evacuation plans for pedestrian facilities in the event of an emergency. There are, however, other interesting problems, such as design problems and crowd management under normal conditions that have not yet been studied in detail (Vermuyten et al., 2016a).

In this thesis, we are interested in the optimisation of people flows that are the result of scheduling decisions in a timetable. The research was inspired by congestion problems at the halls, elevators and stairwells during lecture transitions at KU Leuven Campus Brussels and the observation that the course timetable has an impact on this. For this reason, we formulate the problem starting from this context. However, the methodologies we develop are much more general and could be useful for all situations in which a timetable impacts (the magnitude of) people flows. At large conferences, the schedule of the different talks determines the flows of people who travel from one talk to the next. At music festivals, cultural events, or sports events, the schedule of music bands or sports games has a large impact on the resulting spectator flows before, during, and after the event.

To the best of our knowledge, Al-Yakoob and Sherali (2007), Al-Yakoob et al. (2010), Ferdoushi et al. (2014), Hertz (1991), Pongcharoen et al. (2008), and Rudová et al. (2011) are the only articles that incorporate people flows into timetabling problems. However, these papers do not model the people flows in much detail. Al-Yakoob and Sherali (2007) study the university course timetabling problem and address the problem of parking and traffic congestions by adequately spreading lectures over all the available timeslots by constraints that impose an upper bound on the number of students that follow classes during each timeslot. Al-Yakoob et al. (2010) instead focus on the exam timetabling problem. Parking and traffic congestions are addressed by imposing a constraint on the number of students that can be involved in one exam period. Pongcharoen et al. (2008) and Ferdoushi et al. (2014) use soft constraints to ensure that students attend lectures in the same classroom as much as possible. However, distances between classrooms are not taken into account. Hertz (1991) includes a penalty term in the objective function when pairs of consecutive lectures are scheduled at distant classrooms. Rudová et al. (2011) consider the distances between rooms and penalise class assignments that require students or instructors to travel large distances between consecutive lectures. Moreover, to the best of our knowledge, none of the approaches proposed in the literature takes into account safety aspects. Occupational and public safety regulations provide for general principles to apply, such as the overall responsibility of the employer or organiser to ensure the safety of workers, visitors, contractors and the public at large, the importance of risk analyses and a hierarchy of prevention principles to take into account, such as priority for collective, material and organisational measures (Van Heuverswyn, 2009).

This research is innovative because it is the first attempt to build an optimisation model and develop a solution procedure that explicitly links timetabling processes with the resulting people flows, analysing in a detailed way the impact of scheduling decisions on, e.g., the resulting travel time from a given origin to a given destination. Furthermore, the project is innovative because it is the first attempt (as far as we know) to incorporate safety aspects (dependent on the resulting people flows) when constructing timetables.

1.2 Scope

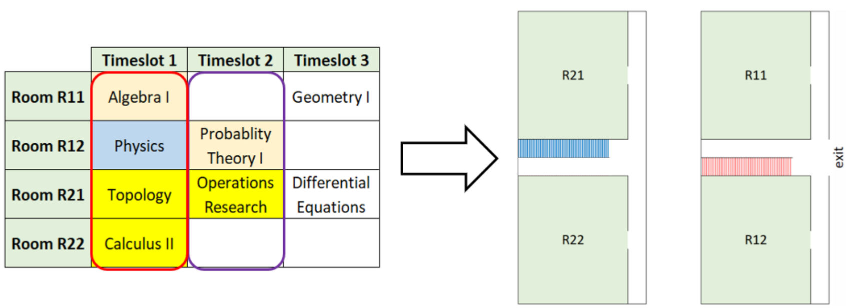
In this dissertation, we focus on developing new techniques for automated timetabling that focus on optimising the resulting people flows and maximising safety. There are many different ways, however, in which these two objectives could be quantified to compare the quality of different timetabling solutions. We limit ourselves to two objective function measures that are common in the literature (see Chapter 2) and the most relevant for our problem setting. Another possible objective function measure that could be used to rate the quality of different solutions is the maximum crowd density. However, we believe this measure is more relevant to mass crowd events such as music festivals and does not pose a problem in the timetabling problems we focus on in this thesis, such as the university course timetabling problem.

A first objective function measure concerns the *minimisation of the travel times between events in consecutive timeslots* (see Figure 1.1). For example, in a university course timetable, at the end of each timeslot, students have to switch rooms to go to their next class. This can cause congestion problems in the halls and stairwells in universities where the class rooms are concentrated in one or a few buildings. These flows occur between each pair of consecutive timeslots in the timetable. We define the *travel time* between timeslot t and timeslot $t + 1$ as the time it takes for the last person to arrive at his or her destination in timeslot $t + 1$. It is assumed that events in the same timeslot start and end at the same time and people start walking from the location of their current event to the location of their next destination immediately after the first event ends. We can then minimise either the sum or the maximum of the travel times between all pairs of consecutive timeslots. The former objective focuses on the average case, while the latter focuses on the worst case.

A second objective function measure concerns the safety aspect. We define

Figure 1.1: An example of scheduling decisions in a timetable and the resulting flows between events in consecutive timeslots.

The table on the left-hand side shows an example of a university course timetable. Events marked with the same colour are followed by the same groups of students. One group takes Algebra I in timeslot 1 and Probability Theory I in timeslot 2. Another group takes Physics in timeslot 1 and has no class in timeslot 2. Finally, multiple groups can take a certain class. This is indicated by the three yellow events. One group takes Topology in timeslot 1 and another group takes Calculus II in timeslot 1, but both take Operations Research in timeslot 2. The assignments of these events to the given rooms in timeslot 1 and 2 determines the flows of students between both timeslots. On the right-hand side a floor plan is shown. This floor plan represents a building consisting of four rooms divided over two floors. The blue stripes indicate stairs going down, while the red stripes indicate stairs going up. The students who take Algebra I and Probability Theory I will walk from room R11 in the building to room R12; the students who take Physics walk from room R12 to the exit of the building; the students who take Topology and Operations Research stay in room R21; finally, the students who take Calculus II and Operations walk from room R22 to room R21.



the *safety* of a timetable as the time required to evacuate the building in the event of an emergency. This evacuation process is also influenced by the scheduling decisions taken when building the timetable. Indeed, the timetable determines how many people are present in the building in each timeslot and in which rooms (see Figure 1.2). We define the *evacuation time* or *egress time* as the time it takes for the last person to exit the building to a safe location. This choice is the most common in optimisation models for evacuation problems, as is shown in Chapter 2. However, we could also define the egress time as the time when a certain percentage of people has reached safety. We explore these choices further in Chapter 5. We can then again minimise either the sum or the maximum of the travel times in all timeslots.

1.3 Thesis outline

An overview of the structure of this thesis is given in Figure 1.3. Chapter 2 provides a review of optimisation models for pedestrian evacuation and design problems. We first provide a classification of the different articles based on the problem type, objective function measures, and decisions that are considered. Next, we discuss relevant empirical research and the implications for modelling pedestrian walking behaviour. We also consider the inclusion of uncertainty into the models and their applicability. Finally, we compare modelling techniques used in optimisation models with techniques used in descriptive models. An article based on this chapter has been published as Vermuyten, H., Beliën, J., De Boeck, L., Reniers, G., Wauters, T. (2016). A review of optimisation models for pedestrian evacuation and design problems. *Safety Science, 87*, 167-178. doi:10.1016/j.ssci.2016.04.001.

In Chapter 3, we focus on the university course timetabling problem

Figure 1.2: An example of scheduling decisions in a timetable and the resulting flows in the event of an evacuation.

The table on the left-hand side shows an example of a university course timetable. In timeslot 2, a first group of students takes Probability Theory I and a second group takes Operations Research. The assignment of these events to the given rooms in every timeslot also determines the people flows if the building were to be evacuated in that timeslot. On the right-hand side a floor plan is shown. This floor plan represents a building consisting of four rooms divided over two floors. The blue stripes indicate stairs going down, while the red stripes indicate stairs going up. Given the assignments of events to rooms, the first group of students needs to walk from room R12 towards the exit of the building, while the second group needs to walk from room R21 towards the exit.

	Timeslot 1	**Timeslot 2**	**Timeslot 3**
Room R11	Algebra I		Geometry I
Room R12	Physics	Probablity Theory I	
Room R21	Topology	Operations Research	Differential Equations
Room R22	Calculus II		

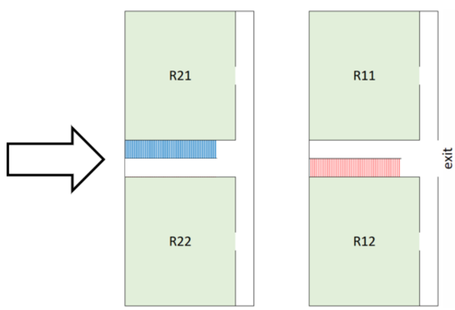
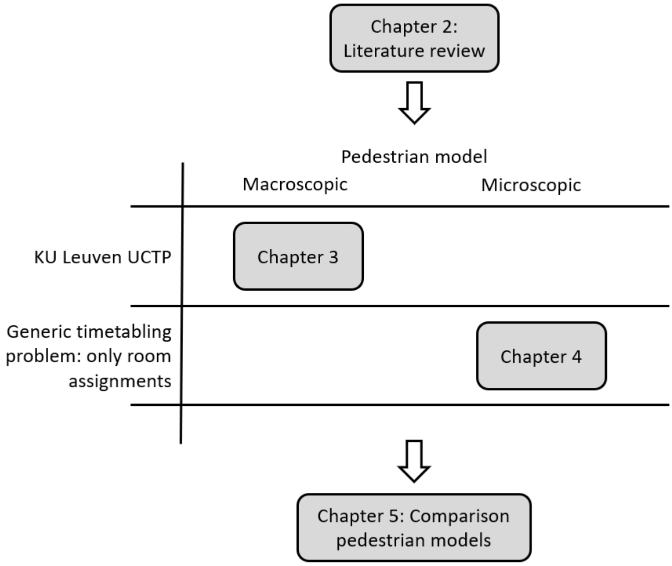


Figure 1.3: Overview of the thesis.



(UCTP) at KU Leuven Campus Brussels, where there are congestion problems in the halls and at the stairwells between lecture transitions. The flows of students between consecutive lectures are modelled using a network model. The layout of the building is represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where the set $\mathcal{N}$ of nodes represents the destinations or junctions and the set $\mathcal{A}$ of arcs the corridors between them. Congestion is taken into account by assuming density-dependent travel times through the arcs based on the relationship between crowd density and walking speed in line with empirical research into pedestrian walking behaviour. The travel times between events in consecutive timeslots are minimised. The problem is formulated as a mixed-integer programming (MIP) model and is solved using a two-stage decomposition approach. The first stage assigns lectures to timeslots and rooms and minimises the penalty score for the scheduling preferences while taking into account the standard non-overlap constraints as well as regulations for the working time of teachers and constraints for a compact timetable. The second stage starts from the solution of the first stage and reassigns lectures to rooms to minimise either the sum or the maximum of the travel times of students between consecutive lectures. We then shift our focus to the safety concern and show that the two-stage model can easily be extended to minimise the sum of the evacuation times over the different timeslots. Due to its hierarchical approach, the two-stage MIP approach can only find a single point on the Pareto front for the trade-off between the optimisation of the scheduling preferences on the one hand and the minimisation of the travel or egress times on the other hand. Moreover, the two-stage MIP approach is not able to solve the complex KU Leuven instance. For these reasons, a metaheuristic approach is developed that does not suffer from these drawbacks. Results show that the metaheuristic succeeds in finding good solutions for the KU Leuven instance and finds solutions for the set of test instances with lower evacuation times than the two-stage MIP model. An article based on parts of this chapter has been published as Vermuyten, H., Lemmens, S., Marques, I., Beliën, J. (2016). Developing compact course timetables with optimized student flows. *European Journal of Operational Research, 251*(2), 651-661. doi:10.1016/j.ejor.2015.11.028.

In Chapter 4 we develop a generic, flexible model for timetabling that incorporates both objective function measures, i.e. the minimisation of the (sum of the) travel times between events in consecutive timeslots and the minimisation of the (sum of the) evacuation times. The model does not focus on the UCTP; instead, it considers a generic timetabling problem so that it can be applied to different settings, such as conference scheduling or scheduling for music festivals and sports events. To keep the problem mathematically tractable, however, only the assignment of events to rooms is optimised, while the assignment of events to timeslots is considered given. This limitation makes it easier to implement the model in practice. The difficult real-world timetabling problem can be solved by other methods, after which our model could be used to optimise the people flows by changing the room assignments. The travel and evacuation times are calculated using the microscopic pedestrian simulator Menge, developed by Curtis et al. (2016). This microscopic simulator allows a much more realistic description of the people flows. The algorithm consists of a tabu search that iteratively searches for better solutions. All candidate solutions in the neighbourhood are first evaluated with a computationally efficient surrogate model and only the best candidate solutions are re-evaluated with the Menge simulator to save computation time. Different surrogate models are compared along three performance criteria both on training data as well as during the search process. Finally, the surrogate-based tabu search is implemented in a scheduling tool with graphical user interface and applied to the timetable for the second semester of academic year 2018-2019 at KU Leuven Campus Brussels.

In Chapter 5 a detailed comparison between the results of the network model of Chapter 3 and the microscopic Menge simulator of Chapter 4 is performed. By means of exhaustive search on small problem instances, the relative objective values assigned to each solution in the solution space by both models are compared. Furthermore, the robustness and modelling power of the models are investigated.

Chapter 2

A review of optimisation models for pedestrian evacuation and design problems

2.1 Introduction

There are many situations in which a large number of people gathers in a single location. Examples include spectators at music and sports events, commuters in railway and metro stations, and employees in large office

An article based this chapter has been published as Vermuyten, H., Beliën, J., De Boeck, L., Reniers, G., Wauters, T. (2016). A review of optimisation models for pedestrian evacuation and design problems. *Safety Science, 87*, 167-178. doi:10.1016/j.ssci.2016.04.001.

buildings. To ensure the safety and comfort of the people present, a careful design of pedestrian facilities and good crowd management are required. Furthermore, in the event of emergencies, such as a fire, a gas leak, or a bomb threat, the efficient evacuation of the facility is of primary importance. The terrorist attacks at the Bataclan theatre in Paris, where 89 people died, and the stampede during the 2015 Hajj pilgrimage in Mecca, where more than 2,070 people died, illustrate the need for developing good crowd management and emergency evacuation procedures.

The study of pedestrian and evacuation dynamics is very complex, due to the large number of people involved and the non-linear interactions between them, psychological factors influencing human behaviour, and the influence of external factors such as the layout of a pedestrian facility. As a consequence, the topic has received attention from researchers in different fields, including psychologists, sociologists, physicists, computer scientists, and traffic scientists (Helbing and Johansson, 2010).

Three distinct, yet interrelated, research streams can be distinguished. The first stream focuses on the empirical study of pedestrian behaviour and crowd dynamics, while the second is concerned with the development of mathematical models to describe the movement and interactions of pedestrians as realistically as possible (Teknomo, 2002). Finally, the third stream of research uses an optimisation-based methodology to develop models which determine optimal evacuation plans or design solutions (Abdelghany et al., 2014). Most of the research falls under the first two categories. Several review articles discuss the empirical research on and modelling of pedestrian and evacuation dynamics. Schadschneider et al. (2009) provide a summary of the empirical studies and theoretical modelling that has been done and give two examples of possible applications of this research. Helbing and Johansson (2010) give a similar overview, and additionally discuss research into situations of panic and critical crowd conditions. Schadschneider and Seyfried (2009) investigate the quantitative data on pedestrian dynamics for the calibration of evacuation models. They focus on the fundamental diagram (see Section 2.3.1) and consider the implications for cellular automata models (see Section 2.4.1). Papadimitriou et al. (2009) assess two different topics of research, namely route choice models and crossing behaviour models, which study how pedestrians cross the street under different traffic conditions. Gwynne et al. (1999) classify 22 evacuation models based on the nature of the model application, the enclosure representation (i.e., how is the building or the area under study represented in the model), the population perspective (i.e., are agents modelled as seperate entities or is the crowd treated as a whole and described using average quantities), and the behavioural perspective (i.e., which assumptions or rules are used to describe the behaviour of pedestrians). Zheng et al. (2009) distinguish seven methodological approaches: cellular automata, lattice-gas, social-force, fluid dynamics, agent-based, game-theoretic models, and experiments with animals. (We give an overview of these approaches in Section 2.4.1.) They also look at the possibility of modelling heterogeneous individuals, the scale of representation, whether time and space are discrete or continuous, whether a normal or an emergency situation is assumed, and the typical phenomena that the model can represent. In addition, Duives et al. (2013) identify eight motion base cases and six self-organising crowd phenomena which a simulation model should be able to reproduce. Furthermore, they look at ten other model characteristics, such as the ability to simulate pressure in crowds and the computational requirements of the model, in order to assess the models' applicability. Their classification distinguishes between cellular automata, social-force, activity-choice, velocity-based, continuum, hybrid, behavioural, and network models. Kalakou and Moura (2014) present a general overview of models from different research areas to analyse the design of pedestrian facilities, while Lee et al. (2003) focus on models for the evacuation of ships. Finally, Bellomo et al. (2012) focus on the mathematical properties of models for pedestrian behaviour. The third category of research has received less attention in the literature. Moreover, to the best of our knowledge, the work of Hamacher and Tjandra (2002) is the only review that focuses on optimisation models for evacuation problems. However, most of the models they discuss are network models with constant (i.e. density-independent) travel times. This chapter tries to fill the gap by critically reviewing the different properties of the optimisation models that are currently available for evacuation and design problems and identifying opportunities for future research.

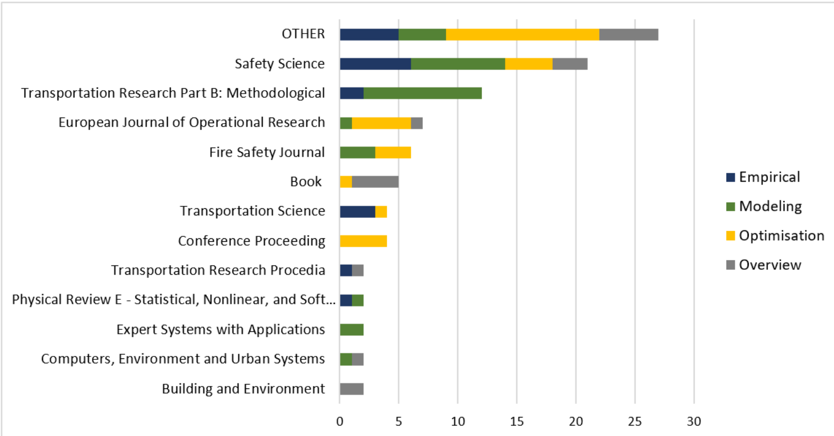
We first searched for literature reviews and articles that discuss general topics related to pedestrian dynamics or evacuation and design problems (Bellomo et al., 2012; Duives et al., 2013; Gwynne et al., 1999; Hamacher and Tjandra, 2002; He et al., 2013; Helbing and Johansson, 2010; Kalakou and Moura, 2014; Lee et al., 2003; Papadimitriou et al., 2009; Schadschneider et al., 2009; Schadschneider and Seyfried, 2009; Sime, 1995; Stanton and Wanless, 1995; Zheng et al., 2009) and checked the references therein. Next, we used the *Web of Knowledge* database to find relevant articles. We used combinations of the keywords 'optimisation', 'problem', 'evacuation, 'pedestrian', 'crowd', 'model', 'movement', and 'flow'. No a priori cut-off date was used, since no previous review articles exist that follow our perspective, apart from the work of Hamacher and Tjandra (2002). Articles on the traffic assignment problem and articles on evacuation and design problems which do not focus on pedestrian traffic and crowd dynamics, are not included. This resulted in a broad, but not exhaustive, overview of the current literature on optimisation models for crowd and evacuation dynamics.

In our review, we distinguish between optimisation and non-optimisation articles. The optimisation category consists of all papers that use a methodology to obtain an optimal or a good solution to a specific problem involving crowd dynamics, such as the efficient evacuation of a building.

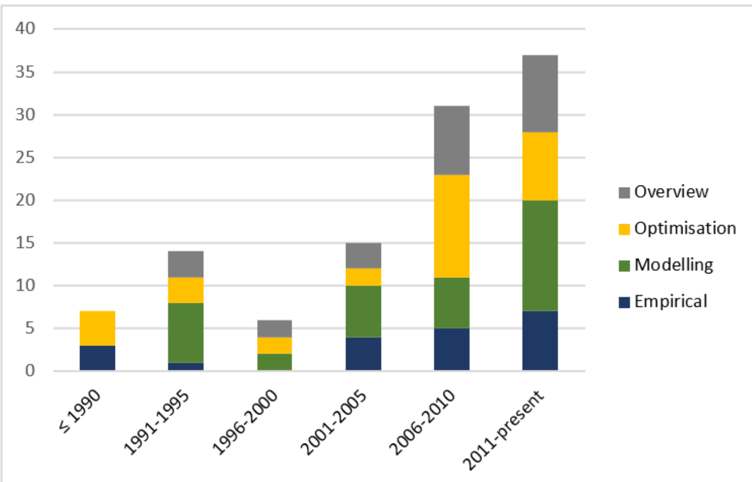
All articles that describe empirical results or descriptive models for the movement of pedestrians that do not use an optimisation methodology, belong to the non-optimisation category. We only take the optimisation articles into account in our classification process. However, we summarise the empirical research and descriptive modelling approaches in our text in order to give the reader the necessary background information for the discussion of the optimisation models. We ended up with 31 optimisation articles that are included in our classification process.

Figure 2.1a lists the journals in which most of the articles in this chapter have been published. We use a cut-off value of two articles. Another 27 articles are all published in different journals. Taking the different types of articles (empirical, descriptive, optimisation, overview) together, *Safety Science* and *Transportation Research Part B: Methodological* are the two journals that publish most of the articles related to pedestrian walking behaviour research. Furthermore, Figure 2.1b gives information on the changing number of articles over the years. It is clear that this research topic has received increasing attention in the last five years.

We use different perspectives for organising the literature. Each section discusses a specific perspective and presents detailed tables in which the relevant articles are categorised. Section 2.2 discusses the different problem types that are studied in the literature, the criteria used to assess the quality of the resulting solutions, i.e. the objective function measures, and the types of decisions that are considered in the model. The realism of the proposed models and their conformity to empirical results on pedestrian dynamics is investigated in Section 2.3. Finally, Section 2.4 analyses the modelling and solution techniques employed to solve the different models. The chapter concludes with the main findings and perspectives for future research in Section 2.5.



(a) Publications per journal (journals with two or more publications).



(b) Publications per year.

Figure 2.1: Overview of publications per journal and per year.

2.2 Problem type, objective function measures, and decisions considered

Optimisation models are used to tackle different types of problems related to pedestrian dynamics. As can be seen from Table 2.1, by far the most attention has been devoted to the development of optimal *evacuation plans* for pedestrian facilities. Many articles specifically focus on a certain type of pedestrian facility, as this enables researchers to tailor models to the specifics of the environment (e.g., Cepolina, 2005). Most models focus on the evacuation of buildings or large rooms with multiple exits. One of the first articles that studied the building evacuation problem was written in 1982 by Chalmet et al. (1982).

A second type of problem is studied by Johansson and Helbing (2005), who look at the problem of finding *designs that improve the flow through a bottleneck*. Flow is the number of pedestrians who pass through a line segment per meter per second. The study of the influence of design on flow was prompted by the observation that placing an obstacle in front of the exit can reduce the magnitude of clogging. A genetic algorithm is used to find the configuration that maximises the outflow.

Thirdly, Selim and Al-Rabeh (1991) study *crowd management* to improve the safety and comfort of pedestrians at mass crowd events. Finally, a fourth type of problem is introduced by Vermuyten et al. (2016b). They minimise student flows in a university course *timetable*, since the assignment of lectures to classrooms in the timetable determines student flows and the resulting travel times between consecutive lectures.

In each of these problem types, different objective function measures can be chosen to evaluate the quality of a solution (see Table 2.2). In the case

Table 2.1: Problem type.

Evacuation planning consists of determining the optimal way to evacuate pedestrian facilities as quickly and safely as possible. Some studies focus on a specific type of facility, such as a building or a room. Design of bottlenecks considers the optimal lay-out that maximises flow or minimises egress time. Crowd management decides on control policies to ensure the safety and comfort of people at mass-crowd events. In timetabling, the problem is to minimise the travel time between events in consecutive timeslots or to minimise the evacuation time in the event of an emergency that is the result of the timing and location of events.

Evacuation planning	
Building	Borrmann et al. (2012); Cepolina (2005, 2009); Chalmet et al. (1982); Chen and Feng (2009); Choi et al. (1988); Deng et al. (2008); Fahy (1994); Georgoudas et al. (2010); Hoppe and Tardos (1994, 2000); Kang et al. (2015); Kisko and Francis (1985); Li and Xu (2014); Park et al. (2009); Talebi and Smith (1985)
Room	Abdelghany et al. (2014); Ding (2011); Pursals and Garzón (2009); Zhao and Gao (2010)
Other	Lim et al. (2015); Ng and Waller (2010); Opasanon and Miller-Hooks (2009); Zarboutis and Marmaras (2007); Zheng and Liu (2010)
Design of bottlenecks	Bakuli and Smith (1996); Berseth et al. (2015); Johansson and Helbing (2005); Tavares (2010)
Crowd management	Selim and Al-Rabeh (1991)
Timetabling	Vermuyten et al. (2016b), Chapter 3, Chapter 4

of evacuation problems, the evacuation time is an important measure of the quality of the proposed plan. Both the average and the maximum evacuation time for all evacuees are used, but the latter is a more popular indicator as it indicates the time that the last person is brought to safety and thus optimises the safety of the least fortunate person. Opasanon and Miller-Hooks (2009) also include the number of people evacuated before a certain time. Other researchers minimise the number of people left in the building at each discrete time step (Hoppe and Tardos, 1994), minimise the maximum probability of congestion that might occur in the evacuation network (Lim et al., 2015), or provide the reader with a set of alternatives to choose from (Zarboutis and Marmaras, 2007). For a further discussion of the many possible performance measures that can be employed for evacuation systems, see Løvås (1995). For design purposes, the maximisation of flow is often used to increase the efficiency of pedestrian facilities, which is important both for normal situations where large pedestrian traffic takes place and for evacuations to reduce congestion and egress times. In the crowd management model of Selim and Al-Rabeh (1991), the authors minimise a penalty function based on the number of people that are denied access at each time interval. Finally, Vermuyten et al. (2016b) minimise the maximum travel time between consecutive lectures across all different timeslots and series of students in their timetabling problem.

In addition to the objective function measures employed, models can also be classified according to the decisions that are included, as is shown in Table 2.3. The choice of evacuation routes for people to use is the most obvious type of decision included in evacuation models. Some models, however, also incorporate phased evacuation, where different groups of people start evacuation at different times. Phased evacuation is used to reduce congestion on the evacuation routes and consequently improve overall egress times. Zarboutis and Marmaras (2007) instead develop

Table 2.2: Problem type and objective function measure.

Evacuation	
Avg. evac. time	Abdelghany et al. (2014); Chalmet et al. (1982); Ng and Waller (2010)
Max. evac. time	Borrmann et al. (2012); Cepolina (2005, 2009); Chalmet et al. (1982); Chen and Feng (2009); Choi et al. (1988); Deng et al. (2008); Ding (2011); Fahy (1994); Georgoudas et al. (2010); Hoppe and Tardos (1994, 2000); Kang et al. (2015); Kisko and Francis (1985); Li and Xu (2014); Lim et al. (2015); Park et al. (2009); Pursals and Garzón (2009); Talebi and Smith (1985); Zhao and Gao (2010); Zheng and Liu (2010)
Number of people to safety	Choi et al. (1988); Hoppe and Tardos (1994); Opasanon and Miller-Hooks (2009)
Other	Hoppe and Tardos (1994); Lim et al. (2015); Zarboutis and Marmaras (2007)
Design	
Max. evac. time	Bakuli and Smith (1996); Tavares (2010)
Flow	Bakuli and Smith (1996); Berseth et al. (2015); Johansson and Helbing (2005)
Crowd management	
Other	Selim and Al-Rabeh (1991)
Timetabling	
Max. travel time	Vermuyten et al. (2016b), Chapter 3, Chapter 4
Max. evac. time	Chapter 3, Chapter 4

generic guidelines for evacuations under different disaster scenarios, instead of proposing a fixed plan for a specific scenario. Furthermore, Talebi and Smith (1985) determine the optimal number of nurses to be assigned to each hospital section to achieve the quickest possible evacuation of patients. A different type of decision is modelled by Selim and Al-Rabeh (1991), who develop an admission control policy for pedestrians on the Jamarat Bridge to ensure crowd density does not reach hazardous levels. For the category of design problems, Bakuli and Smith (1996) determine the optimal widths of exits in a building that maximise throughput, while Berseth et al. (2015) derive the optimal placement of obstacles in corridors and at exits to reduce the amount of clogging. Finally, Vermuyten et al. (2016b) reassign lectures to classrooms in a university course timetable to minimise the maximum travel time of students between consecutive lectures.

2.3 Model realism

It is important that optimisation models represent crowd dynamics in a realistic way and are calibrated with empirical data to provide useful results for evacuation and design purposes. In Subsection 2.3.1, we first present a summary of the main findings of the empirical research on pedestrian and crowd dynamics. In Subsection 2.3.2, we discuss the implications of these findings for the development of optimisation models and the problem of parameter calibration. Finally, in Subsections 2.3.3 and 2.3.4, we discuss the incorporation of uncertainty into the models and their applicability respectively.

Table 2.3: Problem type and decisions considered.

Evacuation	
Route choice	Borrmann et al. (2012); Cepolina (2005); Chalmet et al. (1982); Chen and Feng (2009); Choi et al. (1988); Deng et al. (2008); Ding (2011); Fahy (1994); Georgoudas et al. (2010); Hoppe and Tardos (1994, 2000); Kang et al. (2015); Kisko and Francis (1985); Li and Xu (2014); Lim et al. (2015); Opasanon and Miller-Hooks (2009); Park et al. (2009); Pursals and Garzón (2009); Zhao and Gao (2010); Zheng and Liu (2010)
Phased evacuation	Abdelghany et al. (2014); Cepolina (2009); Ng and Waller (2010)
Generic guidelines	Zarboutis and Marmaras (2007)
Allocation of staff	Talebi and Smith (1985)
Design	
Location of obstacles	Bakuli and Smith (1996); Berseth et al. (2015); Johansson and Helbing (2005); Tavares (2010)
Crowd management	
Admission control policy	Selim and Al-Rabeh (1991)
Timetabling	
Location of events	Vermuyten et al. (2016b), Chapter 3, Chapter 4
Timing of events	Chapter 3

2.3.1 Empirical research on pedestrian and crowd dynamics

A lot of early empirical research focused on the relationship between walking speed, $v\left(\frac{m}{s}\right)$, and density, $\rho\left(\frac{\text{people}}{m^2}\right)$, of pedestrian flows. In the same way, the relationship between flow, $q\left(\frac{\text{people}}{m.s}\right)$, and density can be derived, where $q(\rho) = \rho v(\rho)$. These relationships are called the 'fundamental diagram', because of their importance in determining the optimal dimensions of pedestrian facilities (Schadschneider and Seyfried, 2009). An early study in 1958 by Hankin and Wright (1958) carried out experiments with schoolboys, in which they measured speeds at various concentrations and various passage widths, to obtain the shape of the speed-density and flow-density curves. Then observations were done at a London underground station in order to obtain absolute values for the established relationships. The four parameters that describe this relationship are ρ_{max}, i.e. the maximum density at which walking speed reaches zero, v_0, i.e. the maximum free walking speed at zero density, and ρ_c and q_{max}, which denote the critical density at which the maximum flow is reached.

There are, however, significant differences between the results of various studies (Fruin, 1971; Helbing et al., 2007; Johansson et al., 2008; Mōri and Tsukaguchi, 1987; Polus et al., 1983; Seyfried et al., 2005). Table 2.4 summarizes the values obtained by different authors. Several explanations have been suggested for the differences in the obtained results (Schadschneider and Seyfried, 2009): Helbing et al. (2007) mention cultural and population differences; Predtechenskii and Milinskii (1978) argue that the incentive of the movement matters; and Oeding (1963) suggests the type of traffic plays a role (e.g., commuters compared to shoppers).

Additionally, the standard fundamental diagram is derived for unidirectional flows. There is discussion as to whether the diagram is different for uni- and bidirectional flows (Schadschneider and Seyfried, 2009). Recently, Flötteröd and Lämmel (2015) studied the bidirectional fundamental diagram. They analytically derive a bidirectional fundamental diagram from a simple cellular automata model. Their model is compared with the social-force model of Helbing (1991) and Helbing and Molnár (1995) and their results are validated against empirical data and well-known crowd phenomena. A discussion of these modelling techniques is provided in Section 2.4.1.

Venuti and Bruno (2007) develop a mathematical model for the fundamental relationship that takes into account various factors to reconcile the different observed values in the literature. They specifically focus on the lateral movement of the ground surface, the geographic area, and the travel purpose, but the model can be extended to include other factors as well. Their model is able to explain the differences in results between the various empirical studies. Additionally, Galiza and Ferreira (2013) use the concept of equivalent factors to convert heterogeneous pedestrian flow into an equivalent base flow.

Finally, some researchers have observed that at densities higher than ρ_{max}, walking speed does not reach zero as is predicted in other studies and 'turbulent' crowd conditions arise, in which people can no longer move freely but instead are pushed around by pressure waves in the crowd (Helbing and Johansson, 2010; Helbing et al., 2007; Johansson et al., 2008).

Besides the standard fundamental diagram for walking speeds on regular horizontal surfaces, walking speeds on stairs have been investigated by some researchers, both descending (Ma et al., 2012) and ascending (Lam et al., 2014), as well as for different dimensions (e.g., the height and length of a step) and circumstances (normal and emergency) (Yang et al., 2012).

A second and related topic of study has been the flow through bottlenecks (e.g., exits). Hoogendoorn and Daamen (2005) study the unidirec-

Table 2.4: Parameters for the speed-density and flow-density relationship from various studies.

Study	v_0 $\left(\frac{m}{s}\right)$	ρ_{max} $\left(\frac{\text{people}}{m^2}\right)$
Fruin (1971)	1.30	6.60
Hankin and Wright (1958)	1.61	6.46
Johansson et al. (2008)	0.60	10.79
Mōri and Tsukaguchi (1987)	1.40	9.00
Polus et al. (1983)	1.25	7.18
Seyfried et al. (2005)	1.34	5.55

tional flow through a bottleneck for different widths. They observe that pedestrians dynamically form layers inside the bottleneck, where pedestrians are positioned diagonally to the people in front and behind. This phenomenon is called the 'zipper effect', because the layers overlap like interlocking teeth in a zipper. This implies that the capacity of a bottleneck increases in a stepwise manner with the bottleneck width, instead of linearly, depending on how many layers can be formed. Seyfried et al. (2009), however, do find a linear relationship between flow and bottleneck width. They argue that the stepwise relationship is based on the faulty assumption that within the bottleneck layers are formed with a constant distance. They also find that jamming occurs below the capacity limit and formulate three hypotheses as an explanation: flow fluctuations, the local organisation of pedestrians, and a preference for larger distances than necessary from the person in front. Helbing et al. (2005) and Liu et al. (2014) study bidirectional flows through bottlenecks. They find oscillation effects, where multiple pedestrians consecutively pass the bottleneck in a single direction, and clogging effects, where at high densities the movement of pedestrians comes to a halt and dangerous pressures are built up in the queues.

Aside from studies that derive quantitative results for pedestrian flows under normal circumstances, other studies have focused on evacuations, since the correct estimation of evacuation times is critical for safety. Olsson and Regan (2001) study the evacuation times of three university buildings. They specifically include pre-movement times, i.e. the time people need to realise that they need to evacuate and to decide on a course of action. They argue that the SIMULEX software can be used in evacuation scenario analysis to obtain reliable results. Kady (2012) studies the relationship between the density and crawling movement of pedestrians in the event of a fire. The author finds that exit width has a significant impact on crawling speed, while population size is less important. Spearpoint and MacLennan (2012) use a Monte Carlo simulation model to investigate the impact of gender, age, and obesity on the evacuation time from a high-rise building.

Furthermore, an important factor of safety concerns the pressures which are experienced by pedestrians in extremely high-density crowds (Helbing and Johansson, 2010; Helbing et al., 2007). Smith and Lim (1995) investigate the pressure which people can 'comfortably' endure when pushed against barriers.

Finally, various self-organising crowd phenomena have been observed (Duives et al., 2013; Helbing and Johansson, 2010; Moussaïd et al., 2009). These phenomena are self-organising because they are the result of local interactions between many pedestrians, without any conscious actions of pedestrians to arrive at these phenoma (Helbing and Johansson, 2010). The most important phenomena are:

Lane formation: In bidirectional flows, pedestrians automatically start forming a number of lanes of varying width, with people in each lane moving in the same direction (Schadschneider et al., 2009).

Stripe formation for two intersecting flows: When two pedestrian flows intersect, stripes are formed in which pedestrians move forward with the stripes and sidewards within the stripes. This is a result of pedestrians trying to minimise friction with pedestrians moving in opposite directions. For three or more intersecting flows, no stable patterns emerge (Helbing et al., 2005).

Stop-and-go waves: At high densities pedestrians cannot move continuously. Instead, the crowd moves in waves (Helbing et al., 2007).

Turbulence: At extremely high densities pedestrians cannot control their own movements anymore, but are pushed around by the forces acting upon them (Helbing et al., 2007).

Herding: When individuals do not have knowledge of the optimal route, they start following others. This happens especially during evacuations (Helbing et al., 2005).

Zipper effect: In a bottleneck individuals move diagonally in front of others such that narrower lanes are formed and the capacity of the bottleneck increases (Hoogendoorn and Daamen, 2005).

Faster-is-slower effect: When people keep moving forward when a bottleneck is congested, crowd motion is slowed down by the resulting friction (Helbing and Johansson, 2010).

2.3.2 Implications for modelling

In order to provide realistic results, optimisation models for evacuation or design problems should explicitly incorporate the different empirical results described in the previous section. To asses the realism of the models reviewed, we first focus on three model attributes which capture the different elements of pedestrian and crowd dynamics:

Congestion: Does the model include the relationship between walking speed and density? This means that travel times or flow capacities cannot be assumed to be constants, but should be modelled as endogenous variables dependent on the number of pedestrians present at a certain location.

Bottlenecks: Are bottlenecks such as exits explicitly included in the model? Bottleneck capacities should be based on the width of the bottleneck and the number of people queuing upstream of the bottleneck.

Direction of flow: Does the model distinguish between uni- and bidirectional flows?

The first part of Table 2.5 lists the models which explicitly include these modelling aspects. We see that the majority of articles include congestion in their models, while only a smaller number explicitly include bottlenecks. Finally, most articles do not distinguish between uni- and bidirectional flows. Overall, these results might be considered as being positive, because the most important aspect (congestion) is included in most of the recent articles. Furthermore, incorporation of the direction of flow is less important, because there is still debate as to whether there even is a significant difference between the parameter values for uni- and bidirectional flows (Schadschneider and Seyfried, 2009).

A second way to judge the realism of optimisation models is by looking at their ability to reproduce (some of) the self-organising crowd phenomena that have been observed empirically. We base our assessment on the information the authors provide in their articles (we have not tested the models ourselves). The second part of Table 2.5 shows the results. In only three articles do the authors validate their model by testing its ability to reproduce these phenomena. Of course, only microscopic simulation models, in which each pedestrian is modelled individually, are able to show these dynamics explicitly. However, this does not imply that other modelling techniques cannot reproduce realistic results for evacuation or

Table 2.5: Model realism.

Incorporation of crowd dynamics	
Congestion	Abdelghany et al. (2014); Bakuli and Smith (1996); Berseth et al. (2015); Borrmann et al. (2012); Cepolina (2005, 2009); Choi et al. (1988); Deng et al. (2008); Fahy (1994); Georgoudas et al. (2010); Johansson and Helbing (2005); Kang et al. (2015); Lim et al. (2015); Park et al. (2009); Pursals and Garzón (2009); Talebi and Smith (1985); Tavares (2010); Vermuyten et al. (2016b); Zarboutis and Marmaras (2007); Zhao and Gao (2010); Zheng and Liu (2010), Chapter 3, Chapter 4
Bottlenecks	Berseth et al. (2015); Borrmann et al. (2012); Cepolina (2009); Chen and Feng (2009); Johansson and Helbing (2005); Kang et al. (2015); Li and Xu (2014); Park et al. (2009); Pursals and Garzón (2009); Talebi and Smith (1985); Tavares (2010); Zhao and Gao (2010), Chapter 4
Direction of flows	Berseth et al. (2015); Deng et al. (2008); Georgoudas et al. (2010); Tavares (2010); Zarboutis and Marmaras (2007); Zhao and Gao (2010), Chapter 4
Reproducing crowd phenomena	Borrmann et al. (2012); Johansson and Helbing (2005); Zhao and Gao (2010)
Calibration	
Model tweaking	Borrmann et al. (2012); Zhao and Gao (2010); Zheng and Liu (2010)
Real-world data	Cepolina (2005, 2009); Fahy (1994); Georgoudas et al. (2010); Pursals and Garzón (2009)

design purposes.

Finally, to produce output that has real-world applicability, optimisation models need to calibrate their parameters based on empirical data on walking behaviour and crowd dynamics. There are two ways in which model parameters can be calibrated. The preferred method is to match the value of model parameters, e.g., the preferred walking speed of an individual, to their observed value in empirical studies (Bellomo et al., 2012). However, in reality model parameters are often iteratively adjusted, until the model produces realistic phenomena and output values (Bellomo et al., 2012). The third part of Table 2.5 lists the articles which use the a priori or the a posteriori calibration method respectively. Only a quarter of the papers that we have reviewed mention calibration of their models. One reason for this is the difficulty of calibrating parameters caused by the significant differences in results that have been obtained in empirical studies (Schadschneider and Seyfried, 2009). The approach taken by Venuti and Bruno (2007) of including factors that can explain the differences in results in empirical studies of the fundamental diagram, could lead to progress in this area (Bellomo et al., 2012).

2.3.3 Incorporation of uncertainty into the model

Evacuations often happen in response to a disaster such as a fire. However, this event usually happens unexpectedly, giving rise to a lot of uncertainty. Indeed, the number of people present at a certain facility and their locations are often not known with certainty. Also, the way the disaster affects the environment, e.g. the propagation of smoke during a fire, and the resulting effects on the evacuation process, can often not be predicted accurately. This has prompted researchers to include uncertainty in their models. We make a distinction between two methods of including uncertainty: predefined probabilities, where parameters or events have a

Table 2.6: Incorporation of uncertainty.

Predefined probabilities	Lim et al. (2015); Ng and Waller (2010); Opasanon and Miller-Hooks (2009); Talebi and Smith (1985); Zheng and Liu (2010)
Real-time updating	Chen and Feng (2009); Deng et al. (2008); Fahy (1994); Georgoudas et al. (2010); Li and Xu (2014); Opasanon and Miller-Hooks (2009); Park et al. (2009)

range of possible values or probabilities instead of being deterministic and known, and real-time updating, where the optimisation model uses real-time information on the event to update and adjust the proposed solution. The resulting classification is shown in Table 2.6.

2.3.4 Applicability of the model

Optimisation models should of course be tested to illustrate their applicability to real-world cases. In Table 2.7, we therefore classify the articles into three categories, namely ‘no testing’, ‘theoretical data’, and ‘real-world data’. It is clear that the majority of papers use theoretical data to test their models. So there is still a lack of implementation of the proposed optimisation models to practical problems.

2.4 Modelling and solution techniques

In this section, we discuss the different modelling and solution techniques that are proposed in the literature for evacuation problems and design

Table 2.7: Applicability of research.

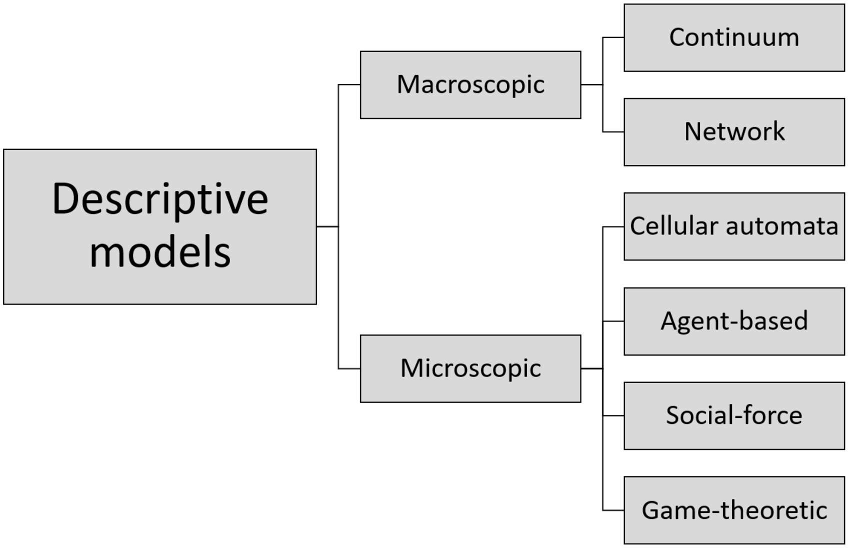
No testing	Choi et al. (1988); Hoppe and Tardos (1994, 2000)
Theoretical data	Abdelghany et al. (2014); Bakuli and Smith (1996); Berseth et al. (2015); Borrmann et al. (2012); Cepolina (2005, 2009); Chalmet et al. (1982); Chen and Feng (2009); Deng et al. (2008); Ding (2011); Johansson and Helbing (2005); Kisko and Francis (1985); Li and Xu (2014); Lim et al. (2015); Ng and Waller (2010); Opasanon and Miller-Hooks (2009); Park et al. (2009); Pursals and Garzón (2009); Tavares (2010); Vermuyten et al. (2016b); Zarboutis and Marmaras (2007); Zhao and Gao (2010); Zheng and Liu (2010), Chapter 3, Chapter 4
Real-world data	Fahy (1994); Georgoudas et al. (2010); Kang et al. (2015); Selim and Al-Rabeh (1991); Talebi and Smith (1985), Chapter 3, Chapter 4

of pedestrian facilities. To provide some background information and ideas for the development of more realistic optimisation models in the future, we first discuss the main techniques used in descriptive models in Subsection 2.4.1 to realistically represent pedestrian walking behaviour. Afterwards, we compare this with the modelling and solution techniques that are currently used in optimisation models in Subsection 2.4.2.

2.4.1 Modelling techniques used in descriptive models

As mentioned above, we briefly discuss some of the approaches that have been developed in the literature for the modelling of pedestrian behaviour and crowd dynamics. We do not intend to give an exhaustive overview of the different modelling techniques or an in-depth discussion of the properties of each model that is included. The interested reader can find detailed

Figure 2.2: An overview of the modelling techniques used in descriptive models.



assessments of the existing modelling approaches and simulation models in Duives et al. (2013), Papadimitriou et al. (2009), and Zheng et al. (2009). An overview of the different techniques is shown in Figure 2.2.

2.4.1.1 Continuum models

Continuum models are macroscopic simulation models. Pedestrians are not represented individually; instead crowds are described as a fluid using average quantities such as the density at a given location. Mathematically, these models consist of a system of partial differential equations, expressing the relationship between average speed, flow, and density at a given location and time (Bellomo et al., 2012). Both time and space are continuous. A distinction can be made between first-order models, which only include an equation for the conservation of mass, and second-order models, which also include a momentum balance equation (Bellomo et al., 2012). Since it is computationally efficient, the continuum approach is often used when very large crowds need to be modelled or when only an estimation of the average quantities is required. One of the first authors that applied these continuum models to pedestrian traffic was Hughes (2002). He develops a first-order model based on three hypotheses: (i) pedestrians' speed is determined by the local density at their location, (ii) pedestrians' movement is perpendicular to lines of constant potential, and (iii) pedestrians want to take the path with the shortest travel time, but only if the density on this path is not too high. Huang et al. (2009) prove that Hughes' model satisfies the reactive dynamic user equilibrium, which means that pedestrians choose the route that minimises their instantaneous travel cost to the destination. They also develop an efficient solution method to solve the model. Hoogendoorn and Bovy (2001) present a continuum model which applies to different types of traffic, i.e. both vehicular and pedestrian traffic. They develop the concept of generalised phase-space density, to include different attributes such as user-class, roadway lane, destination, velocity, and desired velocity. Appert-Rolland et al. (2011) focus on the incorporation of the maximally allowable density into continuum models. Finally, Hänseler et al. (2014) combine the continuum approach and the cell transmission approach from vehicular traffic in order to predict travel times and densities. They apply their model to two case studies and obtain good results. A review of some continuum models is given by Twarogowska et al. (2014).

2.4.1.2 Network-based models

Network models represent a pedestrian facility as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where the set $\mathcal{N}$ of nodes represents the different rooms and the set $\mathcal{A}$ of arcs the links between them. Løvås (1994) describes pedestrian dynamics in the network by a queuing model where each pedestrian is a seperate flow object. This model is implemented in the evacuation software EVAC-SIM (Drager et al., 1992) and solved using discrete-event simulation. In a subsequent paper, the same author (Løvås, 1998) discusses different wayfinding models that can be used in a network setting. Guo et al. (2011) develop a network-based model for the evacuation of pedestrians in indoor areas. The model discretises each part of the building in detail using hexagonal cells and allows consideration of internal obstacles, giving a realistic representation. However, each cell can contain multiple pedestrians, so the model is not microscopic in that sense. This gives computational advantages. Pedestrians choose their route based on a potential field, which denotes the trade-off between distance and congestion.

2.4.1.3 Cellular automata models

Cellular automata models are microscopic simulation models where pedestrians are considered individually. They represent the building lay-out by a grid divided into cells. Usually, each cell can be occupied by a single pedestrian (e.g., Blue and Adler, 2001). However, some models allow several pedestrians into one cell for scaling purposes, while others use smaller cells where each pedestrian occupies multiple cells, to allow for a greater degree of detail (e.g., Guo et al., 2012). Time is discretised and at each time step, pedestrians either move to a neighbouring cell or remain at their current location. The decision taken by a pedestrian depends on the status of the adjacent cells and is based on a predefined set of rules. Updating of cells can be executed either sequentially (e.g., Guo et al., 2012) or in parallel (e.g., Blue and Adler, 2001), in which case movements can only be executed when all conflicts between pedestrians are resolved. One of the first cellular automata models for the simulation of pedestrian movements was developed by Blue and Adler (2001). The authors focus on the various phenomena observed in bidirectional flows. Guo et al.

(2012) develop two route choice models, for the case of good and bad visibility respectively. Bad visibility means that pedestrians cannot see which direction they should walk to. It can be the result of the propagation of smoke through the building in case of a fire. Pereira et al. (2013) explicitly include the relationship between average speed of a pedestrian and the density in the model. An advantage of the approach is its computational efficiency.

2.4.1.4 Agent-based models

Agent-based models take a bottom-up approach as well, where only the behaviour of individual pedestrians is modelled and the resulting interactions between them determine the macroscopic behaviour. Agent-based models can use both discrete and continuous time and space representations. Each agent can have a unique set of behavioural rules, which allows for modelling heterogeneity in the population (e.g., different preferred walking speeds for old and young people). A disadvantage of this flexibility is the high computational cost of running the model. Antonini et al. (2006) use a discrete choice framework in which pedestrians choose a direction and speed based on the utility of each of the alternatives. This utility is influenced by the presence of other pedestrians. Chooramun et al. (2012) combine three space representations (continuous space, fine network, and coarse network) into a single model to achieve an optimal trade-off between computational efficiency and model realism. Behaviour of agents is based on a different set of rules at each representation level. The MOBEDIC tool developed by Doheny and Fraser (1996) models the actions of people in specific emergency situations, specifically focusing on the evacuation of an offshore environment. EXODUS is a similar software tool, developed by Galea and Perez Galparsoro (1994), intended for the evacuation of mass-transport vehicles such as aircraft. It is also able to simulate crawling movement during evacuations (Muhdi et al., 2009). A third software tool, developed for simulating the evacuation of geometrically complex buildings, is the SIMULEX model of Thompson and Marchant (1995a,b,c). Recently, Wagner and Agrawal (2014) developed an agent-based model for the evacuation of concert venues. The propagation of fire and smoke is included in the model and influences the route choice behaviour of individuals. However, there are still many challenges involved in the development of agent-based models, see Crooks et al. (2008) for a discussion.

2.4.1.5 Social-force models

A third set of microscopic models consists of the so-called social-force models. In this type of model, pedestrians have a desired velocity in the direction of their destination and their acceleration (deceleration) is the result of different forces. An individual experiences an attractive force in the direction of his target destination, and repulsive forces from obstacles (e.g., walls) and other pedestrians. Time and space are modelled in a continuous way. The social-force model was developed by Helbing (1991) and Helbing and Molnár (1995). The model reproduces well-known self-organising crowd phenomena such as lane formation in bidirectional flows and oscillatory effects at bottlenecks. Langston et al. (2006) represent pedestrians by three intersecting circles instead of a single circle, to incorporate the rotation of the pedestrians into the model. The model is realistic for dense crowd flow scenarios, but more complex scenarios are not yet fully realistically represented. Yuen and Lee (2012) extend the social-force model to include overtaking behaviour, where pedestrians with a higher desired velocity catch up with and move past pedestrians heading in the same direction with a lower desired velocity. Qu et al. (2014) also use a three-circle representation to model rotation and extend the social-force model to describe pedestrian movement on stairs.

2.4.1.6 Game-theoretic models

Hoogendoorn and Bovy (2003) use the theory of differential games to describe the walking behaviour of pedestrians. In this model, pedestrians predict the behaviour of other pedestrians based on the current state and anticipated actions of other pedestrians in their neighbourhood (predictive dynamic user equilibrium principle). They base their pedestrian walking behaviour model on a clear theoretical foundation based on the microeconomic notion of subjective utility maximisation. The same authors develop a comprehensive theory of pedestrian activity and path determination in the two-dimensional space (Hoogendoorn and Bovy, 2004). Huang et al. (2009) instead use a reactive user equilibrium principle in which pedestrians only evaluate the immediate conditions of their environment without anticipating the behaviour of pedestrians in their surroundings (Tong and Wong, 2000). Their model is an extension of the macroscopic model of Hughes (2002). Lachapelle and Wolfram (2011) present a pedestrian crowd model based on the theory of mean field games. The model is macroscopic, i.e. it describes crowd behaviour in terms of aggregates, but it is based on a realistic microscopic model in the sense that it considers smart pedestrians with rational expectations. Pedestrians are represented as agents having preferences (i.e., they want to maximise their utility) and perform strategic interactions within the crowd. They also anticipate the future. This approach is similar to that of Hoogendoorn and Bovy (2003), but an advantage of the former model is its lower computational cost as compared to microscopic simulation models.

2.4.1.7 Advantages and disadvantages of different models

Agent-based models offer a wide variety of modelling choices. However, this also makes the models more sensitive to the assumptions used and the choice of parameter values. Computational costs are usually high since pedestrians are represented as separate entities and can have complex behaviours. The theory of subjective utility maximisation on which game-theoretic models are based is very intuitive. These models can describe the various motion base cases such as bidirectional flows, four-directional flows, or crossing flows, and exhibit many of the self-organising crowd phenomena discussed in Section 2.3.1 (Duives et al., 2013). These models are again computationally costly for the same reasons as the agent-based models. Social-force models are also based on an intuitive foundation, namely an analogy with physical forces that result in accelerations and decelerations. These models are more or less able to simulate the same behaviours and crowd phenomena as the game-theoretic models (Duives et al., 2013). However, these models are not robust (i.e., they require a small time step when solving the system of differential equations to achieve accurate results) and are thus not as computationally efficient. The first reason is that the forces need to be integrated twice to obtain the positions of pedestrians. The second reason is that the equations for the repulsive forces between pedestrians and obstacles are stiff, meaning that small changes in relative positions have a large impact on the resulting forces (Curtis, 2014). Network models on the other hand, are computationally very cheap but less realistic. Continuum models offer a trade-off between network models and microscopic models. They give fairly accurate results on an aggregated level and their computation time does not depend on the number of people present in the model. Network and continuum models are also less sensitive to the choice of parameter values than microscopic models.

2.4.2 Modelling techniques used in optimisation models

Table 2.8 lists the different papers according to the optimisation modelling technique that is used.

Many early models focus on exact methods, such as standard network flow models and dynamic programming (i.e. shortest path) to determine optimal evacuation plans. Chalmet et al. (1982) represent a building by a graph in which the nodes denote the rooms and the arcs the connections, i.e. doors, between them. They use a dynamic network flow algorithm to simultaneously minimise the average evacuation time, the maximal evacuation time, and to maximise the total number of people evacuated by a given time. Another example is the EVACNET+ software developed by Kisko and Francis (1985), which uses a network flow algorithm to determine optimal evacuation routes.

Additionally, many authors develop a dedicated algorithm to solve their respective models. Ding (2011) presents an evacuation model where people are assigned to different exit routes, each with a certain length and width, such that the total evacuation time is minimised. The author derives an expression for the number of people that should be assigned to each exit route, based on the observation that the evacuation time over all routes should be equal, since it is the last person's egress time that should be minimised. A similar problem is described by Pursals and Garzón (2009). The expressions proposed by Nelson and MacLennan (1995) are used to model the movement of people, i.e. to represent the non-linear relationship between density and travel time on a given route. The authors then adapt the algorithm of Brown (1979) for the knapsack sharing problem to solve their problem.

Table 2.8: Solution technique.

Mathematical programming	
Shortest path	Chen and Feng (2009); Fahy (1994); Park et al. (2009)
Network flow transshipment	Borrmann et al. (2012); Chalmet et al. (1982); Choi et al. (1988); Kisko and Francis (1985)
Integer programming	Kang et al. (2015); Lim et al. (2015); Vermuyten et al. (2016b), Chapter 3
Chance constraint programming	Ng and Waller (2010)
Heuristic	
Simulated annealing	Cepolina (2005, 2009)
Genetic algorithm	Abdelghany et al. (2014); Johansson and Helbing (2005)
Other	Chapter 3, Chapter 4
Simulation	
Cellular automata	Abdelghany et al. (2014); Georgoudas et al. (2010); Zhao and Gao (2010)
Agent-based modelling	Tavares (2010); Zarboutis and Marmaras (2007)
Other	Berseth et al. (2015); Deng et al. (2008); Johansson and Helbing (2005); Zheng and Liu (2010), Chapter 4
Queuing	Bakuli and Smith (1996); Deng et al. (2008); Talebi and Smith (1985)
Dedicated algorithm	Choi et al. (1988); Ding (2011); Georgoudas et al. (2010); Hoppe and Tardos (1994, 2000); Kang et al. (2015); Li and Xu (2014); Opasanon and Miller-Hooks (2009); Pursals and Garzón (2009); Selim and Al-Rabeh (1991)

However, while computationally efficient, these models do not model pedestrian dynamics well, as they assume capacities and arc-traverse times to be constant instead of density-dependent. In recent years, the development of more realistic models which better represent crowd phenomena, has shifted attention towards the use of queuing models and heuristics on the one hand and the use of simulation on the other hand to cope with the increased complexity.

Queuing models can represent buildings as a graph where nodes correspond to rooms or bottlenecks and arcs correspond to the connections between them (Bakuli and Smith, 1996; Talebi and Smith, 1985), or by a lattice where each cell can be occupied by a number of people and has a queuing process associated with it (Deng et al., 2008). The travel and waiting time are modelled by the queuing process at each node. The service rate is a function of the number of people present because of the inverse relationship between walking speed and density of pedestrians. The advantage of these models is that they include this non-linear relationship, instead of assuming constant travel times and capacities, while at the same time being computationally efficient to solve. Deng et al. (2008) combine Markov Decision Process models and queuing theory to model the evacuation of a building. The Markov process describes the typical egress behaviour of an agent, while a queue at each building node is used to model congestion. Optimal evacuation routes are derived using a MaxWeight policy for decentralised routing, where each agent chooses from a set of Markov transition matrices at each time step. It is a myopic policy, because at each time step the routing is chosen based only on the current state of the network, and essentially translates to diverting traffic from the most congested nodes to other routes.

Cepolina (2005) uses simulated annealing to find the optimal evacuation routes in a building. Only a special case of building geometry is considered, which restricts the solution space so that a simple transition rule can be applied in the simulated annealing heuristic. The author extends this work (Cepolina, 2009) to include the capacity drop phenomenon in bottlenecks under oversaturated conditions. The problem is extended not only to finding the optimal egress routes, but also to deciding on the optimal start times of evacuation for each floor of the building (so-called phased evacuation).

Currently, simulation models are often used in an iterative solution procedure to solve evacuation and design problems. The reason is that simulation models represent the complex interactions between pedestrians realistically and can be adapted to many different scenarios, while simultaneously remaining mathematically tractable compared to monolithic non-linear mathematical programming models. An example of such an iterative solution procedure is provided by Abdelghany et al. (2014) who use a genetic algorithm combined with a cellular automata simulation model to evacuate a heterogeneously distributed group of people from a large room with multiple exits. Every chromosome in the population represents a solution where each group of people is assigned to a specified exit. The cellular automata model then simulates the evacuation dynamics resulting from this assignment and the corresponding evacuation time. After each run a new population is created from the previous one, until a stopping criterion is reached. The solution with the lowest evacuation time then represents the best evacuation plan that has been found.

Similar techniques are used by Johansson and Helbing (2005) and Tavares (2010) for design problems. Johansson and Helbing (2005) use a genetic algorithm in combination with the social-force model to find an improved layout to increase the flow through a bottleneck.

Most articles that we studied do not discuss the computational requirements of their models. An exception is the models by Georgoudas et al. (2010) and Li and Xu (2014), which track pedestrians in real-time and suggest rerouting pedestrians based on anticipated congestion at exits. In this case, computational efficiency is of the utmost importance in order to be able to determine optimised evacuation routes in real-time.

2.5 Discussion and conclusion

In this chapter, we have reviewed optimisation models from the field of pedestrian walking behaviour and crowd dynamics. These models are used for a wide range of evacuation and design problems. We have also discussed the relevant empirical research and descriptive modelling techniques to provide a background for the reader and to substantiate the criteria that are used in the assessment of the different models.

Currently, most of the attention is directed to the development of optimal evacuation plans, followed by the effective design of pedestrian facilities. However, there are other interesting problems related to pedestrian flows which have not yet received much attention in the literature, such as crowd management under normal conditions. An example is the minimisation of flows resulting from the timing and location of certain events, such as the assignment of lectures to rooms and timeslots in a university timetable, the scheduling of acts at music festivals, or the planning of different disciplines at large sports events, to ensure the safety (i.e., crowd densities do not reach hazardous levels) and comfort (i.e., people do not have to walk large distances or through high-density crowds and can reach their destinations in time) of the people present.

While many of the earlier models concerning evacuation problems did not include the fundamental relationship between walking speed and crowd density, and instead assumed constant travel times, most of the recent articles represent these dynamics in their models. By way of contrast, the calibration of models should receive more attention in future work. However, calibration is still difficult because of the lack of consensus between data of different empirical studies. More research is needed in this area to reconcile or explain the contradictory results obtained in experiments.

Closely related to this is the validation and application of optimisation models. Currently, most authors only test their models on theoretical data. To implement the models in practice, it is important that their results and predictions closely resemble real-world values. Furthermore, practitioners could benefit if authors describe the different challenges and pitfalls in implementing their models.

Finally, there currently is a discrepancy between the techniques used in descriptive models and those used in optimisation models. The former are mostly variants of microscopic simulation models, because they seek to represent pedestrian dynamics as realistically as possible. By way of contrast, the latter gravitate towards network models in combination with flow transshipment algorithms or queuing processes, because of their mathematical tractability. Some of the recent models use an iterative process where a heuristic searches for good solutions, which are consequently tested by a simulation model that represents the resulting crowd dynamics in a realistic way. Future research should focus on integrating techniques of descriptive models within an optimisation framework to find the optimal trade-off between model realism and tractability.

Chapter 3

Developing compact course timetables with optimised student flows

3.1 Introduction

The growing student numbers at colleges and universities have resulted into an enlarged complexity in terms of planning and organisation. One of the tasks that becomes increasingly complex is the development of course timetables. Daskalaki et al. (2004) define the University Course Timetabling Problems (UCTP) as the construction of a weekly timetable in which all operational rules and requirements of the academic institution

An article based on parts of this chapter has been published as Vermuyten, H., Lemmens, S., Marques, I., Beliën, J. (2016). Developing compact course timetables with optimized student flows. *European Journal of Operational Research, 251*(2), 651-661. doi:10.1016/j.ejor.2015.11.028.

are met and as many wishes as possible of the staff and students are satisfied. According to Carter and Laporte (1998) the UCTP can be formulated as a multi-dimensional assignment problem. Students and lecturers need to be assigned to lectures which are in turn assigned to rooms and timeslots such that no overlap occurs. Course timetables have to satisfy various requirements of different stakeholders including non-overlap of courses, free hours, lecturers' preferences, student preferences, etc. Furthermore, the course timetable can have a huge impact on queues in stair halls and elevators, particularly for universities or colleges with many students that follow courses in a single building. The congestion problems in stair halls and elevators are caused by travelling students that all have to switch rooms at the same time between two consecutive lectures. Clearly, student flows can be controlled and monitored via the course timetables. For example, if the schedules are arranged so that consecutive lessons take place in rooms situated on the same floor (or on a floor as close as possible), there will be far fewer queues at the elevators and in the stairwells. Thus, next to the various constraints and preferences of different stakeholders, the resulting student flows should also be taken into account when building the course timetable.

This research was motivated by the UCTP at the KU Leuven Faculty of Economics and Business (FEB) campus Brussels. As described in Mercy (2012) the FEB campus Brussels has gone through a process of campus consolidation in which several buildings at different locations in Brussels have been sold and the lectures of all economic programs have been concentrated at a single location in the center of Brussels. As a result, over 8000 students daily follow classes in a single building, which inevitably causes major congestion problems at the elevators and the stairwells during lecture transitions. This congestion is already alleviated by assigning different starting times for the academic and professional programmes. However, long waiting times and difficult passages remained to exist. Student flows could also be minimised by maximally spreading the lectures over the day and over the week. However, students and teachers are often dissatisfied with a timetable with free periods in-between. Being not able to attend or to teach lectures consecutively requires more time for travelling towards and away from classrooms. Commuting students especially often prefer to have a compact timetable instead of having free time between lectures. Particularly, days with only one scheduled lecture should be avoided.

Despite the large complexity in building UCTPs, many educational institutes still develop their UCTP manually, which requires a lot of time and creativity of the planners. It is nearly impossible for human planners to solve the enormous puzzle taking into account the constraints and preferences of all stakeholders, let alone to incorporate the resulting student flows. After showing that a monolithic integer programming (IP) model is intractable for a state-of-the-art commercial solver for solving real-life UCTPs taking into account student flows, this chapter presents a two-stage IP approach. In the first stage, lectures are assigned to timeslots taking into account the various constraints and maximising the stakeholders' preferences. The second stage uses the timetable of the previous stage as input and reassigns the classrooms with the objective of minimising the resulting student flows. Through extensive computational tests, we show that, in contrast to a monolithic IP, this two-stage IP approach is capable of finding quality solutions with minimised student flows for real-life UCTPs.

The remainder of this chapter is organised as follows: Section 3.2 discusses related literature of different timetabling problems, modelling and solving techniques. Section 3.3 introduces the timetabling problem of the KU Leuven Campus Brussels. Next, a mathematical formulation for the problem is discussed in Section 3.4, followed by a discussion of the solution method used in Section 3.5. Section 3.6 subsequently applies the model to the data of the Faculty of Economics and Business of KU Leuven Campus Brussels. The latter section also reports on results from tests using data available from the literature. Section 3.10 concludes this chapter and lists directions for future research.

3.2 Literature review

In the following sections, we first give an overview of the solution techniques that have been developed in the literature. Next, we look at the issue of compact timetables, where free hours between consecutive lectures are avoided as much as possible as this is preferred by most students and staff. In the third section, we discuss the literature on the incorporation of student flows into the timetabling problem. In the last section, we outline the approach taken in this chapter.

3.2.1 Solution techniques

Various solution techniques have been proposed for automating the development of course timetables (Chiarandini et al., 2006). Overviews were given by Babaei et al. (2015), Burke and Petrovic (2002), Carter and Laporte (1996, 1998), Lewis (2008), MirHassani and Habibi (2013), Petrovic and Burke (2004), and Schaerf (1999). Below, we discuss three approaches that are most widely used for course timetabling more into detail, namely graph coloring, metaheuristic approaches, and mathematical programming. Other solution approaches include constraint logic programming (e.g., Guéret et al., 1996), case-based reasoning (e.g., Burke et al., 2006a,b), and neural networks (e.g., Carrasco and Pato, 2004).

Graph colouring approaches are often used for timetabling thanks to the ease of implementation (Petrovic and Burke, 2004). In graph coloring approaches the timetabling problem is modelled as a graph in which the nodes correspond to the events (lectures) and the arcs correspond to the event-clash constraints (De Causmaecker et al., 2009). Next, each node needs to be assigned to a color, which represents a timeslot, such that connected nodes have a different color. The goal is to find a solution in which the number of colors used does not exceed the number of available timeslots Lewis (2008). Usually, room assignments are not taken into account in these approaches. Instead, the assignment of lectures to rooms is done after the conflict graph has been constructed and coloured. However, it is possible to include room assignments in the graph colouring process as shown by Redl (2004).

Metaheuristics start with one or a set of solutions which are iteratively improved using local search operators with a protection mechanism that avoids getting stuck in a local optimum. Recent examples of metaheuristic approaches applied to UCTPs can be found in Aladag et al. (2009), De Causmaecker et al. (2009), Lü and Hao (2010), Zhang et al. (2010), and Geiger (2012). A hyperheuristic is a framework in which an upper-level metaheuristic selects the most appropriate heuristic out of a set of lower-level heuristics to solve a particular optimisation problem (Petrovic and Burke, 2004). Hyperheuristics are a growing research topic for tackling timetabling problems (Burke and Petrovic, 2002). Hybrid approaches combine different techniques, for instance Bellio et al. (2012) present a hybrid local search approach, while Gunawan and Kien Ming (2012) propose a hybrid approach that combines Lagrangian relaxation and simulated annealing.

In the past, due to computational difficulties the use of mathematical programming for solving UCTPs has been limited to small size instances. However, thanks to strong advances in computer software and hardware, and in IP formulations, mathematical programming approaches for timetabling problems have become more popular (Daskalaki et al., 2004; Wren, 1996).

Examples of IP formulations for UCTPs can be found in Daskalaki et al. (2004), Dimopoulou and Miliotis (2001), Phillips et al. (2015), and Schimmelpfeng and Helber (2007). One advantage of mathematical programming approaches is the ease of incorporating additional soft constraints (Carter and Laporte, 1998).

Unfortunately, UCTPs continue to cause problems for the planning departments of universities and colleges, because implementations of the proposed solution techniques are scarce. According to McCollum (2007) this is due to incomplete data and the difficulty of incorporating implicit knowledge about the preferences of lecturers and the scheduling policies. There are a few notable exceptions. Daskalaki et al. (2004) apply an integer programming model to the timetabling problem of the department of Electrical and Computer Engineering at the University of Patras. De Causmaecker et al. (2009) use a decomposed metaheuristic approach to solve the timetabling problem for the KaHo Sint-Lieven School of Engineering. Dimopoulou and Miliotis (2001) report on the implementation of a computer system for the joint development of a course and examination timetable at The Athens University of Economics and Business. Schimmelpfeng and Helber (2007) describe the implementation of an integer programming approach to create a complete timetable of all courses for a term at the School of Economics and Management at Hannover University. Badri (1996) develops a two-stage optimisation model to solve a faculty-course-time timetabling problem at United Arab Emirates University. Finally, Al-Yakoob and Sherali (2007) and Al-Yakoob et al. (2010) use integer programming to obtain, respectively, a course and exam timetable at Kuwait University.

As shown in this chapter, computational difficulties inherent to huge IP models can be overcome by decomposing the problem in separate stages that can be solved efficiently with state-of-the-art IP solvers. Badri (1996) also uses a two-stage multi-objective scheduling model for the assignment of faculty members to courses and timeslots. Four types of preferences, each with an associated priority, are grouped into one objective function: the load requirement for each faculty, the satisfaction of the number of available classrooms, the number of evening classes and personal preferences of faculties with respect to course-time assignments. The results of the first stage, the faculty-course assignments, are the input for the second stage. The second stage assigns faculties to timeslots. Burke et al. (2010) propose a general framework for the decomposition of large problems into multiple restricted submodels, which only consider a subset of the objectives at first. The solutions to the subproblems are then aggregated to obtain feasible solutions to the original problem. An advantage to their method is that it is easily implemented using a general IP solver and provides bounds on the solution quality.

3.2.2 Compact timetables

Students and teachers often prefer compact timetables. A compact timetable refers to the absence of free hours between consecutive lectures. Below we describe three contributions that also focus on compact timetables. Santos et al. (2012) include constraints regarding the number of free periods in the timetables of the teachers. A compact and an extended formulation are proposed. The authors use cut and column generation to increase the dual bounds of the extended formulation. Dorneles et al. (2014) present a mixed integer linear programming model to a high school timetabling problem. Among the different requirements that are considered in Brazilian schools, two compactness constraints must be met on a teacher's schedule: the minimisation of working days and the avoidance of idle timeslots. The authors propose a fix-and-optimise heuristic combined with a variable neighbourhood descent method using three different types of decomposition (class, teacher and day). Burke et al. (2010) distinguish four penalty terms: classroom capacity, spread of the lectures of a course, time compactness and classroom stability. The penalisation of classroom capacity and stability is respectively done by penalising classrooms if insufficient seats are available and distinct classrooms are used for different lectures of a course. The spread of the lectures is penalised when the actual spread is smaller than the prescribed spread. For a given curriculum, every time a lecture is not adjacent (an isolated lecture) to another lecture on the same day, time compactness is penalised.

3.2.3 Student flows

As mentioned earlier, the motivation of this chapter is the congestion that occurs in the corridors and at the stairwells at the Faculty of Economics and Business at KU Leuven Campus Brussels and the observation that the timetable has an impact on this. Therefore, we discuss previous work that incorporates the travelling of students between consecutive lectures into the timetabling problem. To the best of our knowledge, the studies in Al-Yakoob and Sherali (2007), Al-Yakoob et al. (2010), Ferdoushi et al. (2014), Hertz (1991), Pongcharoen et al. (2008), and Rudová et al. (2011) are the only ones that, to a limited extent, incorporate student flows. Al-Yakoob and Sherali (2007) present a Mixed Integer Programming (MIP) model for class timetabling problems and consider a related congestion topic. The authors address the problem of parking and traffic congestions for students and faculty members when lectures are inadequately spread over all the available timeslots. Students and faculty members are adequately spread over all the available timeslots by constraints that impose an upper bound on the number of students that follow classes (take exams) during each timeslot. These bounds are not necessary the same for different timeslots. For example, the timeslots when employees and staff start and finish working can have a smaller upper bound. Student flows are also taken into account by Al-Yakoob et al. (2010). The authors present a MIP for exam timetabling and address the same topics: parking and traffic congestions and an inadequately spread of the exams. Therefore, scheduling consecutive exams at distant campuses is undesirable. Parking and traffic congestions are addressed by imposing a constraint on the number of students that can be involved in one exam period. Pongcharoen et al. (2008) present a stochastic optimisation model for the UCTP. They tackle the problem of student movement by a soft constraint ensuring that students attend lectures in the same classroom as much as possible. More recently, Ferdoushi et al. (2014) also consider the minimisation of the movement of students between rooms through soft constraints. The authors develop a modified hybrid particle swarm optimisation approach to a highly constrained realistic environment in the Computer Science and Engineering department of Khulna University of Engineering & Technology, Bangladesh. In both papers, distances between classrooms are not taken into account. Hertz (1991) uses tabu search and graph theory for solving timetabling problems. In addition to the classical feasibility constraints of the timetable, precedence requirements and geographical constraints are taken into account. Precedence requirements are, for example, lectures which should be followed by exercise sessions in the same day. Geographical constraints are related to the distance of two classrooms of two consecutive lectures. The objective function penalises infeasible timetables and pairs of consecutive lectures at distant classrooms. Rudová et al. (2011) use a generic iterative forward search and a branch-and-bound algorithm for a complex university timetabling problem. They try to develop a generic method that is not specifically tailored to a single problem type so that it can be used in practice to solve different concrete timetabling problems with different constraints. The authors also consider the distances between rooms and penalise class assignments that require students or instructors to travel large distances between consecutive lectures.

3.3 Problem description

3.3.1 The KU Leuven Campus Brussels timetabling problem

In the timetabling problem for the KU Leuven Campus Brussels, a weekly timetable needs to be built, where lectures (events) need to be assigned to timeslots and rooms. Two things need to be taken into account:

- Teachers are already assigned to lectures.
- A series is a group of students that have exactly the same timetable. For each series, it is known in advance how many students there are in the series and which lectures they need to attend.

Series are divided in four different types of education: daytime education, morning education, evening education and evening education only on Tuesday and Thursday. The number of available timeslots for a series depends on the type of education of the series. These Educational Preferences (EPF) can be violated by scheduling a lecture at a timeslot when some student series are unavailable to attend this lecture due to their type of education.

Every teacher can submit his teaching preferences regarding the timeslots at the start of the academic year. These Teacher Preferences (TPF) can be violated by scheduling a lecture at a timeslot when a teacher does not prefer to teach this lecture. A first objective is then the minimisation of the violation of the aforementioned teacher preferences and educational preferences.

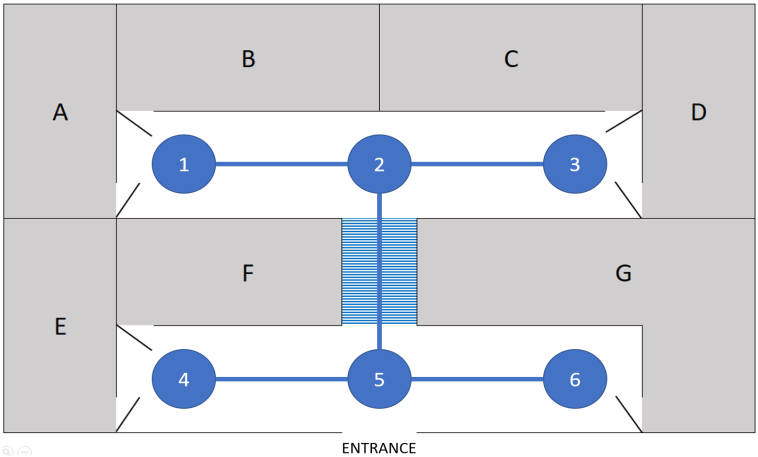
An additional concern in building the timetable for the KU Leuven Campus Brussels, is the congestion caused by students travelling from one classroom to another in between consecutive lectures. A consequence of this congestion is that lectures often start late because it takes a lot of time for students to travel to their next classroom. Therefore, we include the minimisation of the maximum of the travel time for each series of students for all timeslots as a second objective.

3.3.2 Incorporating student flows

To model the flow and resulting travel times of students, we employ some of the modelling techniques used in traffic assignment models. Traffic assignment models try to predict traffic flows and the resulting congestion and travel times on each route in the network, given the estimated number of people who want to travel between different origin-destination pairs (Patriksson, 2015). They represent the road network as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where the set $\mathcal{N}$ of nodes represents destinations or junctions and the set $\mathcal{A}$ of arcs the roads between them. Analogously, to model the flow and resulting travel times of students, we represent the layout of the building by a graph in which a number of adjacent classrooms are grouped into a single node. The number of classrooms that are combined into one node is based on a trade-off between the complexity of the model on the one hand and its realism on the other hand. Next, only nodes which represent physical locations that are adjacent to each other in the actual building are connected by an arc, through which a ‘flow’ of students can pass. This implies that it is possible that students who travel from some classroom A to some classroom B have to pass through multiple arcs to reach their destinations (e.g., if they have to travel from the 3rd floor to the 5th floor, they need to pass through the arc for the stairs between the 3rd and 4th floor first, and then through the arc for the stairs between the 4th and 5th floor). Secondly, in reality it can be that there are multiple routes one can take to reach the same destination from a given location. Therefore, in the model a route choice probability

Figure 3.1: An example of a building layout and the corresponding graph.

In this building, there are 7 classrooms. Rooms A and B are assigned to node 1, rooms C and D to node 3, rooms E and F to node 4 and room G to node 6. Rooms E, F, G and the entrance are on the first floor and rooms A, B, C and D are on the second floor, so arc $(2, 5)$ represents stairs. It is clear that in this specific layout only one route can be taken between any two classrooms.



has to be specified to determine the percentage of students that will cause flow in each possible arc of a certain route. For example, if students can take two possible routes from room A to room B, we need to make an assumption on the percentage of students that will use route 1 and the percentage of students that will use route 2. Figure 3.1 gives an example of layout of a building and the corresponding graph to model the student flows.

An important element in the analysis of traffic assignment models is the notion of congestion (Patriksson, 2015). As traffic volume on a link increases, the average travel speed on the link decreases, until a situation of total congestion is reached. The travel time of a link is modelled with a link performance function, which relates the travel time through a link to the volume of traffic on that link. A similar concept has been observed for pedestrian flows. In the literature this relationship between crowd density and walking speed is called the ‘fundamental diagram’, because of its importance in models describing human walking behaviour (for a general overview of the pedestrian walking behaviour research, see, e.g., Helbing and Johansson (2010) and Kalakou and Moura (2014)). Since we are interested in the travel time of students between classrooms, we will describe this concept in more detail.

Schadschneider and Seyfried (2009) give an overview of the state of empirical research and examine the data relating to the fundamental diagram. Their data only consider planar walking facilities such as corridors and do not apply to stairs. They observe that there is a lot of variance in the data, which has been attributed to a variety of factors. Secondly, there is no consensus whether there is even any significant difference between uni- and multidirectional flows. Therefore, we do not distinguish between uni- and bidirectional flows through an arc.

Based on the data of Schadschneider and Seyfried (2009), we assume the following relationship between crowd density ρ, and walking speed v, i.e.

$$v(\rho) = \frac{\alpha}{\rho}, \tag{3.1}$$

where α is a scaling parameter. The reason for this choice is that the travel time as a function of crowd density is then linear. Another possibility is of course to assume a linear relationship between crowd density and walking speed, and afterwards fit a piecewise linear function to the resulting non-linear travel time function. There are, however, two arguments to support our choice: (i) at high crowd densities, walking speed does not actually reach zero, but ‘turbulent crowd movements’ are observed (Helbing et al., 2007), and (ii) in traffic assignment models it has been observed that asymptotic travel time functions empirically lead to unrealistically high travel times (Boyce et al., 1981). Other empirical studies have looked at the fundamental diagram for the movement on stairs. As expected, walking speed here is lower than on planar surfaces, see e.g. Qu et al. (2014). Therefore, we include a correction term $\gamma \in [0,1]$, such that

$$v(\rho) = \gamma \left(\frac{\alpha}{\rho} \right). \tag{3.2}$$

Then the travel time through arc (i,j) is the length of the physical location represented by this arc divided by the walking speed of the students walking through it; that is, it depends on the total flow of students going through the arc:

$$\mathrm{T}^{\mathrm{arc}}_{tij}(\rho) = \frac{\mathrm{length}_{ij}}{v(\rho)} = \frac{\mathrm{length}_{ij}}{\alpha}\rho + \frac{\mathrm{length}_{ij}}{v_{max}}. \tag{3.3}$$

The second term in equation (3.3) ensures a minimal travel time when the density is zero. Furthermore, the crowd density ρ at time t equals the number of students that travel through arc (i,j) at time t, denoted by F_{tij}, divided by the surface area of the physical location represented by this arc, i.e.

$$\rho = \frac{\mathrm{F}_{tij}}{\mathrm{area}_{ij}}. \tag{3.4}$$

This representation can also be extended to a situation where there are multiple buildings. In this case, it suffices to define an arc between the entrances of each pair of buildings and assume a fixed travel time for that arc, since in public spaces and roads the density is ‘given’ and only marginally influenced by the number of travelling students.

In the computational tests in this chapter, we set $\mathrm{v}_{max} = 1.25$ (Polus et al., 1983), $\alpha = 1$, $\gamma = 1.2^{-1}$. In Chapter 5 we perform a sensitivity analysis to evaluate the impact of changes in the values of these parameters.

3.4 MIP formulation

Building on the explanation of the previous section, we are now able to derive a mixed integer programming formulation for our model to jointly minimise the violation of teacher and educational preferences on the one hand and the travel times of students on the other hand.

3.4.1 Notation

- Constants
  - δ: number of available timeslots in one day. This number is assumed to be the same for every day that lectures can be scheduled.
- Sets
  - $s \in$ S: series of students
  - $l, m \in$ L: lectures. Every lecture takes two hours, is unique and is scheduled once. A course that consists of, for example, two lectures is scheduled twice.
  - $t \in$ T: available timeslots. These are the different time periods that a lecture can be scheduled.
  - $c, d \in$ C: classrooms. Every lecture needs a classroom of the correct type and with sufficient capacity. Different types of classrooms, for example PC-rooms and laboratories, can exist.
  - $r \in$ R: teachers
  - $k \in$ K: days. These are the days (Monday = 1, ..., Friday = 5) that lectures can be scheduled.
  - $p \in$ P: paths
  - $i, j \in$ N: nodes
- Subsets

  - L_s^S: lectures that need to be attended by series s
  - L_c^C: lectures that can be scheduled in classroom c
  - L_r^R: lectures that are taught by teacher r
  - T_k: timeslots on day k
  - C_l: classrooms that can be used to schedule lecture l
  - P_{cd}: all paths that connect room c and room d
- Parameters
  - c_{lt}: penalty cost for scheduling lecture l in timeslot t. These costs include both the teacher preferences and educational preferences.
  - a_{pcd}: percentage of students who use path p to travel from room c to room d
  - b_{ijp}: equals 1 if arc (i, j) is on path p, 0 otherwise
  - n_s: number of students in series s
- Decision variables
  - $x_{ltc} \in \{0,1\}$: equals 1 if lecture l is scheduled at time t in room c, 0 otherwise
  - $U_{tsp} \in [0,1]$: the percentage of students from series s who use path p at time t
  - $F_{tij} \geq 0$: the total student flow through arc (i, j) at time t
  - $T_{tij}^{arc} \geq 0$: the travel time through arc (i, j) at time t
  - $T_{tsp}^{total} \geq 0$: the total travel time for those students of series s that use path p at time t
  - $T_t = \max_{s,p}\{T_{tsp}^{total}\}$: the travel time in timeslot t, i.e. the time when all series of students have reached their destination
  - $T_{max} = \max_t\{T_t\}$

3.4.2 The model

The first set of constraints ensure a feasible timetable. These are hard constraints. Constraint set (3.5) implies that every lecture has to be scheduled in a feasible timeslot and classroom. Constraints (3.6) guarantee that every teacher can teach at most one lecture at a particular timeslot. This lecture is able to be taught by this teacher and is scheduled in a feasible timeslot. Constraint set (3.7) ensures that, for each timeslot, at most one feasible lecture can be scheduled in each classroom. A series of students can only attend one lecture at a time. This is implied by constraint set (3.8).

$$\sum_{t \in \mathrm{T}} \sum_{c \in \mathrm{C}_l} \mathrm{x}_{ltc} = 1 \qquad \forall l \in \mathrm{L} \tag{3.5}$$

$$\sum_{l \in \mathrm{L}_r^{\mathrm{R}}} \sum_{c \in \mathrm{C}_l} \mathrm{x}_{ltc} \leq 1 \qquad \forall r \in \mathrm{R}, \forall t \in \mathrm{T} \tag{3.6}$$

$$\sum_{l \in \mathrm{L}_c^{\mathrm{C}}} \mathrm{x}_{ltc} \leq 1 \qquad \forall t \in \mathrm{T}, \forall c \in \mathrm{C} \tag{3.7}$$

$$\sum_{l \in \mathrm{L}_s^{\mathrm{S}}} \sum_{c \in \mathrm{C}_l} \mathrm{x}_{ltc} \leq 1 \qquad \forall s \in \mathrm{S}, \forall t \in \mathrm{T} \tag{3.8}$$

Labour legislation also enforces a number of constraints regarding the working hours of teachers. The first constraint is that teachers cannot teach more than Δ_1 lectures of two hours per day. Constraints (3.9) ensure these terms of employment. Next, teachers are also not allowed to teach more than Δ_2 lectures consecutively. This is enforced by constraints (3.10). Here $\mathcal{Q}$ denotes a subset of $\Delta_2 + 1$ consecutive timeslots.

$$\sum_{l \in \mathrm{L}_r^{\mathrm{R}}} \sum_{t \in \mathrm{T}_k} \sum_{c \in \mathrm{C}_l} \mathrm{x}_{ltc} \leq \Delta_1 \qquad \forall r \in \mathrm{R}, \forall k \in \mathrm{K} \tag{3.9}$$

$$\sum_{l \in \mathrm{L}_r^{\mathrm{R}}} \sum_{t \in \mathrm{Q}} \sum_{c \in \mathrm{C}_l} \mathrm{x}_{ltc} \leq \Delta_2 \qquad \forall r \in \mathrm{R}, \forall k \in \mathrm{K}, \forall \mathcal{Q} \subset \mathrm{T}_k \tag{3.10}$$

Furthermore, teachers are not allowed to teach in the first timeslot if they taught in the last timeslot on the previous day. Constraint set (3.11) shows how this prohibition is enforced.

$$\sum_{l \in \mathrm{L}_r^{\mathrm{R}}} \sum_{c \in \mathrm{C}_l} (\mathrm{x}_{l,k\delta,c} + \mathrm{x}_{l,k\delta+1,c}) \leq 1 \quad \forall r \in \mathrm{R}, \forall k \in \{1, ..., |\mathrm{K}| - 1\} \tag{3.11}$$

Finally, the legislator does not allow that a docent teaches in the first and last timeslot of a particular day. This is implied by constraint set (3.12).

$$\sum_{l \in \mathrm{L}_r^{\mathrm{R}}} \sum_{c \in \mathrm{C}_l} (\mathrm{x}_{l,1+(k-1)\delta,c} + \mathrm{x}_{l,k\delta,c}) \leq 1 \qquad \forall r \in \mathrm{R}, \forall k \in \mathrm{K} \tag{3.12}$$

Constraints (3.13) are the compactness constraints: these constraints avoid two-hour free periods in the timetables. If a lecture is scheduled at timeslots t and $t+2$ of a particular day, then another lecture needs to be scheduled at timeslot $t+1$ of the same day.

$$\begin{gathered} \sum_{l \in \mathrm{L}_s^{\mathrm{S}}} \sum_{c \in \mathrm{C}_l} (\mathrm{x}_{ltc} + \mathrm{x}_{l,t+2,c} - \mathrm{x}_{l,t+1,c}) \leq 1 \\ \forall s \in \mathrm{S}, \forall k \in \mathrm{K}, \forall t \in \{\delta k + 1, ..., \delta k + |\mathrm{T}_k| - 2\} \end{gathered} \tag{3.13}$$

The second set of constraints determines the student flows. For every series of students s we need to determine which paths they use given the assignment of lectures to classrooms. To this end, U_{tsp} indicates the percentage of students from series s that use path p at time t. The relationship between x_{ltc} and U_{tsp} is then as follows:

$$\begin{gathered} \mathrm{U}_{tsp} \geq \mathrm{a}_{pcd} (\mathrm{x}_{ltc} + \mathrm{x}_{m,t+1,d} - 1) \\ \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l, m \in \mathrm{L}_s^{\mathrm{S}}, \forall p \in \mathrm{P}, \forall c \in \mathrm{C}_l, d \in \mathrm{C}_m \end{gathered} \tag{3.14}$$

However, we also need to include the flow caused by students who leave the building when they do not have class at time $t+1$, and students who enter the building when they did not have class at time t. The following two expressions include the first and second type of flow respectively:

$$\mathrm{U}_{tsp} \geq \mathrm{a}_{pc,exit} \left(\mathrm{x}_{ltc} - \sum_{m \in \mathrm{L}_s^{\mathrm{S}}} \sum_{d \in \mathrm{C}_m} \mathrm{x}_{m,t+1,d} \right) \tag{3.15}$$
$$\forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l \in \mathrm{L}_s^{\mathrm{S}}, \forall p \in \mathrm{P}, \forall c \in \mathrm{C}_l$$

$$\mathrm{U}_{tsp} \geq \mathrm{a}_{pc,exit} \left(\mathrm{x}_{l,t+1,c} - \sum_{m \in \mathrm{L}_s^{\mathrm{S}}} \sum_{d \in \mathrm{C}_m} \mathrm{x}_{mtd} \right) \tag{3.16}$$
$$\forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l \in \mathrm{L}_s^{\mathrm{S}}, \forall p \in \mathrm{P}, \forall c \in \mathrm{C}_l$$

Then, the flow through each arc (i,j) at time t can be calculated as follows:

$$\mathrm{F}_{tij} = \sum_{p \in \mathrm{P}} \sum_{s \in \mathrm{S}} \mathrm{n}_s \mathrm{b}_{ijp} \mathrm{U}_{tsp} \qquad \forall t \in \mathrm{T}, \forall i,j \in \mathrm{N} \tag{3.17}$$

To assure that crowd density does not reach hazardous levels (see e.g. Helbing et al. (2007)), the flow through an arc cannot exceed a predetermined maximum level:

$$\mathrm{F}_{tij} \leq \mathrm{F}_{max} \qquad \forall t \in \mathrm{T}, \forall i,j \in \mathrm{N} \tag{3.18}$$

Now the travel time through arc (i,j) at time t is derived from the flow as follows

$$\mathrm{T}^{\mathrm{arc}}_{tij} = \frac{\mathrm{length}_{ij}}{\alpha} \frac{\mathrm{F}_{tij}}{\mathrm{area}_{ij}} + \frac{\mathrm{length}_{ij}}{\mathrm{v}_{max}} \qquad \forall t \in \mathrm{T}, \forall i,j \in \mathrm{N} \tag{3.19}$$

where the correction factor γ needs to be included if arc (i,j) represents stairs. Then, the travel time of a given series s from their first classroom c to their next classroom d is given by the sum of the individual travel times of each arc (i,j) that is on path p used by that series. When there are multiple paths that students can take, the travel time of the series is taken as the maximum of the travel times over all possible paths. To model this, the following two constraints are added:

$$-\sum_{(i,j)} \mathrm{b}_{ijp} \mathrm{T}^{\mathrm{arc}}_{tij} + \mathrm{T}^{\mathrm{total}}_{tsp} \leq M \left(2 - \mathrm{x}_{ltc} - \mathrm{x}_{m,t+1,d}\right) \qquad \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l, m \in \mathrm{L}^{\mathrm{S}}_{s}, \forall c \in \mathrm{C}_{l}, d \in \mathrm{C}_{m}, \forall p \in \mathrm{P}_{cd} \tag{3.20}$$

$$\sum_{(i,j)} \mathrm{b}_{ijp} \mathrm{T}^{\mathrm{arc}}_{tij} - \mathrm{T}^{\mathrm{total}}_{tsp} \leq M \left(2 - \mathrm{x}_{ltc} - \mathrm{x}_{m,t+1,d}\right) \qquad \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l, m \in \mathrm{L}^{\mathrm{S}}_{s}, \forall c \in \mathrm{C}_{l}, d \in \mathrm{C}_{m}, \forall p \in \mathrm{P}_{cd} \tag{3.21}$$

where M is a large number. These constraints work as follows: if two consecutive lectures l and m, which are followed by series s, are planned in rooms c and d respectively, then (3.20) and (3.21) reduce to:

$$-\sum_{(i,j)} \mathrm{b}_{ijp} \mathrm{T}^{\mathrm{arc}}_{tij} + \mathrm{T}^{\mathrm{total}}_{tsp} \leq 0 \tag{3.22}$$

$$\sum_{(i,j)} \mathrm{b}_{ijp} \mathrm{T}^{\mathrm{arc}}_{tij} - \mathrm{T}^{\mathrm{total}}_{tsp} \leq 0, \tag{3.23}$$

which is equivalent to $\mathrm{T}^{\mathrm{total}}_{tsp} = \sum_{(i,j)} \mathrm{b}_{ijp} \mathrm{T}^{\mathrm{arc}}_{tij}$. This means that the travel time of this series over path p should equal the sum of the individual travel times of all arcs (i, j) that are on path p. On the other hand, if at least one of the variables x_{ltc} and $\mathrm{x}_{m,t+1,d}$ equals 0, then

$$-\sum_{(i,j)} \mathrm{b}_{ijp} \mathrm{T}^{\mathrm{arc}}_{tij} + \mathrm{T}^{\mathrm{total}}_{tsp} \leq M \tag{3.24}$$

$$\sum_{(i,j)} \mathrm{b}_{ijp} \mathrm{T}^{\mathrm{arc}}_{tij} - \mathrm{T}^{\mathrm{total}}_{tsp} \leq M, \tag{3.25}$$

such that nothing is implied for $\mathrm{T}^{\mathrm{total}}_{tsp}$, i.e. $\mathrm{T}^{\mathrm{total}}_{tsp}$ can be set to 0. Furthermore, there can be at most one combination of x_{ltc} and $\mathrm{x}_{\mathrm{m,t+1,d}}$ for which both variables are equal to 1, so $\mathrm{T}^{\mathrm{total}}_{tsp}$ is then uniquely defined.

Finally, the travel time T_{t} in timeslot t is given by:

$$\mathrm{T}^{\mathrm{total}}_{tsp} \leq \mathrm{T}_{t} \qquad \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall p \in \mathrm{P} \tag{3.26}$$

and the maximum travel time over all timeslots by

$$\mathrm{T}_t \leq \mathrm{T}_{max} \qquad \forall t \tag{3.27}$$

When calculating the travel time in a given timeslot (T_t), we do not include series who do not have lecture at time $t+1$, because they leave the building and consequently don't have to arrive at their next lecture as quickly as possible. Similarly, we do not include series who do not have lecture at time t, because they enter the building from outside, so they naturally enter in waves instead of all simultaneously; also, they can come earlier to be in class on time. We also remark that two consecutive timeslots for which there is a lunch break in between or that are on two consecutive days should obviously not be included.

The objective function then consists of two parts: the minimisation of the violation of the teacher and educational preferences on the one hand, and the minimisation of the travel times on the other hand. We can choose to either minimise the average (or equivalently, the sum) of the travel times in each timeslot:

$$\text{minimise} \quad \lambda \sum_{l \in \mathrm{L}} \sum_{t \in \mathrm{T}} \sum_{c \in \mathrm{C}} \mathrm{c}_{lt} \mathrm{x}_{ltc} + (1-\lambda) \sum_{t \in \mathrm{T}} \mathrm{T}_t \tag{3.28}$$

or minimise the maximum over all timeslots:

$$\text{minimise} \quad \lambda \sum_{l \in \mathrm{L}} \sum_{t \in \mathrm{T}} \sum_{c \in \mathrm{C}} \mathrm{c}_{lt} \mathrm{x}_{ltc} + (1-\lambda)\, \mathrm{T}_{max} \tag{3.29}$$

The weight of $\lambda \in [0,1]$ reflects the importance of each of the respective separate terms in the objective function. This parameter should be set by the university based on the relative importance they attach to each term.

3.5 Solution approach

We have tried to solve the mathematical model presented in Section 3.4 directly using an integer programming solver. However, the 'Big M' constraints make the problem formulation intractable for real-world instances. Therefore, we use a two-stage integer programming approach, which is an adaptation of the decomposition method of Burke et al. (2010). The first stage then finds a timetable that is feasible with respect to the hard constraints and minimises the violation of the teacher and educational preferences. Next, the second stage uses the timetable obtained in stage 1 as input and minimises the student flows by reassigning lectures to classrooms.

The first stage model uses the same decision variable x_{ltc} as the monolithic model. It consists of equations (3.5) - (3.13) and its objective function is the first part of equation (3.27). The second stage model uses a variable w_{lc} which equals 1 if lecture l is assigned to room c and 0 otherwise. Let k_{lt} equal 1 if lecture l is planned in timeslot t in the solution of the first stage model and 0 otherwise. The second stage model is now given by:

$$\text{minimise} \quad \mathrm{T}_{max} \qquad \text{or} \qquad \text{minimise} \quad \sum_{t \in \mathrm{T}} \mathrm{T}_t \tag{3.30}$$

subject to:

$$\sum_{c \in \mathrm{C}_l} \mathrm{w}_{lc} = 1 \qquad \forall l \in \mathrm{L} \tag{3.31}$$

$$\sum_{l \in \mathrm{L}_c^{\mathrm{C}}} \mathrm{k}_{lt} \mathrm{w}_{lc} \leq 1 \qquad \forall t \in \mathrm{T}, \forall c \in \mathrm{C} \tag{3.32}$$

$$\mathrm{U}_{tsp} \geq \mathrm{a}_{pcd}\left(\mathrm{k}_{lt}\mathrm{w}_{lc} + \mathrm{k}_{mt}\mathrm{w}_{md} - 1\right) \\ \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l, m \in \mathrm{L}_s^{\mathrm{S}}, \forall p \in \mathrm{P}, \forall c \in \mathrm{C}_l, d \in \mathrm{C}_m \tag{3.33}$$

$$\mathrm{U}_{tsp} \geq \mathrm{a}_{pc,exit}\left(\mathrm{k}_{lt}\mathrm{w}_{lc} - \sum_{m \in \mathrm{L}_s^{\mathrm{S}}} \sum_{d \in \mathrm{C}_m} \mathrm{k}_{m,t+1}\mathrm{w}_{md}\right) \\ \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l \in \mathrm{L}_s^{\mathrm{S}}, \forall p \in \mathrm{P}, \forall c \in \mathrm{C}_l \tag{3.34}$$

$$\mathrm{U}_{tsp} \geq \mathrm{a}_{pc,exit}\left(\mathrm{k}_{l,t+1}\mathrm{w}_{lc} - \sum_{m \in \mathrm{L}_s^{\mathrm{S}}} \sum_{d \in \mathrm{C}_m} \mathrm{k}_{mt}\mathrm{w}_{md}\right) \\ \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l \in \mathrm{L}_s^{\mathrm{S}}, \forall p \in \mathrm{P}, \forall c \in \mathrm{C}_l \tag{3.35}$$

$$\mathrm{F}_{tij} = \sum_{p \in \mathrm{P}} \sum_{s \in \mathrm{S}} \mathrm{n}_s \mathrm{b}_{ijp} \mathrm{U}_{tsp} \qquad \forall t \in \mathrm{T}, \forall i, j \in \mathrm{N} \tag{3.36}$$

$$\mathrm{F}_{tij} \leq \mathrm{F}_{max} \qquad \forall t \in \mathrm{T}, \forall i, j \in \mathrm{N} \tag{3.37}$$

$$\mathrm{T}_{tij}^{\mathrm{arc}} = \frac{\mathrm{length}_{ij}}{\alpha} \frac{\mathrm{F}_{tij}}{\mathrm{area}_{ij}} + \frac{\mathrm{length}_{ij}}{\mathrm{v}_{max}} \qquad \forall t \in \mathrm{T}, \forall i, j \in \mathrm{N} \tag{3.38}$$

$$-\sum_{(i,j)} \mathrm{b}_{ijp}\mathrm{T}_{tij}^{\mathrm{arc}} + \mathrm{T}_{tsp}^{\mathrm{total}} \leq M\left(2 - \mathrm{k}_{lt}\mathrm{w}_{lc} - \mathrm{k}_{mt}\mathrm{w}_{md}\right) \\ \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l, m \in \mathrm{L}_s^{\mathrm{S}}, \forall c \in \mathrm{C}_l, d \in \mathrm{C}_m, \forall p \in \mathrm{P}_{cd} \tag{3.39}$$

$$\sum_{(i,j)} \mathrm{b}_{ijp}\mathrm{T}_{tij}^{\mathrm{arc}} - \mathrm{T}_{tsp}^{\mathrm{total}} \leq M\left(2 - \mathrm{k}_{lt}\mathrm{w}_{lc} - \mathrm{k}_{mt}\mathrm{w}_{md}\right) \\ \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall l, m \in \mathrm{L}_s^{\mathrm{S}}, \forall c \in \mathrm{C}_l, d \in \mathrm{C}_m, \forall p \in \mathrm{P}_{cd} \tag{3.40}$$

$$\mathrm{T}_{tsp}^{\mathrm{total}} \leq \mathrm{T}_t \qquad \forall t \in \mathrm{T}, \forall s \in \mathrm{S}, \forall p \in \mathrm{P} \tag{3.41}$$

$$\mathrm{T}_t \leq \mathrm{T}_{max} \qquad \forall t \tag{3.42}$$

Constraints (3.31) ensure that each lecture is assigned to exactly one room. Constraints (3.32) ensure that in each timeslot at most one lecture is assigned to every room. Constraints (3.33) - (3.42) are the travel time constraints based on the new decision variable definition.

It is thus a hierarchical approach where the first objective is solved to global optimality first, and only then the second objective is improved as much as possible without changing the value of the first objective. This reflects the fact that the first objective is deemed considerably more important than the second one. An advantage of the two-stage model is also that the second stage is guaranteed to find a feasible solution since the first stage already ensures the feasibility of classroom assignments.

3.6 Experimental results

This section discusses the input data of the two-stage model for the KU Leuven FEB Campus Brussels and shows the results of the two-stage model. In addition, this section briefly describes the adaptation of the data available from the literature, as well as the results obtained for the two-stage model with these instances.

3.6.1 Data of the KU Leuven FEB Campus Brussels

An academic year consists of two semesters with 13 weeks of teaching per semester. Lectures can be scheduled from Monday till Friday. Every class takes two hours. This permits an efficient use of the classrooms. Six different timeslots can be distinguished: from 8h30 to 10h30, from 10h30 to 12h30, from 13h30 to 15h30, from 15h30 to 17h30, from 17h30 to 19h30 and from 19h30 to 21h30. There is a lunch break between the second and the third timeslot.

Every course has a certain number of Teaching Hours (THs): 13, 26, 39 or 52. The number of THs determines how many course lectures need to be scheduled per week. A course of 26 THs and 52 THs is scheduled once and twice per week respectively. One lecture per two weeks is needed for

Table 3.1: Number of series that attends a particular type of education.

Type of education	Number of series
Daytime education	365
Morning education	23
Evening education	41
Evening education on Tuesday and Thursday	7

a course of 13 THs. A course of 39 THs needs to be scheduled alternately once or twice per week. Courses of 13 and 39 THs that are attended by the same series can be coupled to each other. Two courses of 13 THs can use the same timeslot every week by scheduling these courses alternately in this particular timeslot. The same can be done for two courses of 39 THs: two timeslots of one week can be used for scheduling two courses. The course scheduled in one of these timeslots alternates weekly. The availability of the teachers needs to be taken into account when courses are coupled. The coupling of courses allows to timetable one week and using this timetable for the whole semester.

The FEB Campus Brussels offers academic programmes, preparatory programmes and bridging programmes. There exist 436 series in total. Table 3.1 shows the number of series that attend a particular type of education. As can be deducted from this table, the majority of the series attends daytime education. The available timeslots for each type of education are shown in Table 3.2.

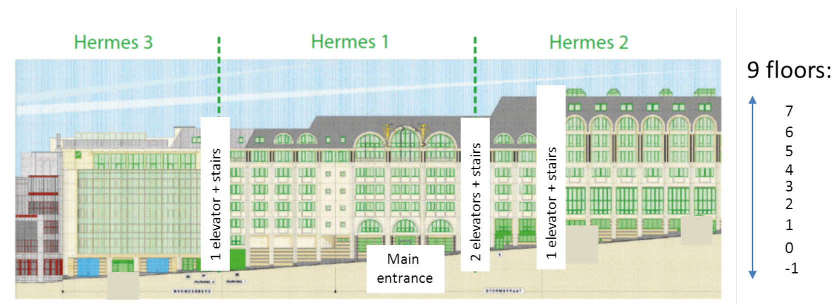
The compactness constraints given by equation (3.13) need to be built only for series that attend daytime education. For other types of education, there can never be free timeslots in between. There are a total of three compactness constraints for the case of 5 and 6 timeslots per day, i.e.

Table 3.2: Available timeslots for each type of education.

Lunch breaks are scheduled between the second and third timeslot.

	Monday						Tuesday						Wednesday						Thursday						Friday					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
Daytime education	x	x	x	x	x		x	x	x	x	x		x	x	x	x	x		x	x	x	x	x		x	x	x	x	x	
Morning education	x	x					x	x					x	x					x	x					x	x				
Evening education					x	x					x	x					x	x					x	x					x	x
Evening education Tuesday and Thursday											x	x											x	x						

Figure 3.2: The FEB campus Brussels building.



between timeslot 1 and 3, between timeslot 2 and 4, and between timeslot 3 and 5.

For these series, 396 lectures need to be scheduled. A PC-room is required for 31 lectures. The other lectures can be scheduled in normal classrooms. The FEB has 56 classrooms at its disposal: 9 PC-rooms and 47 standard classrooms. As shown in Figure 3.2 these rooms are distributed over 9 floors (from -1 (cellar) till 7) in one building, called the Hermes building.

All the lectures are taught by 171 teachers. Teacher working time regulations state that teachers cannot teach more than eight hours per day or more than six hours consecutively. This implies that $\Delta_1 = 4$ in constraint (3.9) and $\Delta_2 = 3$ in constraint (3.10). Four types of teachers are distinguished to determine teacher preferences: guest speakers, researchers, part-time and full-time teachers. Every teacher can submit his teaching preferences regarding the timeslots at the start of the academic year. The penalty cost for violating the teacher preferences depends on the type of the teacher. Guest speakers have the highest freedom regarding their preferences which translates to a preference violation cost of 20. Next, active researchers have a preference violation cost of 15. Finally, part-time and full-time teachers receive the lowest weights, 10 and 5 respectively. A cost of 1000 is incurred for a lecture when at least one series cannot attend this lecture because of the series' unavailability at the timeslot under consideration. These are the penalty costs for the violation of the educational preferences. There is no 'correct' value for the penalty value for the violation of each type of preferences; it should be set by management considerations on the importance attached to each of them. In the case of the FEB timetable, the satisfaction of the educational preferences is deemed much more important than the satisfaction of teacher preferences.

3.6.2 Data from the literature

International timetabling competitions (ITC) regularly provide a number of benchmark problems that are widely used in timetabling literature to develop computational experiments. Badoni et al. (2014) describe a hybrid algorithm combining a genetic algorithm with local search using events based on groupings of students to solve a UCTP. The authors apply their algorithm on instances based on the datasets from the first international timetabling competition (ITC2002). Hao and Benlic (2011) combine tabu search and IP for finding new lower bounds for the ITC2007 curriculum based course timetabling problem. Phillips et al. (2015) validate their IP model for solving a UCTP through a real-life case at the University of Auckland and on instances from the ITC2007. Dorneles et al. (2014) use the ITC2011 instances to test their algorithm dedicated to a high school timetabling problem.

In order to test and validate the two-stage model, we adapt the first set of 7 instances that were used for the Curriculum-based Course Timetabling Track of the International Timetabling Competition in 2007-08 (ITC2007) (Bonutti et al., 2012). These are real cases taken mainly from the University of Udine. Information on the curriculum-based course timetabling problem of the ITC2007 is available at `http://www.cs.qub.ac.uk/itc2007/curriculmcourse/report/curriculumtechreport.pdf`. The instances themselves can be downloaded from `http://www.cs.qub.ac.uk/itc2007/Login/SecretPage.php`. Since the objective of the timetabling problem of the FEB Campus Brussels is novel in the literature (minimisation of the travel times between lectures in consecutive timeslots), we do not intend to compare the results or validate the solutions obtained with the ones available in the web application for benchmarking.

Table 3.3 shows the main features of the *comp* instances: number of available timeslots in one day (δ), number of days ($|\mathrm{K}|$), number of lectures ($|\mathrm{L}|$), number of classrooms ($|\mathrm{C}|$), number of teachers ($|\mathrm{R}|$), number of series of students ($|\mathrm{S}|$), and number of students that attends a particular type of education ($|\mathrm{S}^D|$ for daytime education, $|\mathrm{S}^M|$ for morning education, $|\mathrm{S}^E|$ for evening education, and $|\mathrm{S}^{ETT}|$ for evening education on Tuesday and Thursday). The available timeslots for each type of education for the cases with five timeslots in one day ($\delta = 5$) are shown in Table 3.4 (the cases with six available timeslots per day are described in Table 3.2). For simplicity evening education on Tuesday and Thursday is removed from the table since this type of education uses the same timeslots as evening education but only on Tuesday and Thursday.

Table 3.3: Description of the instances tested.

Instance	δ	|K|	|L|	|C|	|P|	$\|S^D\|$	$\|S^M\|$	$\|S^E\|$	$\|S^{ETT}\|$	|S|
FEB2012	6	5	396	56	171	365	23	41	7	436
comp01	6	5	160	7	24	13	1	0	0	14
comp02	5	5	283	16	71	61	6	3	0	70
comp03	5	5	251	16	61	48	13	7	0	68
comp04	5	5	286	18	70	29	15	10	3	57
comp05	6	6	152	9	47	70	68	0	1	139
comp06	5	5	361	18	87	54	12	4	0	70
comp07	5	5	434	20	99	60	6	10	1	77

Table 3.4: Available timeslots for each type of education on a single day with 5 available timeslots.

The lunch breaks are scheduled between the second and third timeslot.

	1	2	3	4	5
Daytime education	x	x	x	x	x
Morning education	x	x			
Evening education				x	x

Information not available in *comp* instances was randomly generated according to the distribution of the corresponding information in the dataset of the FEB Campus Brussels. More specifically, for each course, the type of teacher (guest speaker, researchers, part-time or full-time teachers) was randomly generated in order to fix the penalty cost for the violation of the teacher preferences. A type of education was assigned to each series in such a way that the number of available timeslots are sufficient to schedule all the lectures that need to be attended by the corresponding series.

Finally, in the *comp* instances the rooms are distributed among buildings. Courses at The FEB Campus Brussels take place in only one building (as shown in Figure 3.2) and the congestion of the students at the escalators and corridors is a real problem. In order to test the models in this thesis, we generate a set of five buildings, named B-8-1, B-8-2, B-16-1, B-16-2, and B-20. Detailed information on these buildings can be found in Appendix B. In the tests in this chapter, we use building B-8-2 for instance comp01, building B-16-1 for instances comp02, comp03, and comp05, and building B-20-1 for instances comp04, comp06, and comp07. Classrooms in the timetable instance are randomly assigned to one of the rooms of the building.

3.6.3 Results

The monolithic model and the two-stage model of Section 3.4 are programmed in C++ and compiled with Microsoft Visual Studio 2017. The callable library of ILOG CPLEX 12.6.3 is used as a MIP solver. The code is executed on a PC with an AMD Ryzen 7 1700X processor of 3.40 GHz and a RAM of 16 GB. The C++ code for both models, as well as detailed information on the problem instances, can be found at the following website: `https://github.com/HendrikBV/ModelsPhDThesisChapter3`.

The second stage model requires a lot of variables and constraints to represent the flow through the different arcs in the graph. To reduce memory requirements, we therefore split our model into a number of submodels, where each submodel solves the problem for the morning or afternoon of each different day respectively. This is possible since no flows occur between the lunch breaks or between different days, so that the classroom assignments in one submodel do not affect flows in another submodel.

The computational results are shown in Tables 3.5, 3.6, and 3.7. The first stage model can be solved to optimality relatively quickly for all eight problem instances. For the second stage model, the small problem instances can be solved to optimality quickly, while for the larger instances

Table 3.5: Results for the first stage of the two-stage model.

Instance	Building	Obj.	Opt. gap (%)	Time (s)
FEB2012	B-KUL	10,080	0	122
comp01	B-8-2	0	0	4
comp02	B-16-1	150	0	402
comp03	B-16-1	17,180	0	84
comp04	B-20	3055	0	12
comp05	B-16-1	168,235	0	626
comp06	B-20	5130	0	44
comp07	B-20	2000	0	83

the computation time grows exponentially. The reason for this is that the ‘Big M’ constraints of equations (3.20)-(3.21) provide poor bounds in the LP relaxation of the problem. The second stage model is able to find considerable improvements compared to the solution from the first stage model for all *comp* instances. Unfortunately, CPLEX is unable to solve the second-stage model for the FEB instance, as the high number of lectures and more importantly series of students in the KUL instance, in combination with the large building with 56 classrooms, lead to an intractable number of variables and especially constraints.

There is also a significant difference in the required computation time between minimising the maximum of the travel times over all timeslots compared to minimising the average of the travel times over all timeslots. The latter requires significantly more computation time for instances comp02 (532 s vs. 920 s), comp04 (1025 s vs. 3934 s), comp06 (6424 s vs. 11,122 s), and comp07 (10,816 vs. 14,340). Table 3.8 compares the results of both types of objective function in more detail. For most instances, there is a clear trade-off between the two objectives.

Table 3.6: Results for the second stage of the two-stage model when T_{max} is minimised.

Each of the subproblems of the second stage MIP had a time limit of 3600s. 'Not opt.' indicates the number of subproblems that could not be solved to optimality within the time limit. 'Init obj.' refers to the initial objective value of the solution provided by the first stage, while 'Obj. BFS' refers to the objective value of the best found solution.

Instance	Building	Init. obj.	Obj. BFS	Not opt.	Time (s)
FEB2012	B-KUL				
comp01	B-8-2	2025	161	0/10	12
comp02	B-16-1	7534	1057	0/10	532
comp03	B-16-1	31,840	5110	0/10	309
comp04	B-20	32,312	776	0/10	1025
comp05	B-16-1	55,277	42,939	0/12	106
comp06	B-20	39,683	3987	1/10	6424
comp07	B-20	54,408	6396	1/10	10,816

Table 3.7: Results for the second stage of the two-stage model when $\frac{1}{|\mathrm{T}|}\sum_{t\in\mathrm{T}}\mathrm{T}_t$ is minimised.

Each of the subproblems of the second stage MIP had a time limit of 3600s. 'Not opt.' indicates the number of subproblems that could not be solved to optimality within the time limit. 'Init obj.' refers to the initial objective value of the solution provided by the first stage, while 'Obj. BFS' refers to the objective value of the best found solution.

Instance	Building	Initial obj.	Obj. BFS	Not opt.	Time (s)
FEB2012	B-KUL				
comp01	B-8-2	910	42	0/10	12
comp02	B-16-1	5966	205	0/10	920
comp03	B-16-1	24,555	1160	0/10	472
comp04	B-20	19,151	344	0/10	3934
comp05	B-16-1	26,422	10,527	0/12	101
comp06	B-20	36,627	1316	1/10	11,122
comp07	B-20	50,384	2670	3/10	14,340

Table 3.8: Comparison between the two types of objectives for the second stage model.

'Min max' refers to the objective function where T_{max} is minimised and 'Min avg' refers to the objective function where $\frac{1}{|T|}\sum_{t\in T} T_t$ is minimised.

Instance	Building	$\frac{1}{\lvert T\rvert}\sum_{t\in T} T_t$		T_{max}	
		Min max	Min avg	Min max	Min avg
comp01	B-8-2	49	42	161	161
comp02	B-16-1	270	205	1057	1339
comp03	B-16-1	1413	1160	5110	5110
comp04	B-20	422	344	776	785
comp05	B-16-1	12,433	10,527	42,939	42,939
comp06	B-20	1437	1316	3987	3987
comp07	B-20	2976	2670	6396	7780

3.7 The impact of timetabling on the efficient evacuation of a building

The importance of the efficient evacuation of a building in the event of an emergency such as a fire cannot be overstated. In the U.S. alone, there were an estimated average of 15,400 structure fires in high-rise buildings and 5690 in educational properties per year, resulting in an overall average of 47 deaths and 615 injuries (National Fire Protection Association, 2016).

As a result, the study of the building evacuation problem has increasingly received attention from researchers over the last decade. Most of the existing optimisation models minimise the maximum egress time, i.e. the time that the last person reaches a safe location, by determining the optimal evacuation routes for all people (Vermuyten et al., 2016a). In practice, each room in the building is assigned to a certain (emergency) exit and the people present in the respective room are expected to follow a certain route through the building to their given exit (e.g., Cepolina, 2005, 2009). Some authors consider so-called phased evacuation, which tries to reduce congestion and to improve overall egress times, by letting different parts of the building start their evacuation at a different time (Abdelghany et al., 2014; Cepolina, 2009; Ng and Waller, 2010).

However, to the best of our knowledge, no articles currently include the impact of timetabling decisions on the evacuation process (Vermuyten et al., 2016a). In the same way that the timetable impacts the flow of students who travel from lectures in one timeslot to lectures in the next timeslot, the timetable also impacts the student flows during an evacuation. By reassigning lectures to different timeslots peaks in the number of people present could be avoided, which leads to less congestion and consequently a lower egress time. Analogously, by optimising the assignment of lectures to classrooms, congestion on certain routes within the building could be minimised, again with a lower egress time as a result.

We can extend the model of Section 3.4 to include evacuations. Section 3.7.1 describes how the MIP model can be extended to include evacuations. Section 3.7.2 discusses the computational results.

3.7.1 MIP formulation

In this case, all students that attend the same lecture travel from the room to which the lecture is assigned towards the exit of the building. We thus define $\mathrm{U}_{ltp}^{\mathrm{evac}}$ as the percentage of students in lecture l that use path p at time t to walk towards the exit of the building. The relationship between

x_{ltc} and U^{evac}_{ltp} is then as follows:

$$U^{evac}_{ltp} \geq a_{pc,exit} x_{ltc} \qquad \forall t \in T, \forall l \in L, \forall c \in C_l, \forall p \in P \tag{3.43}$$

The flow through each arc (i, j) at time t is then given by:

$$F^{evac}_{tij} = \sum_{p \in P} \sum_{l \in L} n_l b_{ijp} U^{evac}_{ltp} \qquad \forall t \in T, \forall i, j \in N, \tag{3.44}$$

where n_l denotes the number of students that attend lecture l.

The travel time through arc (i, j) at time t can be derived from the flow as follows:

$$T^{evac,arc}_{tij} = \frac{length_{ij}}{\alpha} \frac{F^{evac}_{tij}}{area_{ij}} + \frac{length_{ij}}{v_{max}} \qquad \forall t \in T, \forall i, j \in N, \tag{3.45}$$

where the correction factor γ needs to be included if arc (i, j) represents stairs.

The evacuation time for the students in a given lecture l is the sum of the individual travel times of each arc (i, j) that is on path p used by that group of students. We model this using the following two constraints:

$$-\sum_{(i,j)} b_{ijp} T^{evac,arc}_{tij} + T^{evac,total}_{tlp} \leq M\left(1 - x_{ltc}\right) \qquad \forall t \in T, \forall l \in L, \forall c \in C_l, \forall p \in P_{c,exit} \tag{3.46}$$

$$\sum_{(i,j)} b_{ijp} T^{evac,arc}_{tij} - T^{evac,total}_{tlp} \leq M\left(1 - x_{ltc}\right) \qquad \forall t \in T, \forall l \in L, \forall c \in C_l, \forall p \in P_{c,exit} \tag{3.47}$$

The maximum evacuation time in timeslot t is then given by:

$$T^{evac,total}_{tlp} \leq T^{evac}_{t} \qquad \forall t \in T, \forall l \in L, \forall p \in P \tag{3.48}$$

and the maximum over all timeslots by

$$T^{evac}_{t} \leq T^{evac}_{max} \qquad \forall t \in T \tag{3.49}$$

Table 3.9: Results for the second stage of the two-stage model when $\frac{1}{|\mathrm{T}|}\sum_{t\in\mathrm{T}}\mathrm{T}_t^{\mathrm{evac}}$ is minimised.

'Init obj.' refers to the initial objective value of the solution provided by the first stage, while 'Obj. BFS' refers to the objective value of the best found solution.

Instance	Building	Initial obj.	Obj. BFS	Time (s)
FEB2012	B-KUL			
comp01	B-8-2	340	243	26
comp02	B-16-1	3693	2107	153
comp03	B-16-1	3633	1618	137
comp04	B-20	9021	4510	217
comp05	B-16-1	2908	2773	217
comp06	B-20	12,835	5167	287
comp07	B-20	12,696	7103	407

We can again minimise either the sum of the evacuation times over all timeslots

$$\text{minimise} \quad \sum_{t\in\mathrm{T}}\mathrm{T}_t^{\mathrm{evac}} \tag{3.50}$$

or the maximum evacuation time over all timeslots

$$\text{minimise} \quad \mathrm{T}_{max}^{\mathrm{evac}} \tag{3.51}$$

3.7.2 Results

We use the same timetable instances and buildings to test the model with respect to the minimisation of the evacuation times. We take as the objective function the average evacuation time over all timeslots, i.e. $\frac{1}{|\mathrm{T}|}\sum_{t\in\mathrm{T}}\mathrm{T}_t^{\mathrm{evac}}$. Since the evacuation in every timeslot is independent of the evacuations in other timeslots, we can again split the second stage model into a number of subproblems to reduce the complexity. We thus solve a subproblem for every timeslot.

All submodels could be solved to optimality for all problem instances. The results show that the model is able to significantly improve the evacuation times and thus the safety of students by simply improving upon the room assignments. The required computation time for the second stage is much smaller when the evacuation times are minimised compared to when the travel times between lectures in consecutive timeslots are minimised. This is because in the former problem scheduling decisions in one timeslot do not impact the evacuation time of other timeslots, while in the latter problem those decisions are linked.

3.8 Trade-offs, solution quality and scalability

In this section, we look at the trade-offs between the minimisation of the different objectives. We also discuss the solution quality and scalability of the two-stage model compared to the monolithic model. There are three types of trade-offs: (i) the trade-off between the optimisation of the scheduling preferences and the minimisation of the travel times, (ii) the trade-off between the optimisation of the scheduling preferences and the minimisation of the evacuation times, and (iii) the trade-off between the minimisation of the travel times and the minimisation of the evacuation times.

We generate a small instance (ML) that can be solved by the monolithic model and test it again with building B-8-2. We first find the optimal solution with respect to one objective. Then we iteratively solve the problem with the objective of minimising the other objective while we set a con-

Table 3.10: Trade-off between the minimisation of the penalty score for the scheduling preferences and the minimisation of the travel times.

Model		Preferences	Travel times	Time (s)
Monolithic unconstrained	$\lambda = 1$	125		0.33
	$\lambda = 0.5$			>3600
	$\lambda = 0$			>3600
Monolithic constrained		125	3276	13.45
		130	3080.2	19.05
		135	2777.8	47.66
		140	2405.6	64.33
		145	2207.3	58.41
Two-stage		125	3382.4	0.67

straint on the maximally allowable value for the first objective, where each time the right-hand side of the constraint is increased slightly. This way we can derive the set of Pareto optimal solutions for the problem instance.

Table 3.10 shows the results for the trade-off between the minimisation of the penalty score for the scheduling preferences and the minimisation of the travel times between lectures in consecutive timeslots. First, there is indeed a trade-off between the two objective function measures. Moreover, the trade-off is not convex, as is clear from Figure 3.3. The two-stage model finds a solution with an optimality gap of 3.2 percent. Secondly, the computation time of the two-stage model is up to 100 times smaller than that of the constrained monolithic model. For the unconstrained monolithic model, the difference is much larger still. We can thus conclude that our two-stage approach achieves a significant reduction in required computation time and at the same time is still able to obtain good quality solutions that are close to one of the endpoints on the Pareto-optimal frontier.

Figure 3.4 shows the same trend for the trade-off between the minimisation of the penalty score for the scheduling preferences and the minimisation

Figure 3.3: Visualisation of the trade-off between the minimisation of the penalty score for the scheduling preferences and the minimisation of the travel times.

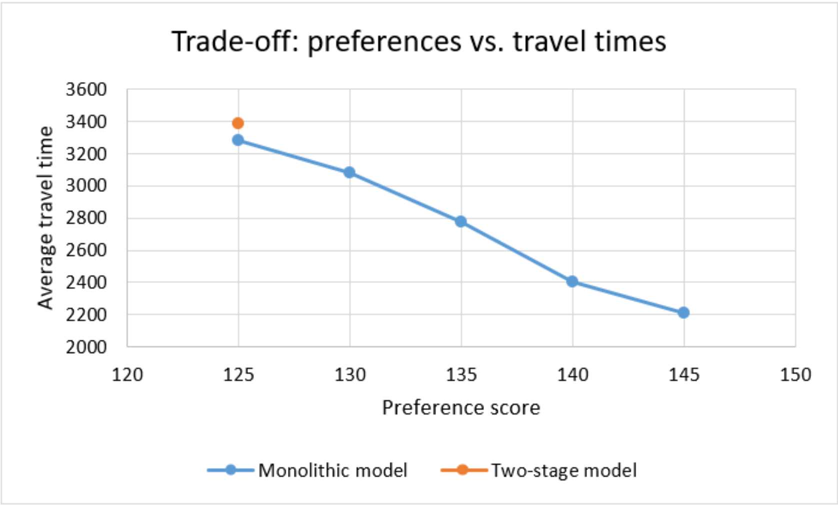


Figure 3.4: Visualisation of the trade-off between the minimisation of the penalty score for the scheduling preferences and the minimisation of the evacuation times.

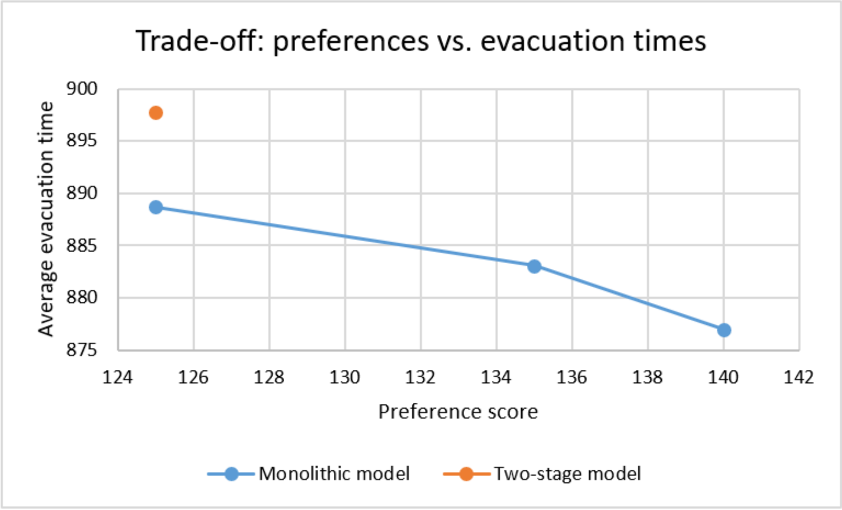
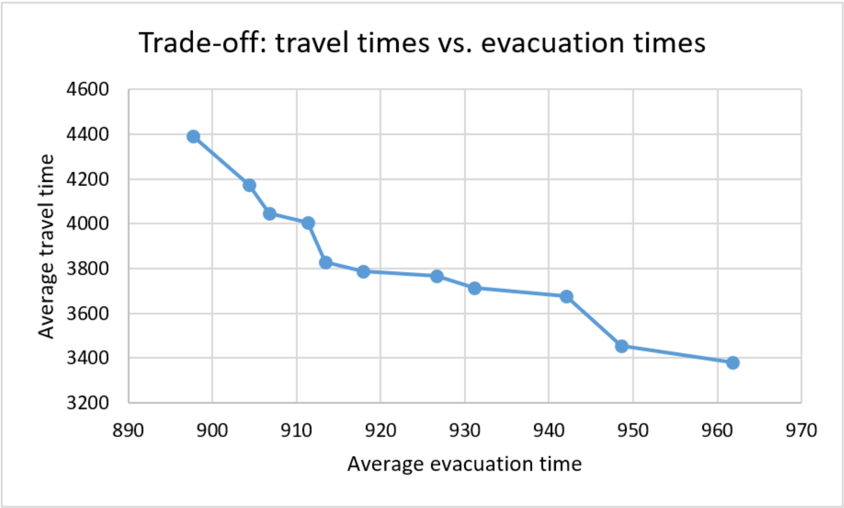


Figure 3.5: Visualisation of the trade-off between the minimisation of the travel times and the minimisation of the evacuation times.



of the evacuation times. The two-stage model again achieves a very good solution with an optimality gap of 1 percent.

For the trade-off between the minimisation of the evacuation times and the minimisation of the travel times between lectures in consecutive time-slots, we use the two-stage model and iteratively solve the second-stage to find the Pareto-optimal frontier. The results are visualised in Figure 3.5. It can be seen that there are also considerable trade-offs between the minimisation of the travel times on the one hand and the evacuation times on the other hand. Again, the trade-off is not convex.

First, we can conclude that there are significant trade-offs between the three types of objectives. Because the trade-offs are not convex, an objective function that optimises a weighted average of the different types of objectives might not be able to find all Pareto-optimal solutions. Second, the two-stage model quickly finds solutions that are close to one endpoint of the Pareto-optimal frontier, but it is unable to explore other solutions on the frontier.

3.9 Heuristic approach

In the previous sections, a MIP formulation has been developed that describes the UCTP at KU Leuven Campus Brussels with the objective of either minimising the travel times between lectures in consecutive timeslots or minimising the evacuation times in the event of an emergency. The mathematical model was solved with a two-phase decomposition approach using a commercial MIP solver and was able to find good solutions in reasonable computation times. However, the two-stage approach is only able to find solutions near one endpoint on the Pareto front. Furthermore, the required computation time increases exponentially for larger problem instances and the model is unable to solve the large KU Leuven FEB instance. Therefore, in this section, we turn to metaheuristic approaches to solve our problem.

Metaheuristics are a popular approach for tackling timetabling problems, because they are easy to implement, can accommodate different types of constraints and objectives and achieve good results (De Causmaecker et al., 2009; Geiger, 2012; Lewis, 2008; Shambour et al., 2013). Examples of metaheuristics often used in timetabling problems are simulated annealing (SA) (e.g., Gunawan and Kien Ming, 2012; Kalender et al., 2012; Zhang et al., 2010) and tabu search (TS) (e.g., Aladag et al., 2009; Hertz, 1991; Lü and Hao, 2010). A metaheuristic starts with one or a set of solutions which are iteratively improved using local search operators. Worsening moves are also accepted from time to time based on some criterion (such as a certain acceptance probability in SA), to enable the algorithm to break free from local optima.

In the next section, we explain the different steps of our algorithm. First, an initial solution is constructed, as is explained in Section 3.9.1. Next, this solution is improved using a metaheuristic with a local search operator. The neighbourhood structure used in our local search is explained in Section 3.9.2. Subsequently, we describe the metaheuristic implemenation we use in combination with our neighbourhood structure in Section 3.9.3. Finally, Section 3.9.4 discusses the computational results.

3.9.1 Initial solution

Since the first stage of the two-stage model is able to find optimal solutions relatively quickly for most problem instances, we simply use this MIP formulation in combination with a commercial MIP solver to obtain a good initial solution for the heuristic.

3.9.2 Neighbourhood structure

The initial solution obtained by the constructive methods is iteratively improved using local search within a metaheuristic framework. During each iteration of the metaheuristic a candidate solution is obtained by changing some assignments of lectures to timeslot-room pairs. We define a *cell* as a timeslot-room pair and we will refer to a change in some assignments of lectures to cells as a *move*. All solutions that can be reached within one move of the current solution constitute the *neighbourhood* of the solution. The neighbourhood structure used in our implementation consists of three types of moves, which are explained below.

An important speed-up technique in local search is incremental cost recalculation (Ross et al., 1994). Since only small changes are made to the solution, it is not necessary to calculate the objective value of this candidate solution from scratch. Instead, it is much more efficient to start from

Figure 3.6: Example of a lecture swap move, where two lectures are swapped.

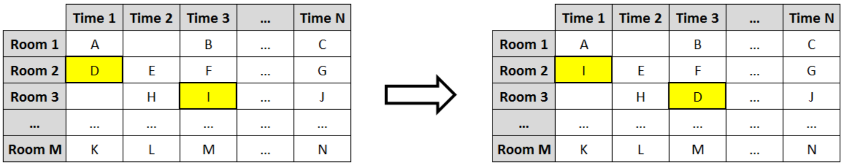
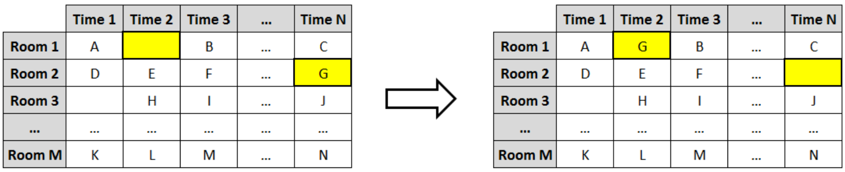


Figure 3.7: Example of a lecture swap move, where a lecture is moved to an empty timeslot and room.



the previous value and add or subtract only the impact the changes have on the objective value. For example, when a room swap is executed, the part of the objective value that depends on the timing of lectures does not change and as such does not need to be calculated again for the candidate solution.

3.9.2.1 Lecture swap

In a lecture swap (LS) move, two timeslots t_1 and t_2 are randomly selected. Then, within each timeslot a classroom is randomly chosen. If both timeslot-classroom combinations contain lectures, then both lectures swap between their timeslots and rooms (Figure 3.6). Alternatively, if only one cell contains a lecture, this lecture is moved from its current cell to the empty cell (Figure 3.7).

Figure 3.8: Example of a room swap move, where two lectures are swapped.

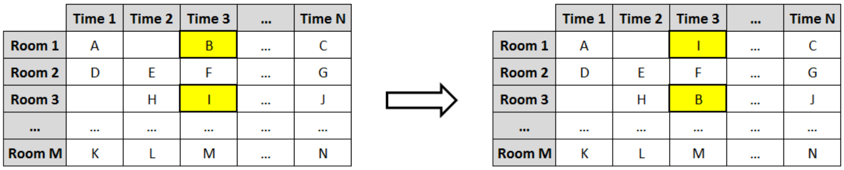
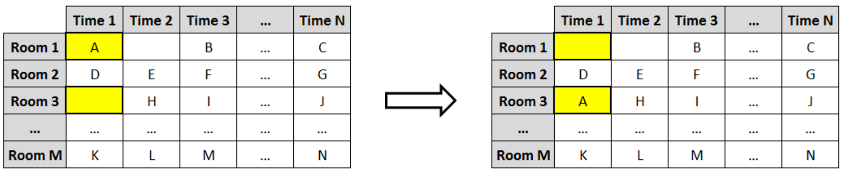


Figure 3.9: Example of a room swap move, where a lecture is moved to an empty room.



3.9.2.2 Room swap

A room swap (RS) move first selects a certain timeslot at random and then randomly chooses two classrooms in this timeslot. Again, either both lectures swap rooms (Figure 3.8), or a single lecture is moved from its current room to an empty room (Figure 3.9).

3.9.2.3 Kempe chain

The previous two moves have a high chance of replacing one set of infeasibilities with a new set of infeasibilities, due to the many dependencies between curricula and teachers in the timetable. Therefore, multiple researchers have included Kempe chain (KC) moves in their neighbourhood search (e.g., Tuga et al., 2007; Fonseca et al., 2016b). Minor differences between implementations exist, ours follows the implementation of Tuga et al. (2007). Two timeslots t_1 and t_2 are randomly selected. Then a bipartite graph is constructed where all lectures planned in timeslot t_1 are denoted by a first group of nodes $\mathcal{N}_1$ and all lectures planned in timeslot t_2 are denoted by a second group of nodes $\mathcal{N}_2$. A node in $\mathcal{N}_1$ is connected by an edge to a node in $\mathcal{N}_2$ if and only if the two corresponding lectures have a conflict (i.e., if they have the same teacher, if they are part of the same curriculum, or if they are currently planned in the same room). Let a chain be a connected subset of nodes from $\mathcal{N}_1 \cup \mathcal{N}_2$. For every chain, we calculate the change in the objective function if all lectures corresponding to the nodes in $\mathcal{N}_1$ on this chain are swapped with their corresponding lectures in $\mathcal{N}_2$. Then the new candidate solution is obtained by swapping all lectures on the chain which improves the objective value most.

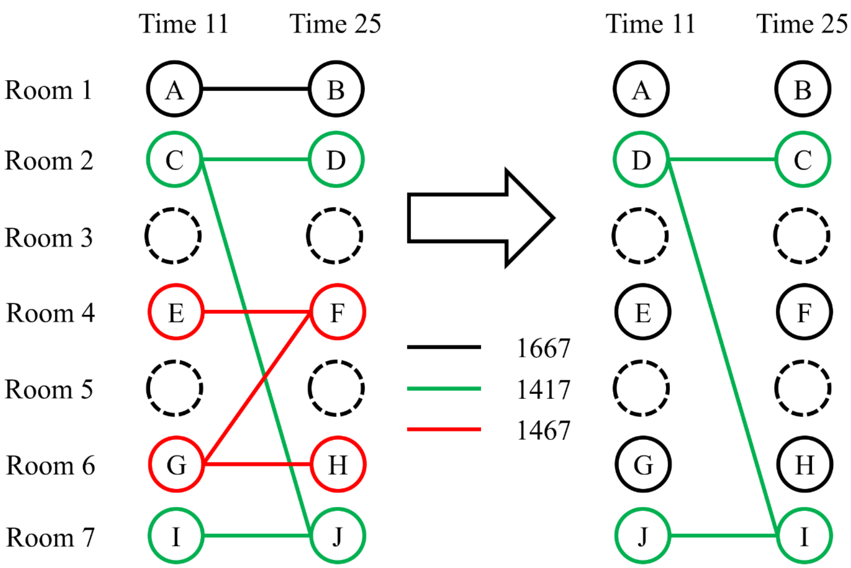
Figure 3.10 illustrates the move. Timeslots 11 and 25 are randomly selected. Then all lectures planned in timeslot 11 are denoted by a node and connected to the nodes in timeslot 25, if and only if they have a conflict. The dotted nodes indicate that no lecture is planned in this room in this timeslot. Subsequently, we calculate the resulting objective value if all nodes from $\mathcal{N}_1$ are swapped with their connected counterpart in $\mathcal{N}_2$. This gives an objective value of 1667 for the blue chain, 1467 for the red chain, and 1417 for the green chain. Consequently, the lectures that are part of the green chain are swapped, since this chain has the lowest objective value.

3.9.3 Parallel simulated annealing

SA was developed by Kirkpatrick et al. (1983), based on an analogy with thermodynamics. Starting from a given initial solution, at each iteration a candidate solution is obtained by performing one of the neighbourhood moves on the current solution. If this candidate solution is better than the current solution, it is always accepted. If the candidate solution is worse, it is accepted with a certain probability, which enables the algorithm to break free from local optima. This acceptance probability is based on

Figure 3.10: Example of a Kempe chain move.

Three chains are identified. The numbers for each chain indicate the objective value of the candidate solution that is obtained by swapping the lectures on the respective chain. Since swapping the lectures of the green chain results in the best candidate solution with a value of 1417, we retain this candidate solution.



the current temperature of the system T and the difference between both objective values Δ, i.e. $\Pr(\text{accept}) = \exp(-\Delta/T)$. After a given number of neighbourhood moves at the same temperature, the temperature is lowered to decrease the probability of accepting worsening moves. After a certain time the temperature will be so low that no worsening moves are accepted and as a result the algorithm will converge to a local optimum.

Nowadays most CPUs have multiple cores, which allows us to increase the efficiency of heuristic methods by means of parallelisation. Parallelisation means that different cores of the CPU execute different calculations simultaneously. This enables us to explore significantly more candidate solutions in the same computation time compared to single core implementations. While the SA heuristic is intrinsically sequential, different methods exist to parallelise it (Lee and Lee, 1996). A first method is to decompose the cost function in a number of disjunct functions. The different functions can then calculate the objective value of a candidate solution in parallel. A second method is to decompose the search space into disjunct sets. Each thread then explores one of the subdomains. A third method is to run multiple SA procedures in parallel. Two implementations are possible (Ferreiro et al., 2013). In the asynchronous implementation, each thread executes one SA procedure. The different threads can start either from the same initial solution or from different initial solutions. Similarly, they can use either the same parameter settings or different parameter settings. Each SA procedure runs independently from the other and when the time limit is reached the best solution found over all threads is returned. In the synchronous method on the other hand, threads run independently for a given number of iterations or time limit. Once all threads have finished, they communicate their best found solutions and the best one is selected. The different threads then each restart their SA procedure from this best found solution. The temperature is controlled by the top level and is reduced each time the threads share the best found solution (Ferreiro et al., 2013).

Since it is not straightforward how the search space or the cost function should be divided into disjunct subsets of more or less the same size, we use the third approach in our implementation. After an initial solution has been obtained as explained in Section 3.9.1, N_{threads} SA procedures are started from this initial solution. During the search process, the different threads communicate N_{sync} times at evenly space intervals. After synchronisation, every thread restarts its SA procedure from the best found solution over all threads.

3.9.4 Results

The heuristic is programmed in C++ and compiled with Microsoft Visual Studio 2017. The callable library of ILOG CPLEX 12.6.3 is used as a MIP solver. The code is executed on a PC with an AMD Ryzen 7 1700X processor of 3.40 GHz and a RAM of 16 GB. The C++ code can again be found at the following website: `https://github.com/HendrikBV/ModelsPhDThesisChapter3`. Since the SA heuristic is stochastic, for every test five independent algorithm runs are executed.

3.9.4.1 Parameter settings

The parameters for our algorithm are based on results from other authors (Fonseca et al., 2016a), as well as our own observations. In each iteration, the lecture swap is executed with a probability of 60 percent, the room swap with a probability of 38 percent and the Kempe chain with a probability of 2 percent. The reason is that the Kempe chain requires considerably more computation time than the other two moves. The temperature updating rule is chosen as $T' = \alpha\, T$. The initial temperature

T_0 is set to 100,000 and $\alpha = 0.9$, since initial tests revealed that the objective function contains many steep valleys which makes it difficult to escape from local optima.

3.9.4.2 Impact of parallelisation

In this section, we compare the impact of parallelisation on the performance of the heuristic. We test three configurations: (i) a standard SA implementation with a single thread ($\mathrm{SA}_{\mathrm{standard}}$), (ii) an asynchronous parallel SA implementation with 12 threads where each thread runs independently and afterwards the best solution is returned ($\mathrm{SA}_{\mathrm{async}}$), and (iii) a synchronous parallel SA implementation with 12 threads where threads exchange best solutions N_{sync} times during the search process in evenly spaced intervals ($\mathrm{SA}_{\mathrm{sync}}$). We use comp07 in the tests as it is a relatively difficult instance. We set the time limit at 500 seconds and $N_{\mathrm{sync}} = 20$.

While $\mathrm{SA}_{\mathrm{async}}$ can perform a total of 165,000 iterations compared to only 20,000 for $\mathrm{SA}_{\mathrm{standard}}$, both implementations achieve nearly the same results, with median objective values of 187,562 and 182,110, respectively. Compared to $\mathrm{SA}_{\mathrm{async}}$, $\mathrm{SA}_{\mathrm{sync}}$ can execute slightly fewer iterations (157,000) because of the communications overhead. However, $\mathrm{SA}_{\mathrm{sync}}$ finds solutions with a median objective value of 158,202 in the same time limit thanks to the sharing of information between threads. We can thus conclude that the synchronous parallel SA performs considerably better than a standard SA implementation or an asynchronous parallel SA implementation.

3.9.4.3 Comparison with two-stage MIP approach

We compare the performance of the heuristic to the performance of the two-stage model on the same set of timetable instances and buildings. We

Table 3.11: The results of the heuristic.

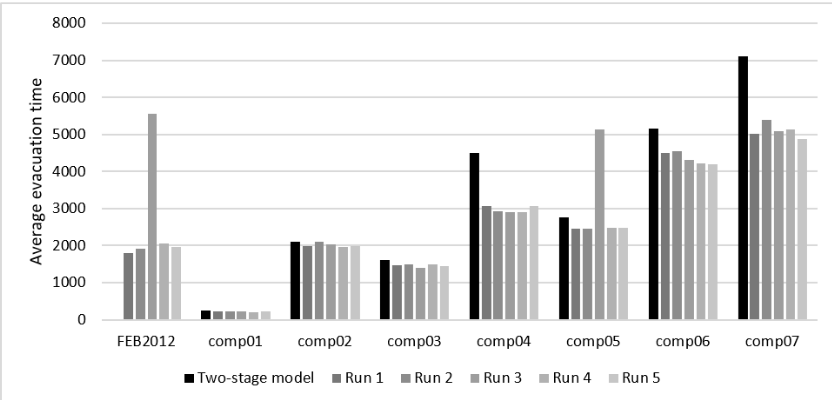
The average evacuation time $\left(\frac{1}{|\mathrm{T}|}\sum_{t\in\mathrm{T}}\mathrm{T}^{\mathrm{evac}}_t\right)$ is minimised. The results of the heuristic are based on five independent runs with different seeds.

Instance	Building	Two-stage model	Heuristic Worst	Median	Best
FEB2012	B-KUL		5550	1970	1795
comp01	B-8-2	243	214	214	211
comp02	B-16-1	2107	2107	1996	1972
comp03	B-16-1	1618	1493	1477	1399
comp04	B-20	4510	3069	2933	2894
comp05	B-16-1	2773	5146	2478	2447
comp06	B-20	5167	4553	4313	4195
comp07	B-20	7103	5398	5084	4878

focus on evacuations only and minimise the average evacuation time over all timeslots, i.e. $\frac{1}{|\mathrm{T}|}\sum_{t\in\mathrm{T}}\mathrm{T}^{\mathrm{evac}}_t$. The stopping criterion for each run of the heuristic is a time limit of 1000 seconds, as in (Fonseca et al., 2016a).

The results are shown in Table 3.11. The results of the two-stage model are also shown for comparison. First, the heuristic is able to improve significantly upon the results obtained by the two-stage model. Apart from two independent runs of the heuristic, all runs for all instances achieve better solutions than the two-stage model. Second, in contrast to the two-stage MIP approach, the heuristic has no difficulties in dealing with the large timetable instance and network model for the KU Leuven FEB instance. Third, in general, the variation between different runs of the heuristic is relatively small, thanks to the synchronous parallel SA implementation. The only exceptions are one run of the FEB2012 instance and one run of the comp05 instance (see Figure 3.11). This again shows that the objective function contains many steep valleys which makes it difficult to escape from local optima.

Figure 3.11: Visualisation of the results of the heuristic compared to the results of the two-stage model.



3.10 Conclusions

In this chapter, the problem of developing compact university course timetables with optimised student flows has been studied. The first part of this chapter has presented a two-stage MIP model for the UCTP with the aim of minimising the travel times of students between lectures in consecutive timeslots. The first stage minimises the violation of the teacher preferences by assigning lectures to timeslots and rooms. The second stage reassigns classrooms to lectures of the timetable of the first stage and minimises the travel times of students who go from their lectures in one timeslot to their lectures in the next timeslot. Student flows in the corridors and at the stairwells are modelled using a graph in which the arcs represent the corridors and stairs in the building. The total travel time of each series of students to go from their first classroom to their next classroom is calculated as the sum of the travel times through each arc on their route, which itself is a function of the total student flow through each arc. Through extensive computational tests we have shown that, in contrast to a monolithic MIP model, our two-stage MIP approach is capable of finding quality solutions with significantly reduced travel times for real-life UCTPs. The approach can find good quality solutions for each of the first set of seven instances of the Curriculum-Based Course Timetabling Track of the ITC2007, proving its applicability to a wide range of real-life problem dimensions. However, the model proved incapable of solving the complex KU Leuven FEB instance.

Next, the evacuation of the university building in the event of an emergency has been studied. We showed that our two-stage MIP formulation could easily be extended to include evacuations. Computational tests on the same set of problem instances show that optimising the assignment of lectures to rooms can also significantly improve on the egress time if the university building were to be evacuated in case of an emergency. We have also showed that there are important trade-offs between the optimisation of the three different objectives, namely the scheduling preferences, the travel times, and the evacuation times.

Finally, in the third part of this chapter we developed a heuristic approach for our problem. The implementation is based on a synchronous parallel simulated annealing heuristic (Ferreiro et al., 2013) and uses three neighbourhood moves, namely a lecture swap, a room swap, and a Kempe chain. Computational tests show that our heuristic implementation is able to find solutions that are considerably better than the solutions obtained by the two-stage MIP approach. The reason is that the two-stage MIP approach can only change the assignment of lectures to rooms to improve the objective of minimising the travel or evacuation times, while the heuristic can also change the assignment of lectures to timeslots. Moreover, while the two-stage MIP approach can only find solutions on or close to one endpoint of the Pareto-optimal frontier, the heuristic can explore the entire frontier. Finally, in contrast to the two-stage MIP approach, the heuristic has no problem dealing with the large KU Leuven FEB instance.

One possible direction for future research is improving the ability of the heuristic to escape local optima. Another possible direction for future research is the implementation of dedicated multi-criteria approaches to solve the problem.

Chapter 4

A surrogate-based tabu search heuristic to optimise the people flows in a timetable

4.1 Introduction

In this chapter, we develop a model that is both a generalisation, as well as a particularisation, of the model of the Chapter 3. It is a generalisation because of two reasons. First, instead of focusing on the curriculum-based university course timetabling problem, we consider a generic timetabling problem that can be applied to many settings, such as university course timetabling, conference timetabling, or timetabling for music and sports events. It is also a particularisation, because we assume that each event has already been assigned to a certain timeslot and we only take the decision of assigning events to rooms into account. However, this limitation confers two advantages. First, it allows us to keep the model generic. Second, it is also useful for implementation in practice, because real-life timetabling problems are already extremely complex as they need to take many different objectives and restrictions into account, which makes the inclusion of additional objectives related to people flows unlikely. Because the assignment of timeslots to events is considered given, other models could be used to construct an initial timetable, after which our model can be used in a second step to optimise the assignments of events to rooms.

In Chapter 3, we have used a network model to describe people flows because of its computational efficiency. However, a downside of these types of models is that they are less realistic and cannot describe distinct groups of pedestrians with different behaviours. Moreover, they cannot readily be used to visualise the flow of pedestrians. In this chapter, we will use a microscopic pedestrian simulation model to model the people flows.

The literature contains many different models that can be used to describe pedestrian walking behaviour (Vermuyten et al., 2016a). However, developing a fully functional crowd simulator that implements one of these pedestrian models is difficult and time consuming. Moreover, without a common framework, comparing different models is not straightforward. Indeed, reimplementations of a certain model by other researchers might differ from the original implementation. The Menge crowd simulation framework[1] (Curtis et al., 2016) addresses these issues. It is an open-source, cross-platform, modular framework that decomposes the problem of crowd simulation into different components or subproblems. It already offers implementations of different models for each subproblem. However, it can also be extended with new functionality through plug-ins. As

[1] `http://gamma.cs.unc.edu/Menge/`

a result, researchers can tailor Menge to their specific problem setting. Moreover, the efficient multi-threaded implementation allows to simulate thousands of agents at interactive rates.

The remainder of this chapter is structured as follows. Section 4.2 defines the problem and Section 4.3 describes the solution approach. Next, Section 4.4 provides an introduction to the Menge crowd simulation framework. A discussion of the different pedestrian models available in Menge is conducted in Section 4.5. Subsequently, Section 4.6 explains how our problem is translated into Menge. In Section 4.7, we discuss our choice of pedestrian model, followed by a computational analysis of the simulator in Section 4.8. Specifically, we look at the distribution of the simulation results of the Menge simulator and the relationship between the number of agents in the simulation and the required computation time. Computational results for two different applications are presented in Section 4.9. The model is also applied to a real-life case study at KU Leuven Campus Brussels in Section 4.10. Finally, Section 4.11 concludes the chapter and lists possible directions for future research.

4.2 Problem statement

Let E be the set of events that need to be planned, T the set of timeslots, and R the set of rooms. We assume fixed timeslots, i.e. all events assigned to a given timeslot start and end at the same time. Each event $e \in E$ is already assigned to a timeslot $t \in T$. Let $E_t^T \subseteq E$ be the subset of events that are planned at time t and let $R_e \subseteq R$ be the subset of rooms that can be assigned to event e. Furthermore, define G as the set of eventgroups. Each eventgroup refers to a series of events that are followed by the same group of people. Events can be part of more than one eventgroup. Let $E_g^G \subseteq E$ be the subset of events that are attended by eventgroup $g \in G$

and let $G_e \subseteq G$ be the subset of eventgroups that attend event $e \in E$. The problem is then to assign every event $e \in E$ to a room $r \in R$. We define the following decision variable

$$x_{er} = \begin{cases} 1 & \text{if event } e \in E \text{ is assigned to room } r \in R \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

The problem can then be formulated as follows.

$$\text{minimise} \quad \mathcal{F} \tag{4.2}$$

subject to:

$$\sum_{r \in R_e} x_{er} = 1 \qquad \forall e \in E \tag{4.3}$$

$$\sum_{e \in E_t^T} x_{er} \leq 1 \qquad \forall t \in T,\ \forall r \in R \tag{4.4}$$

$$x_{er} \in \{0, 1\} \qquad \forall e \in E,\ \forall r \in R \tag{4.5}$$

The objective function (4.2) minimises a general function of the people flows that occur in the timetable as a result of the scheduling decisions taken. Constraint set (4.3) ensures that each event is assigned to a room. Constraint set (4.4) states that at most one event can be assigned to a given room in each timeslot. Finally, constraints (4.5) are the domain constraints of the decision variables.

4.3 Solution approach

No analytic expression for the function $\mathcal{F}$ is available. Instead, simulation will be used to evaluate $\mathcal{F}(\mathbf{x})$ for a given solution $\mathbf{x}$. More specifically, we will use the Menge simulator that is explained in the subsequent sections of this chapter to simulate the people flows. Our problem can thus be classified as a simulation optimisation problem (Ólafsson and Jumi Kim, 2002).

Introductions to the field of simulation optimisation are given by Fu (2001) and Ólafsson and Jumi Kim (2002). More recent reviews are available in Hong and Nelson (2009) and Fu et al. (2005). Jalali and Van Nieuwenhuyse (2015) provide a classification of simulation optimisation models used in inventory replenishment problems.

The solution techniques can be divided into techniques for problems with continuous variables and techniques for problems with discrete variables (Jalali and Van Nieuwenhuyse, 2015). If the decision variables are discrete and the number of feasible solutions is small, then multiple comparisons and ranking and selection procedures are two possible approaches (Goldsman and Nelson, 1998). On the other hand, for problems with discrete variables and a large number of feasible solutions, metaheuristics are a popular approach (Ólafsson and Jumi Kim, 2002).

A relatively new technique in simulation optimisation is surrogate-based or metamodel-based optimisation (Queipo et al., 2005). Because simulations are usually computationally expensive even for small problem instances, the real objective function values of candidate solutions are approximated by a surrogate model that is computationally much less costly. Surrogate models can be built using a variety of statistical techniques depending on the shape of the (real) objective function, such as polynomial regression models, radial basis functions, and kriging (Forrester and Keane, 2009). However, surrogate-based optimisation is usually applied to continuous optimisation problems, due to the nature of the statistical models used (Yin, 2011).

In our approach, we integrate surrogate-based optimisation into a metaheuristic framework. A metaheuristic is used to guide the search and iteratively find improved solutions, because it is not possible to evaluate all feasible solutions due to the size of the solution space. A surrogate model is used during the search to speed up the evaluation of candidate solutions.

While the decision variables in our problem are discrete, it is possible to fit a surrogate model that can predict the value of $\mathcal{F}$ at a given solution $\mathbf{x}$. To this end, we do not use the original decision variables x_{er}, which represent the assignments of events to rooms in the different timeslots. $\mathcal{F}$ does not depend on these assignments as such, but on the number of people going from each origin $r \in R$ to each destination $r \in R$. We can thus fit the surrogate model using any of the available techniques in e.g. Queipo et al. (2005) as a function of the number of people on each route. Since each solution $\mathbf{x}$ of the original decision variables uniquely determines the number of people on each route, every solution can then be evaluated by the surrogate model.

For the metaheuristic, we opt for tabu search. Tabu search was developed by Glover (1986) and is a well-known metaheuristic for combinatorial optimisation problems. In each iteration, tabu search evaluates all possible neighbourhood moves and executes the best one. When no improving move can be found, it chooses the least bad move. To avoid exploring the same solutions over and over again, tabu search stores a tabu list of previous moves which are forbidden. This mechanism allows it to escape from local optima. We choose tabu search because of two reasons. First, it employs best-first search in which all candidate solutions in the current neighbourhood are evaluated before accepting a new solution, in contrast to a depth-first search in which a candidate solution that is better than the current solution is immediately accepted. This allows us to use a preselection strategy (Yin, 2011), in which we first evaluate candidate solutions by the computationally cheap surrogate model and only then re-evaluate one or more promising solutions with the expensive simulator (see below). Second, the tabu list offers a good mechanism to escape local minima, which is independent of the simulated objective values.

Before we explain the neighbourhood move used in the tabu search, we first explain some notation. $\mathcal{F}(\mathbf{x})$ denotes the true objective value of solution $\mathbf{x}$, which is unknown; $\hat{\mathcal{F}}(\mathbf{x})$ refers to the objective value predicted by the surrogate model, $\bar{\mathcal{F}}(\mathbf{x})$ stands for the objective value estimated from n_{rep} replications of the Menge simulator, and $CI^{+}(\mathbf{x})$ and $CI^{-}(\mathbf{x})$ refer to the upper 95 percent and lower 95 percent confidence interval for the real objective value estimated from n_{rep} replications of the Menge simulator.

We use the neighbourhood move shown in Figure 4.1. In each iteration, a random timeslot is selected and then all possible swaps of two events between rooms are evaluated. Only swaps that are feasible and not tabu are considered. All candidate solutions are first evaluated with the surrogate model. Next, the candidate solutions are sorted from most promising (i.e., lowest $\hat{\mathcal{F}}$) to least promising (i.e., highest $\hat{\mathcal{F}}$) and the most promising candidate solution is re-evaluated with the Menge simulator. If $\bar{\mathcal{F}}(\mathbf{x}_{\text{candidate}}) < \bar{\mathcal{F}}(\mathbf{x}_{\text{current}})$, we accept the candidate solution and put it in the tabu list. Otherwise, we go to the next candidate solution. If after ξ candidate solutions no improving solution has been found, we resort the first ξ candidate solutions based on $\bar{\mathcal{F}}$ and take the best one. This strategy ensures that the limited computational budget is not wasted on evaluating many candidate solutions when the current solution is of poor quality but instead is used to explore the area around a potential local minimum more thoroughly. Moreover, limiting the maximum number of candidate solutions that are evaluated to ξ can further improve this strategy, since we can be relatively certain that the current solution is a local minimum because $\hat{\mathcal{F}}$ is worse for the remaining candidate solutions.

During the search, the candidate with the best average evacuation time as evaluated by the Menge simulator is saved. If the search ends, this

Figure 4.1: The neighbourhood move used in the surrogate-based tabu search.

a)

	room 1	room 2	room 3	room 4
timeslot 1	Event 1		Event 4	
timeslot 2		Event 3	Event 6	
timeslot 3	Event 2	Event 5		Event 7

	room 1	room 2	room 3	room 4
timeslot 1	Event 1		Event 4	
timeslot 2		Event 3	Event 6	
timeslot 3	Event 5	Event 2		Event 7

b)

	room 1	room 2	room 3	room 4
timeslot 1	Event 1		Event 4	
timeslot 2		Event 3	Event 6	
timeslot 3	Event 2	Event 5		Event 7

	room 1	room 2	room 3	room 4
timeslot 1	Event 4		Event 1	
timeslot 2		Event 3	Event 6	
timeslot 3	Event 2		Event 5	Event 7

solutions is returned. On the flip side, no aspiration criterion is used during the tabu search, as initial experiments showed that this can lead to cycling between the same solutions.

4.4 The Menge crowd simulation framework

4.4.1 Problem decomposition

As is common in the crowd simulation literature (Curtis, 2014), Menge divides the problem of describing pedestrian walking behaviour into four subproblems, namely goal selection, plan computation, plan adaptation, and motion synthesis.

4.4.1.1 Goal selection

The goal selection subproblem consists of determining what each pedestrian wants to achieve. This subproblem spans the largest time horizon. Typically goals are set at the start of the simulation, although in more complex scenarios they can be updated during the simulation.

Menge uses a *behavioural finite state machine* (BFSM) to represent agent behaviours. A *goal* is the location that an agent wants to reach. Goals can be grouped into *goalsets*. Goalsets are useful when agents need to select one goal out of a specific set of goals, such as the nearest goal, the farthest goal, or a random goal.

At each time step, each agent in the simulation exists in a certain *state*. A state has a *goal selector* which determines the current goal of the agent, and a *velocity component* which determines how the agents tries to reach the target (see Section 4.4.1.2). States can also have *actions* associated with them. Actions are executed when an agent enters a state and allow the modelling of complex behaviours.

Transitions determine how the current state of an agent is updated to a new state. Each transition has a *condition* that determines when the transition is executed.

Different populations of agents can be described by different *agent profiles*. An agent profile lists the behavioural characteristics of a group of pedestrians, such as the preferred free walking speed and the maximum acceptable speed. An *agent group* describes a number of agents that share the same agent profile and that have the same starting conditions in the simulation. The agent group describes the agent profile for the agents, their initial positions, and their initial state.

4.4.1.2 Plan computation

In this second subproblem, for each agent a static plan is computed to achieve the goal that was selected in the previous subproblem. This plan results in an instantaneous preferred velocity. Only static obstacles are taken into account. The movement of other pedestrians is not taken into account in this step, which is done during plan adaptation. The time horizon of this subproblem is the medium term.

Currently, Menge offers three types of approaches for computing paths, namely road maps, navigation meshes, and guidance fields. These approaches are referred to as the *velocity component*. Road maps are graphs in which each vertex represents a separate area in the environment and the edges connect adjacent areas that are not separated by obstacles. For every possible location, there needs to be a vertex that is visible from that location (i.e., unobstructed by obstacles) so that agents can find a feasible route. Navigation meshes extend road maps, since each vertex is replaced with a convex polygon that represents an area that can be freely traversed by agents. Agents can use a straight line to travel to their target within the current polygon since the polygon is convex and free of obstacles. To travel between polygons agents use the edges in the same way as in the road map. Finally, a guidance field divides the environment into different areas and associates a potential field with each area. The potential field acts as a force that directs the pedestrian towards his or her goals and away from obstacles. It is possible to extend Menge with other wayfinding algorithms.

4.4.1.3 Plan adaptation

The third subproblem takes the preferred velocity as input from the previous subproblem and transforms it into a feasible velocity by taking dynamic obstacles and agents into account. This subproblem has the shortest time horizon. Most pedestrian models in the literature are concerned with this subproblem.

Menge uses a plug-in architecture that allows different pedestrian models to be used in conjunction with the simulator. Menge already includes the implementation of six different models, namely the Optimal Reciprocal Collision Avoidance (ORCA) model (van den Berg et al., 2011), the Pedestrian Velocity Obstacles (PedVO) model (Curtis, 2014), the Social-Force Model of Helbing et al. (2000), the Generalized Centrifugal Force

Model (Chraibi et al., 2010), the predictive collision avoidance model for pedestrian simulation of Karamouzas et al. (2009), and the social-force model with explicit collision prediction of Zanlungo et al. (2011). The first two models belong to the class of 'velocity obstacle models', while the other four models belong to the class of 'social-force models'. We give an overview of these two classes of models in Section 4.5. Researchers can also extend Menge with new pedestrian models.

4.4.1.4 Motion synthesis

This optional stage translates the movement computed by the previous stages into physical character motion for visual applications.

4.4.2 Mathematical formulation

In this section, we discuss how Menge translates the different conceptual problems in describing human walking behaviour into a mathematical problem. In the subsequent sections, we use *velocity* to refer to a two-dimensional vector and *speed* to refer to a scalar or the norm of a velocity vector. Suppose there are m agents. The positions and instantaneous velocities of agents in two-dimensional Euclidean space as a function time, respectively, are given by

$$r : \mathbb{R} \to \mathbb{R}^{m\times 2} : t \mapsto r(t) = \begin{pmatrix} r_{1,x}(t) & r_{1,y}(t) \\ r_{2,x}(t) & r_{2,y}(t) \\ \vdots & \vdots \\ r_{m,x}(t) & r_{m,y}(t) \end{pmatrix}, \tag{4.6}$$

$$v : \mathbb{R} \to \mathbb{R}^{m\times 2} : t \mapsto v(t) = \begin{pmatrix} v_{1,x}(t) & v_{1,y}(t) \\ v_{2,x}(t) & v_{2,y}(t) \\ \vdots & \vdots \\ v_{m,x}(t) & v_{m,y}(t) \end{pmatrix}. \tag{4.7}$$

The problem of computing the trajectories of agents is then given by the following initial value problem

$$\begin{cases} \frac{\mathrm{d}r(t)}{\mathrm{d}t} = v(t) = \mathcal{V}(S(t)) \\ r(0) = r_0, \end{cases} \tag{4.8}$$

where $S : \mathbb{R} \to \mathbb{S} : t \mapsto S(t)$ gives the simulation state at time t, $\mathcal{V} : \mathbb{S} \to \mathbb{R}^{m\times 2} : S \mapsto \mathcal{V}(S)$ is a function that determines each agent's instantaneous velocity as a function of the simulation state, and $r_0 \in \mathbb{R}^{m\times 2}$ represents the initial positions of agents. The positions of agents over time can then be determined by solving for $r(t)$.

Each of the subproblems in Section 4.4.1 can be translated to a mathematical function. The goal selection problem can be described by the function $G : \mathbb{S} \to \mathbb{R}^{m\times 2}$ that maps the simulation state into two-dimensional goal positions for each agent. The plan computation problem can be represented by the function $P : \mathbb{S} \times \mathbb{R}^{m\times 2} \to \mathbb{R}^{m\times 2}$ that maps the simulation state and each agent's position into an instantaneous preferred velocity. Finally, the plan adaptation problem can be expressed by the function $A : \Sigma \subseteq \mathbb{S} \times \mathbb{R}^{m\times 2} \to \mathbb{R}^{m\times 2}$ that maps the local simulation state and position of each agent into a feasible velocity for each agent. The instantaneous preferred velocity can then be computed by the composition of these functions:

$$v(t) = \mathcal{V}(S(t)) = A\left(P\left(G\left(S\left(t\right)\right)\right)\right). \tag{4.9}$$

4.5 Overview of pedestrian models in Menge

4.5.1 Social-force models

The social force model was developed by Helbing (1991) and Helbing and Molnár (1995). The movement of a pedestrian is the result of various forces, based on an analogy with physics: an attractive force in direction of the pedestrian's destination, repulsive forces forces from other pedestrians or obstacles such as walls, and possibly attractive forces from other pedestrians (e.g., friends) or objects (e.g., window displays). However, the latter are usually not included as they are difficult to specify in many scenarios.

The movement model is described by the following differential equations

$$\begin{cases} \frac{\mathrm{d}r(t)}{\mathrm{d}t} = v(t) = g(w(t)) \\ \frac{\mathrm{d}w(t)}{\mathrm{d}t} = F^{\text{total}}(r(t), v(t)), \end{cases} \tag{4.10}$$

where $r : \mathbb{R} \to \mathbb{R}^{m\times 2}$ gives the positions the m pedestrians as a function of time, $v : \mathbb{R} \to \mathbb{R}^{m\times 2}$ and $w : \mathbb{R} \to \mathbb{R}^{m\times 2}$ give, respectively, the *actual* and *desired* velocities of the m pedestrians as a function of time, $g : \mathbb{R}^{m\times 2} \to \mathbb{R}^{m\times 2}$ gives the relation between the desired velocity and the actual velocity and $F^{\text{total}} : \mathbb{R}^{m\times 4} \to \mathbb{R}^{m\times 2} : (r(t), v(t)) \mapsto F^{\text{total}}(r(t), v(t))$ gives the total forces acting on the pedestrians based on their current locations and actual velocities. The desired velocities are determined by the forces acting on the pedestrians. Because these forces can grow very large (e.g., when two pedestrians are going to collide), the resulting desired velocities can grow larger than physically possible. Therefore, the actual velocities of pedestrians are limited by their maximal acceptable speed $v^{\max} \in \mathbb{R}^{m\times 1}$.

The main difference between the different social force models in the literature lies in how they define the various forces. We first give an overview of the foundational model of Helbing and Molnár (1995) and then discuss some of the extensions that are implemented in Menge.

4.5.1.1 Model of Helbing and Molnár (1995)

Force towards target The force towards the target depends on the difference between a pedestrian's desired velocity $v_i^0 \in \mathbb{R}$ and his or her actual velocity. The change in a pedestrian's walking velocity happens with a certain relaxation time $\tau \in \mathbb{R}$. Let $d_i \in \mathbb{R}^2$ be the destination of pedestrian i. The force towards the target is then given by

$$F_i^{\text{target}} = \frac{1}{\tau}\left(v_i^0 e_i(t) - v_i(t)\right). \tag{4.11}$$

The term $e_i(t) = \frac{d_i - r_i(t)}{\|d_i - r_i(t)\|}$ represents the desired direction of motion of pedestrian i.

Force from other pedestrians The force experienced by pedestrian i from other pedestrians j is given by

$$\begin{aligned} F_i^{\text{ped}} &= \sum_{j \neq i} F_{i,j}^{\text{ped}} \\ &= \sum_{j \neq i} -V_0^{\text{ped}} w_{ij}(t) \frac{r_i(t) - r_j(t)}{\|r_i(t) - r_j(t)\|} \exp\left(\frac{-b_{ij}(t)}{\sigma}\right), \end{aligned} \tag{4.12}$$

where

$$b_{ij}(t) = 0.5\sqrt{\left(\|r_{ij}(t)\| + \|r_{ij}(t) - v_j(t)\Delta t e_j(t)\|\right)^2 - (v_j(t)\Delta t)^2}, \tag{4.13}$$

with $r_{ij}(t) = \|r_i(t) - r_j(t)\|$ and

$$w_{ij}(t) = \begin{cases} 1 & \text{if } \frac{\langle v_i(t), r_j(t) - r_i(t)\rangle}{\|v_i(t)\|\|r_j(t) - r_i(t)\|} \geq \cos(\phi) \\ c & \text{otherwise,} \end{cases} \tag{4.14}$$

with $\sigma \in \mathbb{R}$, $\phi \in [0, 2\pi]$, and $0 < c < 1$ constants.

The factor $\frac{r_i(t)-r_j(t)}{\|r_i(t)-r_j(t)\|}$ represents the direction of the force, V_0^{ped} is a scaling factor, the factor $b_{ij}(t)$ takes into account both the distance between pedestrians as well as their direction of motion (i.e., pedestrians require more space in front of them than orthogonal to them), and $w_{ij}(t)$ takes the perception of pedestrians into account (i.e., situations behind a pedestrian will have less influence than situations right in front of him or her), with ϕ the angle of sight of a pedestrian.

Force from obstacles The force experienced by pedestrians from obstacles k is given by:

$$\begin{aligned} F_i^{\mathrm{obs}} &= \sum_k F_{i,k}^{\mathrm{obs}} \\ &= \sum_k -V_0^{\mathrm{obs}} w_{ik}(t) \frac{r_i(t)-r_k}{\|r_i(t)-r_k\|} \exp\left(-\frac{\|r_i(t)-r_k\|}{\sigma}\right), \end{aligned} \tag{4.15}$$

with

$$w_{ik}(t) = \begin{cases} 1 & \text{if } \frac{\langle v_i(t), r_k - r_i(t)\rangle}{\|v_i(t)\|\|r_k - r_i(t)\|} \geq \cos(\phi) \\ c & \text{otherwise.} \end{cases} \tag{4.16}$$

The factor $\frac{r_i(t)-r_k}{\|r_i(t)-r_k\|}$ represents the direction of the force, V_0^{obs} is a scaling factor, $\|r_i(t) - r_k\|$ gives the distance between the pedestrian and the obstacle, and w_{ik} again takes the perception of pedestrians into account.

Total force The total force acting on pedestrian i is equal to the sum of the individual forces acting on that pedestrian, i.e.

$$F_i^{\mathrm{total}} = F_i^{\mathrm{target}} + F_i^{\mathrm{ped}} + F_i^{\mathrm{obs}}. \tag{4.17}$$

Relationship between actual and desired velocity The speed-cutoff function g is given by

$$g(w(t)) = \begin{cases} w(t) & \text{if } \|w(t)\| \leq v^{\max} \\ \frac{v^{\max}}{\|w(t)\|} w(t) & \text{otherwise.} \end{cases} \tag{4.18}$$

The function g assures that the actual velocity of pedestrians never exceeds their maximal acceptable velocity $v^{\max}$.

4.5.1.2 Extensions

Helbing et al. (2000) focus on evacuations and adapt the forces exerted by other pedestrians and obstacles to include additional body compression and sliding friction forces caused by high densities in overcrowded situations. Let $\rho_{ij} = \rho_i + \rho_j$ be the sum of the radii of pedestrians i and j, let $d_{ij}(t) = \|r_i(t) - r_j(t)\|$ denote the distance between the centres of mass of pedestrians i and j, let $\nu_{ij}(t) = \frac{r_i(t) - r_j(t)}{d_{ij}(t)}$ be the normalised vector pointing from pedestrian j to pedestrian i, let $\theta_{ij}(t)$ be the tangential direction (i.e., perpendicular to $\nu_{ij}(t)$), and let $\Delta v^{\theta}_{ji}(t) = (v_j(t) - v_i(t))\theta_{ij}(t)$ be the tangential velocity difference. Then the force function between pedestrians i and j is given by:

$$\begin{aligned} F_i^{\text{ped}} &= \sum_{j \neq i} F_{i,j}^{\text{ped}} \\ &= \sum_{j \neq i} \left(V_0^{\text{ped}} \exp\left(\frac{\rho_{ij} - d_{ij}(t)}{\sigma} \right) + k\mathcal{G}\left(\rho_{ij} - d_{ij}(t)\right) \right) \nu_{ij}(t) \\ &+ \kappa\mathcal{G}(\rho_{ij} - d_{ij}(t))\Delta v^{\theta}_{ji}(t)\theta_{ij}(t), \end{aligned} \tag{4.19}$$

where the function $\mathcal{G}(x)$ is zero if the pedestrians do not touch each other (i.e. if $d_{ij}(t) > r_{ij}(t)$) and is otherwise equal to the argument x.

The first term in Eq. (4.19) is the regular force between pedestrian i and j in the absence of friction. (Note that in contrast to Helbing and

Molnár (1995), Helbing et al. (2000) do not include the perception term and assume circular instead of elliptical equipotential lines.) The second term is a body force counteracting body compression and the third term is a sliding friction force impeding relative tangential motion.

The force between pedestrians and walls is analogous:

$$
\begin{aligned}
F_i^{\text{obs}} &= \sum_k F_{i,k}^{\text{obs}} \\
&= \sum_k \left(V_0^{\text{obs}} \exp\left(\frac{\rho_i - d_{ik}(t)}{\sigma}\right) + k\mathcal{G}(\rho_i - d_{ik}(t)) \right) \nu_{ik}(t) \\
&+ \kappa\mathcal{G}(\rho_i - d_{ik}(t))(v_i(t)\theta_{ik}(t))\theta_{ik}(t).
\end{aligned} \tag{4.20}
$$

Chraibi et al. (2010) developed the Generalised Centrifugal Force (GCF) model, where agents are represented by ellipses that change size based on the agents' velocity instead of fixed circles. The reason is that pedestrians who walk faster take larger steps and thus require more space in front of them. Secondly, they use the inverse of the relative distance between two pedestrians in the expression for the force between them, instead of an exponentially decreasing function of the distance. However, they limit the magnitude below a certain threshold, to avoid the force going to infinity as the distance goes to zero.

In the previous models, the repulsive forces between agents are calculated based on their current positions and velocities. By contrast, Karamouzas et al. (2009) and Zanlungo et al. (2011) take future expected interactions into account when calculating these forces.

Finally, most model simply add all forces acting on a pedestrian to arrive at the resulting force acting on this pedestrian. It is possible, however, that certain forces neutralise each other. For this reason, Karamouzas et al. (2009) sequentially take the impact of the various forces into account.

They first adjust the current velocities based on the force that has the most immediate expected interaction and then recompute the remaining forces. They repeat this process until all forces have been taken into account. However, there is no proof that this approach avoids that forces neutralise each other.

4.5.2 Velocity obstacle models

In velocity obstacle models, agents predict future positions of obstacles and other pedestrians and adjust their velocities to avoid potential collisions. However, if only pairwise interactions are taken into account, predictions of future positions and velocities might be wrong, and collisions might still occur. The ORCA model (van den Berg et al., 2011) solves these problems by calculating the optimal (changes in) preferred velocities for all agents simultaneously.

The ORCA model was developed for robots (Curtis, 2014). As a result, some of the behaviours of agents in the ORCA model are not realistic representations of human behaviours. A first element is that in the ORCA model inter-agent relationships are assumed to be perfectly reciprocal. This means that to avoid a collision, both agents will evenly divide the burden of changing their preferred velocities. This is realistic for robots, as the energy cost of adjusting their behaviour is then shared equally between all robots. However, for pedestrian interactions, it is not really realistic. The PedVO model of Curtis (2014) proposes two techniques to include asymmetries, namely proxy agents and right of way. Proxy agents are agents that are attached to real agents to allow the expression of different behaviours. For example, the authority proxy can be used to model the behaviour of a line of policemen who control a large crowd. The concept of right of way is borrowed from vehicular traffic and allows certain agents to have priority over other agents. Agents with lower priority face an increasing burden to avoid collision.

Secondly, methods for plan computation usually represent immediate goals as points. This leads to unrealistic behaviours where agents who travel towards the same destination all try to reach the same intermediate point, even if enough space is available in the vicinity to simultaneously travel towards their final destination without interference. Therefore, the PedVO model uses so-called wayportals, which use line segments instead of points to represent immediate goals of agents. Instead of a single preferred velocity vector, agents then have a set of velocity vectors, from which the best alternative is to be determined. As a result, wayportals improve the flow of agents by allowing them to use all available space instead of artificially hindering each other when trying to move along a certain line through an area.

Finally, agents in the ORCA model do not adhere to the fundamental diagram that states that walking speed decreases as crowd density increases (Curtis, 2014). Since the ORCA model is only concerned with collision avoidance, nothing prohibits agents from moving in dense crowds with arbitrarily large speeds. This can lead to unrealistically low evacuation times in scenarios with dense crowds. The PedVO model includes this relationship by explicitly linking the preferred free walking speed of agents to the crowd density in their surroundings.

4.6 Problem translation in Menge

4.6.1 Building representation

Menge simulates crowd dynamics in the two-dimensional plane. However, buildings consisting of multiple floors are three dimensional. We can represent such buildings in Menge by projecting the different floors next to each other in the two-dimensional plane. We will represent stairs by 'teleporting' agents who enter the stairwell on the origin floor to the destination floor.

The walls in the building translate to an obstacles in Menge. Menge implements obstacles as closed polygons. Each point of the polygon is represented by a coordinate in two dimensional Euclidean space. Connected walls (e.g., all walls of the same room) are represented by a single obstacle, allowing Menge's reasoning to include looking around corners.

4.6.2 Road map

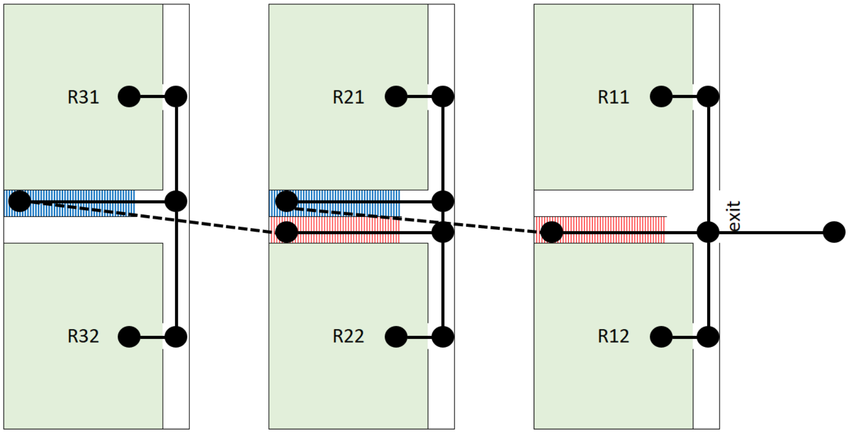
We use road maps to allow Menge to compute feasible paths from an agent's current position to his or her destination. We opt for the road maps since they can be used in combination with our building representation to represent paths between the different floors of the building. Moreover, they are easier to implement than navigation meshes and are sufficient to describe all possible routes in our building representation.

We place one vertex at the inside of the room for each door of a room of the building, one vertex in the hallway for each door of each room, one vertex at the 'entrance' of each stairwell, one vertex at the 'exit' of each stairwell, and one outside of each building exit. Vertices on a given floor of the building are connected by edges if there are no obstacles that stand between them. Moreover, every vertex at the 'entrance' of a stairwell is connected to the vertex at the 'exit' of the stairwell on the floor below or above, depending on whether the stairwell goes up or down.

Figure 4.2 represents an example of a building with six rooms divided over three floors and the corresponding road map.

Figure 4.2: Example building and corresponding road map.

The building consists of six rooms (R11, R12, R21, R22, R31, and R32) divided over three floors. The read stripes indicate the stairwell that goes upstairs and the blue stripes indicate the stairwell that goes downstairs. The road map is represented by the black vertices and edges.



4.6.3 Goals and goalsets

For each room in the building, there is a corresponding goal. All these goals are grouped in a first goalset. For every exit in the building, there is also a corresponding goal, which are grouped in a second goalset.

4.6.4 States and transitions

There are three types of states, as is summarised in Table 4.1. First, there is a final state named ‘Stop’ that indicates that an agents has reached his or her goal. If all agents have reached this final state, the simulation is finished and the elapsed time is recorded. Second, there is a state for every destination to indicate that an agent wants to reach that destination. For each room in the building, the state has a goal selector that selects the corresponding goal and a velocity component that specifies how that target can be reached. The velocity component we use is the same for every state and is a road map that represents the feasible paths in the building, as explained in Section 4.6.2. For exits, the goal selector selects the nearest goal from all goals in the goalset corresponding to the exits. At the start of the simulation, each agent starts in one of these states depending on his or her destination. Third, when a pedestrian reaches a stairwell, (s)he enters a special state corresponding to that stairwell. This state does not have a goal selector or velocity component, since the goal does not change. Instead, this state executes an action, whereby the agent is teleported to the corresponding stairwell of the floor below or above depending on his or her destination. Because pedestrian models work in a two-dimensional plane, this allows us to represent three-dimensional buildings with multiple floors by projecting the different floors next to each other in the same plane.

Table 4.1: States used in our scenario implementation in Menge.

State	final	goal selector	velocity component	action
Stop	yes	-	-	-
WalkToRoom...	no	specific room	road map	-
WalkToExit	no	nearest exit	road map	-
Stairs...	no	-	-	teleport

Table 4.2: Transitions used in our scenario implementation in Menge.

From state	condition	to state
WalkToRoom... / WalkToExit	goal reached	Stop
WalkToRoom... / WalkToExit	area reached	Stairs...
Stairs...	auto	WalkToRoom... / WalkToExit

There are three types of transitions between states, as is shown in Table 4.2. First, there is a transition between the initial state of a pedestrian to the final state. The condition type is ‘goal reached’, i.e. as soon as the agent reaches his or her goal, the transition is executed. Second, there is a transition from an agents current state to the special stairs state when the agent reaches a stairwell. The transition is executed if the agent reaches a certain area. Finally, the third transition changes a pedestrian’s state from the special stairs state back to his or her previous state so that (s)he continues moving towards his or her goal. This transition is automatically executed immediately after the agent has been teleported.

4.6.5 Agent profile and agent groups

We assume that all people in the simulation share the same characteristics. Therefore, we generate a single agent profile to represent the characteristics of pedestrians. Following Helbing and Molnár (1995), the preferred free walking speed of each pedestrian is drawn from a normal distribution with mean equal to 1.34 m/s and standard deviation equal to 0.26 m/s and the maximum free walking speed is set equal to 1.74 m/s.

To simulate the evacuation in timeslot t, we generate one agent group for every event that is planned in that timeslot. The number of agents in the agent group is set equal to the number of people in the corresponding event. At the start of the simulation, the agents in this agent group are randomly positioned in the corresponding room in the building and they start in the 'WalkToExit' state.

To simulate the flows from events in timeslot t to events in timeslot $t+1$, we generate one agent group for every eventgroup that has at least one event in both timeslots. The number of agents in the agent group is set equal to the number of people in the corresponding eventgroup. If the eventgroup has an event at time t and at time $t+1$, the agents in this agent group are randomly positioned in their origin room in the building and they start in the 'WalkToRoom...' state corresponding to their destination room. If the eventgroup only has an event at time t, the agents in this agent group are randomly positioned in their origin room in the building and they start in the 'WalkToExit' state. Finally, if the eventgroup only has an event at time $t+1$, the agents in this agent group are randomly positioned outside of the building and they start in the 'WalkToRoom...' state corresponding to their destination room.

4.6.6 Interface

Currently, the Menge library does not offer the possibility to programmatically construct simulations through the exposure of C++ classes. Instead, instantiation of a simulation has to be done through an XML-interface. Menge requires three XML-files, namely a behaviour file, a scene file, and a view file. The behaviour file lists all goals, states, and transitions. The scene file specifies the parameters of the pedestrian models, the agent profiles, the agents groups, and the obstacles such as walls. Finally, the view file details how the simulation should be visualised. For every simulation scenario, we programmatically generate the required XML-files and then call the Menge library with these files as arguments.

4.7 Choice of pedestrian model

Curtis (2014) enumerates three properties of an ideal crowd simulator. First, the simulator should be computationally efficient. This efficiency depends on the *cost* of computing a single time step and the *stability* of the pedestrian model. The stability of a model indicates the maximum time step that can be taken so that the accuracy of the results is not compromised. Some models allow large time steps, while others require small time steps to produce reliable and consistent behaviour. Second, the simulator should be robust. This means that the parameters of the model should not be tailored to each specific simulation scenario, but instead should only depend on the pedestrian population (e.g., the age of the pedestrians). Third, the simulator should accurately represent actual human walking behaviour.

Social-force models score poorly on the first two criteria (Curtis, 2014). They have a poor stability because of two reasons. First, they are second-order models. The forces determine the accelerations of pedestrians, which have to be integrated twice to obtain the positions of pedestrians. Second, the repulsive forces are stiff when pedestrians are near each other or obstacles, which means that small changes in relative positions have a large impact on the size of the forces. As a result, social-force models require a small time step to produce accurate results. Furthermore, the force parameters often need to be tuned to the simulation scenario (Chraibi et al., 2010).

Velocity Obstacles models on the other hand avoid these problems (Curtis, 2014). They have a high stability and thus can be use a large time step, increasing their efficiency. The reason is that they are first order models. The velocities are only a function of the simulator state and are thus independent of the time step. For these reasons, we use the PedVO model in our simulation runs.

4.8 Analysis of the Menge simulator

In this section, we analyse the distribution of the egress times and travel times calculated by Menge. We also take a closer look at the computational efficiency of Menge relative to the number of people in the simulation.

4.8.1 Test sets

We test Menge on two different building configurations. The first building B-8-2 consists of eight rooms and is shown in Figure B.3. The second building B-16-1 consists of sixteen rooms and is shown in Figure B.5.

For evacuations, we then generate three test instances for evacuations per building as follows. The number of people in each room at the start of the simulation is drawn from a uniform distribution. We use three levels for the number of people in each room, namely $U(0, 10)$, $U(10, 20)$, and $U(20, 40)$. We will refer to these instances as *Evac-8-L*, *Evac-8-M*, and *Evac-8-H*, for building B-8-2 and a low, medium, and high number of people, respectively, and *Evac-16-L*, *Evac-16-M*, and *Evac-16-H* for building B-16-1 and a low, medium, and high number of people, respectively.

For flows between consecutive events, we also generate three test instances per building as follows. We randomly select $R - 1$ room pairs (i.e. one

Table 4.3: Results of the Menge simulations for the test instances.

The results are based on 100 simulations runs.

Instance	# People	Comp. time (s)	Mean	Stddev	C.O.V.	Skewness
Evac-8-L	53	7.3	89.84	13.96	0.16	1.39
Evac-8-M	112	24.6	126.71	7.99	0.06	1.21
Evac-8-H	240	91.1	225.82	4.62	0.02	1.19
Evac-16-L	86	11.3	107.71	16.19	0.15	1.03
Evac-16-M	234	57.6	150.69	9.51	0.06	0.89
Evac-16-H	526	221.1	240.84	4.29	0.02	0.49
Travels-8-L	56	7.6	80.30	11.06	0.14	1.43
Travels-8-M	135	22.3	137.68	17.75	0.13	3.64
Travels-8-H	252	68.5	216.90	37.92	0.17	1.91
Travels-16-L	95	13.0	116.64	19.25	0.17	0.99
Travels-16-M	267	73.9	219.05	27.79	0.13	1.56
Travels-16-H	462	229.5	343.73	86.34	0.25	4.10

origin and one destination room per pair), 1 incoming flow from outside of the building to a random room, and 1 flow starting from a random room and going to the exit. The number of people on each of the chosen paths is drawn from a uniform distribution. We again use the following three levels for the number of people per path, namely $U(0,10)$, $U(10,20)$, and $U(20,40)$. We will refer to these instances as *Travels-8-L*, *Travels-8-M*, and *Travels-8-H*, for building B-8-2 and a low, medium, and high number of people, respectively, and *Travels-16-L*, *Travels-16-M*, and *Travels-16-H* for building B-16-1 and a low, medium, and high number of people, respectively.

4.8.2 Results

We execute 100 simulations for every test instance. The mean, standard deviation, coefficient of variation, and the skewness of the evacuation times for each instance are reported in Table 4.3.

It is clear that an increase in the number of people in the simulation leads to an increase in the computation time of the simulation. For the *Evac-8-L* and the *Travels-8-L* instance the computation time per simulation equals 0.07 and 0.08 seconds, respectively, while for the *Evac-16-H* and the *Travels-16-H* instance the computation time increases to 2.21 and 2.30 seconds per simulation, respectively.

As the number of people in the building increases, the mean evacuation time increases as would be expected. However, the standard deviation decreases. A possible explanation is that in more congested scenarios, individual differences in behaviour between pedestrians matter less, leading to less variation in the output. Moreover, the distribution of the results of the evacuation times is skewed to the right for all instances.

Analogously, as the number of people in the building increases, the mean travel time between events in consecutive timeslots also increases as would be expected. In contrast to evacuations, for the travel time between events in consecutive timeslots the standard deviation increases with the mean and the coefficient of variation is larger on average. This is because now people flow in opposite directions and hinder each other much more. The skewness of the distributions are also noticeably larger for flows between consecutive timeslots than for evacuations.

4.9 Computational results

In this section, we show how the surrogate-assisted tabu search algorithm can be used to solve different types of problems related to crowd management, where the schedule of events has an impact on the resulting people flows. Therefore, in Section 4.9.1 and Section 4.9.2, we illustrate the algorithm on two possible applications, where the function $\mathcal{F}$ has a different form. First, however, we analyse the efficiency of the Menge simulator and the distribution of the simulation results in Section 4.8.

All tests are executed on a PC with an AMD Ryzen 7 1700X @ 3.40 GHz processor and 16 GB RAM under the Windows 10 operating system. The model is coded in Qt Creator 4.5.0 with Qt 5.10. We use the open source Dlib machine learning library for C++ developed by King (2009) for the surrogate models. The datasets, the C++/Qt code for the algorithm, as well as the instance generator can be found at the following website: `https://github.com/HendrikBV/A-surrogate-based-tabu-search-heuristic-to-optimise-people-flows-in-a-timetable`.

4.9.1 Application 1: evacuation problem

A first type of problem where the optimisation of people flows is important are evacuation problems. At mass crowd events like music festivals, cultural events, or sport events, the schedule of music bands or sport games has a large impact on the resulting spectator flows during an emergency evacuation.

If a building (sports stadium, festival area, etc.) were evacuated during timeslot $t \in T$, then the assignments of the events to rooms (locations) in that timeslot impact the number of people present in each room of the building (zone in the sports stadium or festival area) and thus the resulting evacuation time. Because there are multiple timeslots and a different set of events that are planned in each timeslot, the evacuation time between each timeslot differs. We propose as objective function

$$\mathcal{F} = \sum_{t \in T} ET_t, \tag{4.21}$$

where ET_t is the evacuation time in timeslot t. The evacuation time is defined as the time when the last person exits the building.

We can use incremental cost recalculation (Ross et al., 1994) when calculating the real objective value of a candidate solution with Menge during the tabu search. Because our neighbourhood structure only changes assignments in a single timeslot t and all ET_t are independent, we only need to simulate the evacuation time in timeslot t instead of for all timeslots.

4.9.1.1 Test instances

Four different building layouts are used to analyse the people flows during an evacuation. Building B-8-1 (Figure B.1) consists of 8 rooms divided over 2 floors. Building B-8-2 (Figure B.3) also consists of 8 rooms, but divided over 4 floors. Both buildings only have a single stairwell. Building B-16-1 (Figure B.5) consists of 16 rooms divided over 4 floors and 2 stairwells. Finally, building B-16-2 (Figure B.7) also consists of 16 rooms divided over 4 floors, but with 3 stairwells.

We developed an instance generator to generate ten random timetable instances and corresponding start solutions. Five instances have 8 rooms and five instances have 16 rooms. Each timetable instance consists of 5 timeslots. First, for each timeslot the number of events is drawn from a uniform distribution between 3 and the number of rooms in the building. The number of eventgroups is set equal to the number of rooms. In a second step, the events in each timeslot are randomly assigned to an eventgroup. Third, feasiblity for each event-room combination is determined by a draw from a Bernoulli distribution with $p = 0.7$. Finally, the number of people for each eventgroup is drawn from a uniform distribution between 5 and 25 people. For each event, the number of people is then set equal to sum of people of all eventgroups to which the event belongs. In the start solution, each event is randomly assigned to a feasible room.

4.9.1.2 Choice and quality of the surrogate model

We compare two types of machine learning techniques, namely kernel ridge regression (KRR) and support vector regression (SVR), each time in combination with one of four kernel types, namely a radial basis kernel (RBK), a histogram intersection kernel (HIK), a linear kernel (LK), and a quadratic kernel (QK).

During an evacuation each person walks to one of the exits of the building. While there can be multiple exits in the building, the Menge simulator assumes that each person travels to the nearest exit. All people in a given room thus follow the same path when evacuating the building. As a result, the input for the surrogate models is simply the number of people in each room.

We randomly generate 500 configurations (i.e., a random number of people in each room) and simulate the evacuation time with the Menge simulator. We set the maximum time limit to train the surrogate models at 1800 seconds. Let $\mathbf{z}_i = (z_{i1}, ..., z_{i|R|})$ be the number of people in each room and y_i the evacuation time in each observation $i \in I$, respectively. This same set of 500 observations is used to train each of the surrogate models. We use the built-in optimiser of the Dlib libary to automatically find the best regularisation parameter for KRR and SVR as well as the best parameters for each of the kernels.

Let $\hat{y}_i$ be the evacuation time predicted by the surrogate model for observation $i \in I$, let $\bar{y} = \sum_{i \in I} y_i$, and let $\bar{\hat{y}} = \sum_{i \in I} \hat{y}_i$. We then test the quality of each of the surrogate models based on this same set of observations by computing the mean squared error (MSE)

$$MSE = \frac{1}{|I|} \sum_{i \in I} (y_i - \hat{y}_i)^2 , \tag{4.22}$$

the Pearson correlation coefficient (PCC)

$$PCC = \frac{\sum_{i\in I}(y_i - \bar{y})\left(\hat{y}_i - \bar{\hat{y}}\right)}{\sqrt{\sum_{i\in I}(y_i - \bar{y})^2}\sqrt{\sum_{i\in I}\left(\hat{y}_i - \bar{\hat{y}}\right)^2}}, \tag{4.23}$$

and the mean average error (MAE)

$$MAE = \frac{1}{|I|}\sum_{i\in I}|y_i - \hat{y}_i|, \tag{4.24}$$

using 10-fold cross validation. While MAE gives the average prediction error, MSE gives more weight to large deviations between predicted and simulated values.

We do this analysis once with building B-8-1 of Figure B.1 and once with building B-16-1 of Figure B.5 in the simulations. The results are listed in Table 4.4. Some of the combinations of machine learning technique and kernel type lead to a prohibitively large computation time needed to train the surrogate model. The KRR trainer with LK is the fastest, followed by the KRR trainer with QK. The KRR trainer with QK has the best MSE on both tests and the best PCC on the second test. It thus provides a good trade-off between required computation time and accuracy and therefore we use it in all subsequent tests.

We also test the performance of the surrogate model during the tabu search. For this purpose, we calculate the MSE, PCC, and MAE between the predicted and simulated values of all candidate solutions within a given iteration. The simulated objective value for each solution is the average of 100 replications of the Menge simulator. We do this analysis for the 1st, 10th and 100th iteration of for 4 datasets as shown in Table 4.5.

Table 4.4: The performance of the different surrogate models for the evacuation times.

MLT refers to the machine learning technique used. SDE is the standard deviation of the MAE over the different cross validations.

Building	MLT	Kernel	Time (s)	MSE	PCC	MAE	SDE
B-8-1	KRR	RBK	246	81.1	0.814	6.6	6.2
		HIK	366	99.3	0.787	7.5	6.6
		LK	4	81.4	0.814	6.6	6.1
		QK	59	79.4	0.819	6.6	6.0
	SVR	RBK	838	80.2	0.821	6.3	6.3
		HIK	>1800				
		LK	>1800				
		QK	>1800				
B-16-1	KRR	RBK	316	217.6	0.553	11.3	9.5
		HIK	1052	238.1	0.563	12.0	9.7
		LK	11	217.6	0.556	11.3	9.5
		QK	179	214.1	0.564	11.2	9.4
	SVR	RBK	312	220.7	0.560	10.9	10.1
		HIK	>1800				
		LK	>1800				
		QK	>1800				

Table 4.5: The performance of the KRR QK surrogate model during the tabu search for the evacuation times.

Building	Dataset	Iteration	MSE	PCC	MAE
B-8-1	T-8-1	1	54.4	0.840	6.8
		10	39.6	0.957	6.0
		100	48.9	0.928	6.6
B-8-2	T-8-2	1	12.3	0.936	3.2
		10	86.3	0.922	8.1
		100	15.1	0.924	3.4
B-16-1	T-16-1	1	62.6	0.667	7.4
		10	462.7	0.816	20.5
		100	66.6	0.857	7.8
B-16-2	T-16-2	1	29.7	0.932	4.6
		10	10.0	0.949	2.6
		100	21.6	0.862	4.1

The surrogate model performs well for all four instances and different phases of the tabu search. The MAE varies between less than 3 seconds and around 8 seconds, except for one outlier of 20.51 seconds. Moreover, the PCC lies between 0.67 and 0.96, with an average of 0.88. This means that the surrogate model succeeds in distinguishing between good and bad solutions.

4.9.1.3 Parameter setting tabu search

The tabu search implementation itself has two parameters, namely the length of the tabu list, ℓ, and the maximum number of solutions that are re-evaluated with Menge in a single iteration, ξ. We test three values for ℓ and three values for ξ, once on building B-8-1 in combination with instance T-8-1 and once on building B-16-1 in combination with instance T-16-1. We set the computational budget for the tabu search equal to 10,000 simulations. Also, we set $n_{\text{rep}} = 10$ (Law and Kelton, 1991).

Table 4.6: The performance of the tabu search for different parameter settings.

Building	Dataset	ℓ	ξ	$\bar{\mathcal{F}}\left(\mathbf{x}^{\text{start}}\right)$	$\bar{\mathcal{F}}\left(\mathbf{x}^{\text{best}}\right)$	Pct. improvement
B-8-1	T-8-1	10	1	514.4	460.8	10.41
			10	512.5	451.4	11.93
			∞	515.5	448.9	12.93
		40	1	515.4	461.9	10.38
			10	515.7	452.4	12.29
			∞	516.0	452.2	12.36
		70	1	515.7	461.8	10.45
			10	518.2	451.6	12.85
			∞	518.3	455.9	12.05
B-16-1	T-16-1	10	1	618.1	489.7	20.77
			10	620.9	477.7	23.06
			∞	624.1	484.3	22.40
		40	1	621.4	490.2	21.13
			10	618.6	485.8	21.48
			∞	619.2	490.1	20.84
		70	1	617.5	496.3	19.62
			10	621.8	482.8	22.35
			∞	618.8	479.8	22.47

The results are shown in Table 4.6. To test which of the parameter settings achieves the best performance, we first execute an ANOVA test with the percentage improvement as dependent variable and the problem instance, ℓ, and ξ as independent variables. We include all main and interaction effects and set the acceptance level at 0.05. Only the main effects of ξ and the problem instance are significantly different from zero. In a second step, we fit a regression model on these two variables to see how much of the variance in performance they explain. The p-value of the regression model is 9.84e-14 and the adjusted R^2 equals 0.986. The estimated coefficients are listed in Table 4.7. $\xi = 10$ achieves the best results, but they are not

Table 4.7: The results of the regression analysis.

The value of ξ is treated as a categorical variable, since for the case of $\xi = \infty$ no specific value exists based on which we can calculate a linear regression coefficient for ξ.

Factor	Estimate	Standard error	t-statistic	p-value
(Intercept)	10.54	0.2911	36.22	0.0000
$\xi = 10$	1.86	0.3565	5.23	0.0001
$\xi = \infty$	1.72	0.3565	4.81	0.0003
Instance = T-16-1	9.83	0.2911	33.78	0.0000

significantly different from $xi = \infty$. ℓ does not have a significant impact on the results. Therefore, we simply choose $\xi = 10$ and $\ell = 40$ in all subsequent tests.

4.9.1.4 Results

We again use the parameter settings of Section 4.9.1.3. For every instance 500 observations are generated to train the surrogate models. The results are shown in Table 4.8.

The algorithm takes between 1200 and 1800 seconds for the small instances and between 2500 and 4200 seconds for the larger instances. This difference is due to the fact that a single evaluation with the Menge simulator takes longer for simulations with more people. The time to generate observations and to train the surrogate model is only a small fraction of the time spent on the tabu search. It is clear that the algorithm succeeds in finding solutions that are a significant improvement compared to the random start solution. Moreover, the results show that not only the timetable instance, but also the building layout have a significant impact on the evacuation time and analogously on the possible improvement in this evacuation time.

Table 4.8: The results of the surrogate-based tabu search for the evacuation problem.

TGO refers to the time required to generate the 500 observations; TTM refers to the time required to train the metamodels, TTS refers to the time spent on the tabu search. All times are in seconds. Iter. TS refers to the number of iterations performed during the tabu search. Finally, Pct. Improv. refers to the percentage improvement between the mean value of the start solution and the mean value of the best solution.

Building	Dataset	TGO	TTM	TTS	Total time (s)	Iter. TS	Start solution			Best solution			Pct. Improv.
							CI^-	mean	CI^+	CI^-	mean	CI^+	
B-8-1	T-8-1	71	55	1576	1702	328	511	514	518	451	455	459	11.5
B-8-1	T-8-2	71	50	1374	1495	347	504	507	510	427	433	440	14.6
B-8-1	T-8-3	66	37	1082	1185	340	451	454	457	386	392	398	13.7
B-8-1	T-8-4	80	50	1040	1170	318	431	436	440	381	386	392	11.3
B-8-1	T-8-5	66	52	1369	1487	351	483	486	489	434	440	445	9.6
B-8-2	T-8-1	80	51	1674	1805	341	615	619	624	471	478	485	22.8
B-8-2	T-8-2	80	63	1512	1655	328	544	549	544	446	456	466	17.0
B-8-2	T-8-3	75	46	1145	1266	319	546	550	554	394	402	411	26.8
B-8-2	T-8-4	90	51	1086	1227	338	576	581	587	389	401	412	31.1
B-8-2	T-8-5	74	47	1478	1599	353	583	587	591	460	466	473	20.6
B-16-1	T-16-1	217	291	2994	3502	294	615	621	627	491	504	516	18.9
B-16-1	T-16-2	203	471	3544	4218	339	664	670	676	498	508	517	24.2
B-16-1	T-16-3	213	468	3192	3873	303	619	625	632	483	493	504	21.1
B-16-1	T-16-4	204	268	2242	2714	305	593	599	606	411	423	435	29.4
B-16-1	T-16-5	203	373	2817	3393	290	611	617	623	453	462	471	25.2
B-16-2	T-16-1	195	365	2663	3223	281	526	532	537	431	443	456	16.6
B-16-2	T-16-2	184	365	3247	3796	298	625	630	635	456	466	477	25.9
B-16-2	T-16-3	193	371	2878	3442	289	534	539	544	401	412	422	23.6
B-16-2	T-16-4	185	216	2088	2489	292	545	551	557	370	381	393	30.8
B-16-2	T-16-5	184	410	2509	3103	317	540	545	550	374	383	392	29.7
B-16-2	T-16-5	184	410	2509	3103	317	540	545	550	374	383	392	29.7

4.9.1.5 Validation of results

In Section 4.9.1.4 it was shown that the tabu search succeeds in finding solutions with significantly improved evacuation times compared to a random start solution. However, it is not clear how good the solutions actually are compared to the unknown optimal solution. Therefore, in this section we use exhaustive search to validate the performance of the heuristic.

We generate five small instances for which exhaustive search is feasible and use building B-8-2 in all tests. All instances consist of a single timeslot. We run an exhaustive search and save all generated solutions to plot the entire objective function of the solution space. We then run the algorithm and use a replication budget of only 1000 replications, because the instances are considerably smaller than the instances in Section 4.9.1.4.

The results are listed in Table 4.9. The table lists the objective value of the worst solution, the third quartile, the median, the first quartile, and the best solution. The half width of the confidence intervals for the objective values is also listed, since all values are based on ten replications of the Menge simulator. In two of the five instances the heuristic finds the optimal solution, while for the other instances the optimality gaps are less than 1 percent, 3.5 percent, and 7.1 percent respectively. These results show that our heuristic is indeed able to find high quality solutions.

4.9.2 Application 2: flows between events in consecutive timeslots

A second type of problem is when we want to minimise the people flows between events in consecutive timeslots. We have already studied this

Table 4.9: Validation of the heuristic results with exhaustive search.

	ES1	ES2	ES3	ES4	ES5
Number of events	4	5	6	7	8
Worst	143	163	150	171	163
Q3	122	138	131	157	142
Median	115	127	124	150	135
Q1	105	119	114	143	129
Best	79	97	92	136	116
Heuristic	79	97	99	141	117
Half width CI	2	2	2	2	2

type of problem in Chapter 3 for university course timetabling. In academic institutions where the class rooms are concentrated in one or a few buildings, congestion problems may occur in the halls, at the stairwells and elevators at time of course changes. These congestion problems are caused by travelling students that all have to switch rooms at the same moment. When the class rooms are spread over a city, the impact of the resulting student flows on traffic congestions can be studied. In both cases these student flows can be controlled through the course timetable. Another example can be conference scheduling, where people attend different presentations throughout the day. The assignment of talks to rooms has an impact on the people flows between consecutive talks.

In this case, there is a flow between every pair of timeslots in the timetable, when people travel from events in timeslot t to events in timeslot $t+1$. We propose as objective function

$$\mathcal{F} = \sum_{t \in T \setminus \{|T|\}} TT_{t,t+1}, \tag{4.25}$$

where $TT_{t,t+1}$ is the travel time between timeslot t and timeslot $t+1$. The travel time is defined as the time when the last person arrives at his or her destination.

We can again use incremental cost recalculation (Ross et al., 1994) when calculating the real objective value of a candidate solution with Menge during the tabu search. For this problem type, however, $TT_{t,t+1}$ depends on assignments in timeslot t as well as timeslot $t+1$. When the neighbourhood move changes assignments in timeslot t, we thus need to simulate both $TT_{t-1,t}$ and $TT_{t,t+1}$.

We use the same timetable instances and buildings as in Section 4.9.1.1 for all tests.

4.9.2.1 Choice and quality of the surrogate model

We compare the same machine learning techniques as in Section 4.9.1.2.

In this case, every group of people has a different destination. Every individual travels from the location of the event (s)he attends at timeslot t to the location of the event (s)he attends at timeslot $t+1$. If a person does not attend an event at time t, but does attend an event at timeslot $t+1$, (s)he enters the building from outside and travels to the location of the event. On the other hand, if a person attends an event at timeslot t, but no event at timeslot $t+1$, (s)he exits the building. Since the Menge simulator assumes people take the shortest route between two locations, there are thus $(|R|+1)\,|R|$ possible paths people can travel on. However, this number quickly grows very large, even for small problem instances. Therefore, for the larger buildings we group nearby rooms into a single node to reduce the number of paths. For every instance, we use 8 nodes corresponding to the rooms. Moreover, we do not distinguish between paths going from A to B and paths going from B to A. As a result, the input for the surrogate models is the number of people on each path.

We again randomly generate 500 configurations (i.e., a random number of people on each path) and simulate the evacuation time with the Menge simulator. We use the same quality measures as in Section 4.9.1.2 to com-

Table 4.10: The performance of the different surrogate models for the travel times between events in consecutive timeslots.

MLT refers to the machine learning technique used. SDE is the standard deviation of the MAE over the different cross validations.

Building	MLT	Kernel	Time (s)	MSE	PCC	MAE	SDE
B-8-1	KRR	RBK	454	2892.6	0.761	38.1	38.0
		HIK	693	3195.3	0.733	38.1	40.9
		LK	42	3251.0	0.727	40.5	40.2
		QK	489	2894.2	0.774	38.3	37.8
	SVR	RBK	142	2708.1	0.782	34.7	38.8
		HIK	96	3314.8	0.726	39.2	42.2
		LK	>1800				
		QK	>1800				
B-16-1	KRR	RBK	561	934.7	0.640	24.3	18.6
		HIK	698	973.4	0.646	24.5	19.3
		LK	43	957.3	0.632	24.3	19.1
		QK	505	939.4	0.641	24.5	18.4
	SVR	RBK	29	913.4	0.652	23.8	18.7
		HIK	37	881.7	0.665	23.5	18.2
		LK	>1800				
		QK	>1800				

pare the performance of the different machine learning techniques, namely MSE, PCC, and MAE in combination with 10-fold cross validation.

We do this analysis once with building B-8-1 of Figure B.1 and once with building B-16-1 of Figure B.5 in the simulations. The results are listed in Table 4.10. The required computation time for SVR in combination with LK or QK again exceeds 1800 seconds. SVR in combination with RBK achieves the best performance for B-8-1 on all three criteria and the second best performance for B-16-1. Moreover, the required computation time is relatively small. Therefore, we use it in all subsequent tests.

Table 4.11: The performance of the SVR RBK surrogate model during the tabu search for the travel times between events in consecutive timeslots.

Building	Dataset	Iteration	MSE	PCC	MAE
		1	611.2	0.569	23.6
B-8-1	T-8-1	10	3036.4	0.704	49.2
		100	1915.6	0.185	40.1
		1	2086.7	0.665	28.3
B-8-2	T-8-2	10	771.1	0.636	26.1
		100	50.3	0.952	5.1
		1	6926.2	0.320	83.0
B-16-1	T-16-1	10	14,344.4	0.018	119.5
		100	41,111.1	-0.118	202.2
		1	1637.1	0.296	39.1
B-16-2	T-16-2	10	14,804.2	0.164	121.0
		100	6758.3	0.366	81.9

We again test the performance of the surrogate model during the tabu search. Table 4.11 shows the results. In contrast to the evacuation problem of Section 4.9.1, the predictions of the surrogate models during the search are considerably poorer. However, for the two small datasets, the PCC is still reasonably high, meaning the surrogate model succeeds in distinguishing between good and bad solutions. By contrast, for the two large datasets, the PCC is significantly lower, and particularly bad for the 10th and 100th iteration of building B-16-1 with instance T-16-1.

4.9.2.2 Results

We use the same parameter settings as in Section 4.9.1.3. For every instance 500 observations are generated to train the surrogate models. The results are shown in Table 4.12.

For this problem, the algorithm takes between 2000 and 2600 seconds for the small instances and between 5400 and 9600 seconds for the larger instances. The required computation times are larger than for the evacuation time, because for most of the neighbourhood moves two sets of simulations (for $TT_{t-1,t}$ and $TT_{t,t+1}$) need to be executed instead of just one. While the quality measures for the surrogate model shown in Table 4.11 seem to be quite poor, the algorithm nevertheless succeeds in finding solutions with significantly lower total travel times compared to the random start solution. Moreover, the number of iterations of the tabu search lies between 263 and 353, which corresponds to the results for the evacuation problem of Table 4.8. This means that the surrogate model does in fact succeed in distinguishing good solutions from bad solutions and thus considerably reduce the need for expensive simulations. Also, the average percentage improvement for this problem type is larger than for the evacuation problem. This can be explained by the fact that planning consecutive events attended by the same group of people in the same room cancels that flow of people. By contrast, for the evacuation problem it is impossible to cancel flows as everyone still has to exit the building.

Table 4.12: The results of the surrogate-based tabu search for the problem where the travel times between events in consecutive timeslots are minimised.

TGO refers to the time required to generate the 500 observations; TTM refers to the time required to train the metamodels, TTS refers to the time spent on the tabu search. All times are in seconds. Iter. TS refers to the number of iterations performed during the tabu search. Finally, Pct. Improv. refers to the percentage improvement between the mean value of the start solution and the mean value of the best solution.

Building	Dataset	TGO	TTM	TTS	Total time (s)	Iter. TS	Start solution			Best solution			Pct. Improv.
							CI^-	mean	CI^+	CI^-	mean	CI^+	
B-8-1	T-8-1	181	117	2191	2489	329	446	453	459	291	303	315	33.0
B-8-1	T-8-2	174	89	1759	2022	329	431	436	441	307	315	323	27.7
B-8-1	T-8-3	157	76	1777	2010	309	406	411	416	308	317	326	22.8
B-8-1	T-8-4	236	79	1979	2294	285	376	381	385	316	325	334	14.6
B-8-1	T-8-5	157	256	1794	2207	312	418	423	429	273	282	292	33.3
B-8-2	T-8-1	169	75	2394	2638	335	563	569	575	311	319	327	44.0
B-8-2	T-8-2	168	113	1917	2198	327	431	437	443	327	337	347	22.9
B-8-2	T-8-3	149	80	1871	2100	353	540	546	551	290	300	311	44.9
B-8-2	T-8-4	207	67	1972	2246	328	574	580	586	328	335	342	42.2
B-8-2	T-8-5	146	134	2107	2387	341	459	469	464	308	317	325	32.5
B-16-1	T-16-1	93	28	6175	6296	263	655	664	674	527	544	561	18.1
B-16-1	T-16-2	98	29	9437	9564	284	771	781	791	587	609	630	22.1
B-16-1	T-16-3	103	32	7923	8058	277	702	710	719	544	564	584	20.6
B-16-1	T-16-4	92	29	5752	5873	288	640	650	661	481	498	515	23.4
B-16-1	T-16-5	89	28	6345	6462	288	766	776	785	522	540	559	30.3
B-16-2	T-16-1	83	28	5331	5442	275	602	611	620	439	460	481	24.7
B-16-2	T-16-2	86	26	7911	8023	277	745	760	774	501	526	550	30.8
B-16-2	T-16-3	89	27	7131	7247	291	696	707	718	496	509	521	28.1
B-16-2	T-16-4	90	28	5443	5561	266	595	605	616	450	465	480	23.2
B-16-2	T-16-5	77	30	6046	6153	286	665	676	686	516	526	537	22.1

4.10 Case study at KU Leuven Campus Brussels

In this section, we present a real-life application of the surrogate-based tabu search heuristic. In Chapter 3, we introduced the timetabling problem at KU Leuven Campus Brussels. Hundreds of students attend lectures in a single building and congestion problems occurs in the corridors and at the stairwells during lecture transitions. We will show that the surrogate-based tabu search of Chapter 4 can handle the large problem instance at KU Leuven Campus Brussels without any problem.

Section 4.10.1 discusses the data requirements to apply the heuristic. In, Section 4.10.2 the implementation of the model in a scheduling tool with a graphical user interface (GUI) is presented. Section 4.10.3 presents the results of the algorithm and compares the quality of the generated schedules with the quality of the current schedule proposed by the planning department.

4.10.1 Required data

We again consider the university course timetable at KU Leuven Campus Brussels, as was introduced previously in Chapter 3. However, in Chapter 3, we only considered the Faculty of Economics and Business (FEB). Here, we take the lectures of all faculties (The FEB, The Faculty of Law, and The Faculty of Arts) into account. Morevoer, in Chapter 3, the data requirements were more complex as the lectures needed to be assigned to timeslots and rooms, while in this chapter we only focus on the room assignments and consider the assignment of lectures to timeslots chosen by the Planning Department as fixed. Finally, the data in Chapter 3 were gathered by Mercy (2012) for the academic year 2011-2012. Now, we focus on the second semester of academic year 2018-2019.

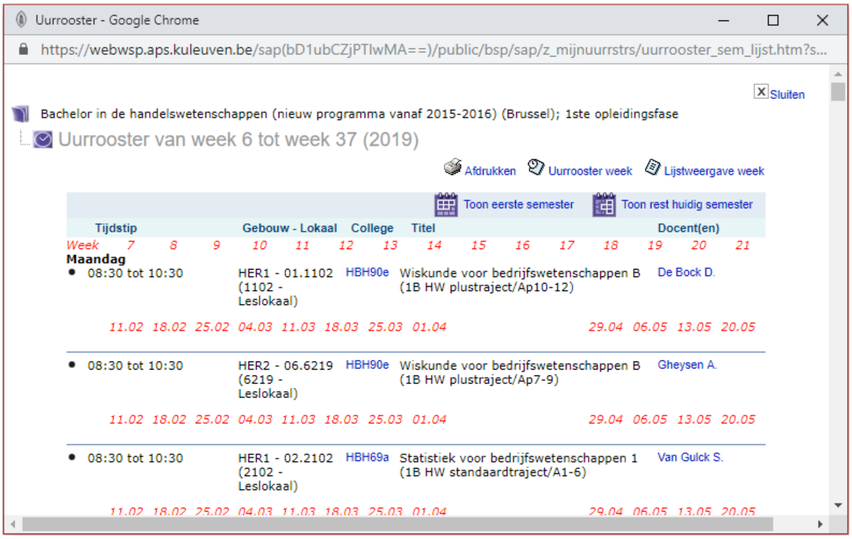
As explained previously, lectures are planned from Monday till Friday and every day consists of six timeslots. Different types of rooms are available for the various lectures, namely regular classrooms, PC rooms, and specific conversation and interpreter rooms for language classes. The timetable is built around a basic schedule that is repeated every week of the semester. However, there can be minor differences between weeks, as some classes only meet every two weeks. In this case study, we focus on the third week of the semester because the first and second week are often a bit different from the rest of the semester.

The surrogate-based tabu search requires as input the set of lectures ('events') that are planned in each timeslot, $E_t^T \subseteq E$, the set of rooms in which each lecture can be planned, $R_e \subseteq R$, and the set of curricula ('eventgroups') that attend a given lecture, G_e. Furthermore, we need to know the number of students in each curriculum π_g^G, and the number of people in each event $\pi_e^E = \sum_{g \in G_e} \pi_g^G$. Finally, the tabu search requires an initial solution to start from. This means we need to provide an initial assignment of lectures to rooms, x_{er}.

The course timetable is developed by the Planning Department of KU Leuven Campus Brussels a few weeks before the start of the academic year. The current version is available on the university website: `https://onderwijsaanbod.kuleuven.be/opleidingen/n/`. Figure 4.3 shows as an example the timetable for the first year of the Bachelor of Business Administration. For every lecture, one can see the day, the start and end time, and the classroom in which the lecture is planned. The teacher and the unique course code can also be seen.

We use an online tool that uses the source code of the website containing the KU Leuven timetable to build a CSV-file containing the required information on all lectures, i.e. the name, the date, the start and end time,

Figure 4.3: The timetable for the first year of the Dutch Bachelor of Business Administration.



and the room (Verraedt, 2018). With the help of the Planning Department, we also compile a file that contains information on the curricula (i.e., the curriculum name, the number of subgroups, and the number of students), the available rooms (i.e., the room name, the type of room, and the capacity), and room requirements for the different lectures.

A C++ program was written that synthesises this information and builds a problem dataset in the format required by the algorithm. However, it is not always easy to translate the complex real-world timetable to our idealised problem setting. Therefore, some choices need to be made, which are explained below.

First, in our problem setting, each timeslot has a fixed duration of two hours. However, in the real timetable, a few lectures have a duration of only one hour and a few others have a duration of four hours. In the first case, assume that the lecture either fills the entire timeslot or we combine two one-hour lectures into a single lecture. In the second case, we split the lecture into two separate lectures.

Second, curricula are sometimes divided into different subgroups. This can happen for three different reasons. In the Bachelor of Applied Linguistics, for example, students choose two main languages from English, French, German, Spanish, Italian, and Polish. Each combination of languages attends a different set of lectures. A second example is when students can choose different majors or minors. Students following a different major attend a different set of lectures. Lastly, in some programmes the number of students is simply too large for every student to attend the same lecture in the same group. Therefore, the students are split into subgroups that attend different versions of the same lecture at different times or in different rooms. The Bachelor of Business Administration is one example. In our dataset, we generate a different eventgroup for each subset of students.

Finally, the algorithm needs information on which lectures can be assigned to which rooms. Of course, lectures cannot be assigned to rooms with insufficient capacity. Next, the type of room needs to be taken into account. Regular lectures should be planned in regular rooms, PC classes should be planned in PC rooms, language classes should be planned in specific language rooms. Initially we only took these two constraints into account. However, when we showed our initial solution to the Planning Department, they listed several scheduling preferences that should be taken into account as much as possible. Some teachers want a specific set of rooms (e.g., rooms with a blackboard), others ask not to be assigned to certain rooms. Some rooms should not be used on certain days. Because our algorithm is not designed to deal with room preferences, we enforce these preferences through hard constraints. The assignments in the current solution built by the Planning Department, which we use as our starting point, are also assumed to be feasible, so these hard constraints do not lead to infeasibility problems.

The resulting dataset contains 740 lectures followed by 168 different eventgroups.

4.10.2 Graphical user interface

The surrogate-based tabu search heuristic of Chapter 4 has been implemented in a scheduling tool with a graphical user interface (GUI) to facilitate implementation in practice. The GUI is programmed in Qt Creator 4.5.0 with Qt 5.10 and compiled with the MSVC2017 compiler. The C++/Qt source code can be found at the following website: `https://github.com/HendrikBV/A-surrogate-based-tabu-search-heuristic-to-optimise-people-flows-in-a-timetable`.

Figure 4.6 shows the current timetable (i.e., the initial solution) at KU Leuven Campus Brussels. The upper and lower left pane show the timetable and building data in a tree view so that they can be validated by the user. The upper right pane consists of four tabs. The first two tabs represent the timetable in two different ways. In the first tab, each row refers to a room and each column to a timeslot, while in the second tab, each row refers to an eventgroup and again each column to a timeslot (see Figure 4.7). The user can choose to simulate the evacuation time in each timeslot and travel time between each pair of consecutive timeslots for the current solution. Confidence intervals for these evacuation and travel times are then shown in the third and fourth tabs, respectively. Finally, the computational results (e.g., computation time and solution quality) are shown in the lower right pane.

The user can choose the parameters for the tabu search and the surrogate model using the dialog shown in Figure 4.4. If all input data are read

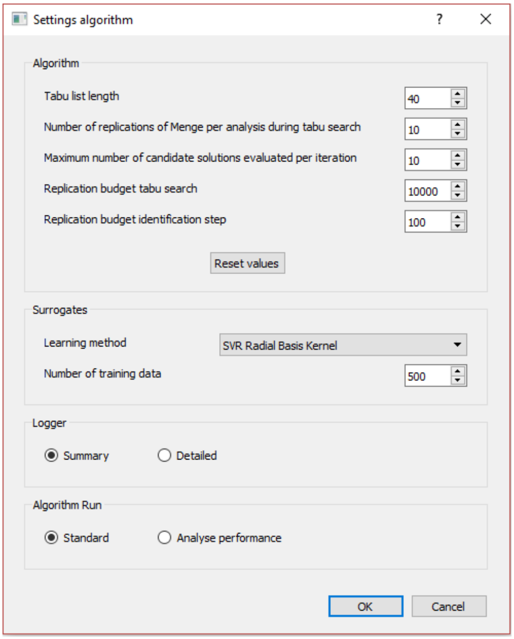


Figure 4.4: GUI: dialog to specify the algorithm parameters.

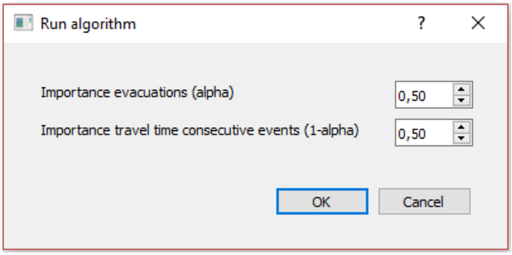


Figure 4.5: GUI: dialog to start the algorithm.

in and the settings are specified, the surrogate-based tabu search can be started. The user can divide a unit weight between two optimisation criteria, namely minimisation of the evacuation times or minimisation of the travel times between events in consecutive timeslots (see Figure 4.5).

Figure 4.8 shows the GUI during the optimisation process. If a new best solution is saved, the changes with respect to the previous solution are shown in green. After the algorithm has finished, the best found solution can be saved to a file. The user can also manually implement changes to the best found solution by clicking on timetable cells and swapping two events or moving an event to an empty cell. The GUI alerts the user whether the move is feasible or not.

Aside from running the surrogate-based tabu search heuristic, the user can also analyse the current solution in different ways. A first possibility is to calculate the egress times and travel times for every timeslot without visualisation. Before the simulation starts, a dialog appears asking the user to specify the number of replications. Because the results of the simulation are stochastic, the user can specify how many replications should be used to construct confidence intervals for the expected egress or travel time in each timeslot. The results for the evacuation times are shown in the third tab in a bar chart (see Figure 4.10). The horizontal axis denotes the different timeslots. For every timeslot, there are three bars that show the lower 95% confidence interval, the average, and the upper 95% confidence interval for the evacuation time, respectively. In the same way, the results for the travel times are shown in the fourth tab (see Figure 4.11).

A second possibility is to simulate either the evacuation process for a specific timeslot t or the process of travelling from events in timeslot t to events in timeslot $t+1$. This allows a more in-depth analysis for timeslots with a high evacuation or travel time. A dialog is shown that asks the user to specify the timeslot to be analysed. For flows between consecutive events, this timeslot refers to the first timeslot t where the people flows originate. If the configuration has been specified, the simulation starts. A new screen is shown in which the building is drawn and every pedestrian is represented by a circle (see Figure 4.12). Users can zoom in or out using the mouse wheel. Next, at every time step, the positions of the pedestrians are updated by Menge (see Figure 4.13). Figure 4.14 shows the end of the simulation. We use building B-16-1 of Figure B.5 instead of the KU Leuven building so that the simulation can be clearly visualised. Figure 4.15 shows the building representation and the initial distribution of pedestrians at the start of the simulation for the evacuation process in timeslot 1 for the case study at KU Leuven Campus Brussels. Figure 4.16 shows the road map that pedestrians use to navigate towards their goals.

Finally, the user can choose different pedestrian models and parameter settings of the simulation through the dialog in Figure 4.17. The percentage of the number of people that have reached their destination that is used to define the evacuation or travel time of the simulation can also be changed.

Figure 4.6: GUI: visualisation of the timetable at KU Leuven Campus Brussels.

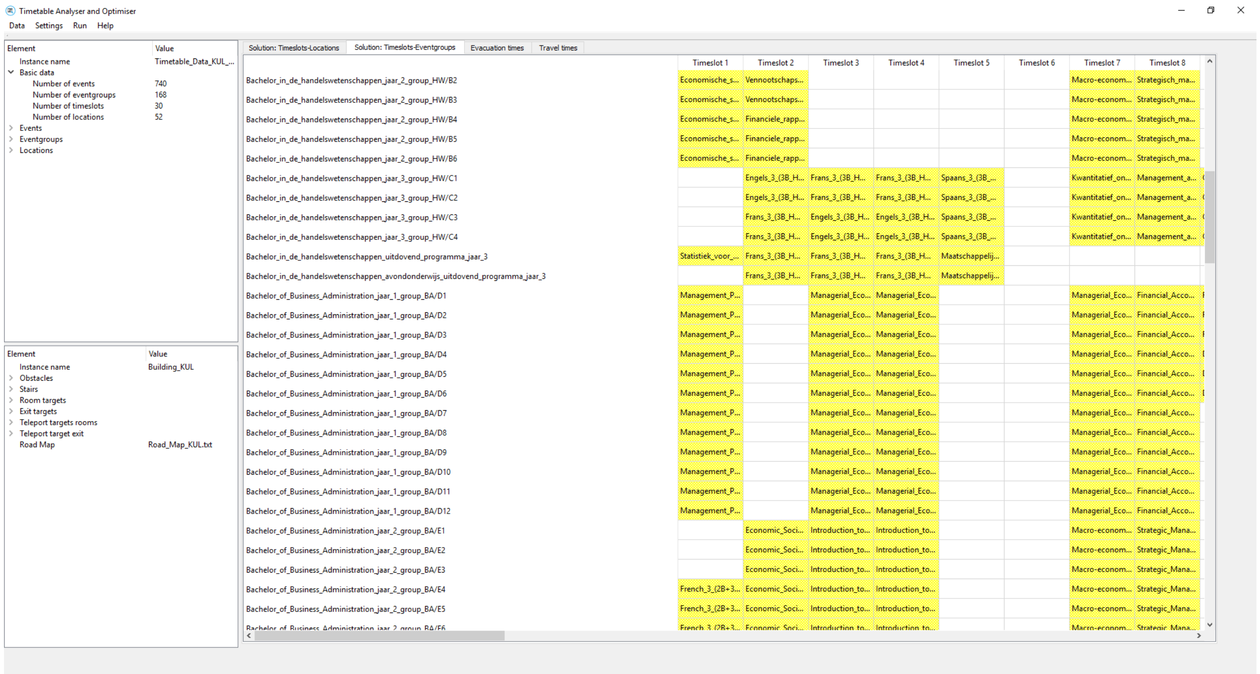


Figure 4.7: GUI: alternative visualisation of the timetable at KU Leuven Campus Brussels.

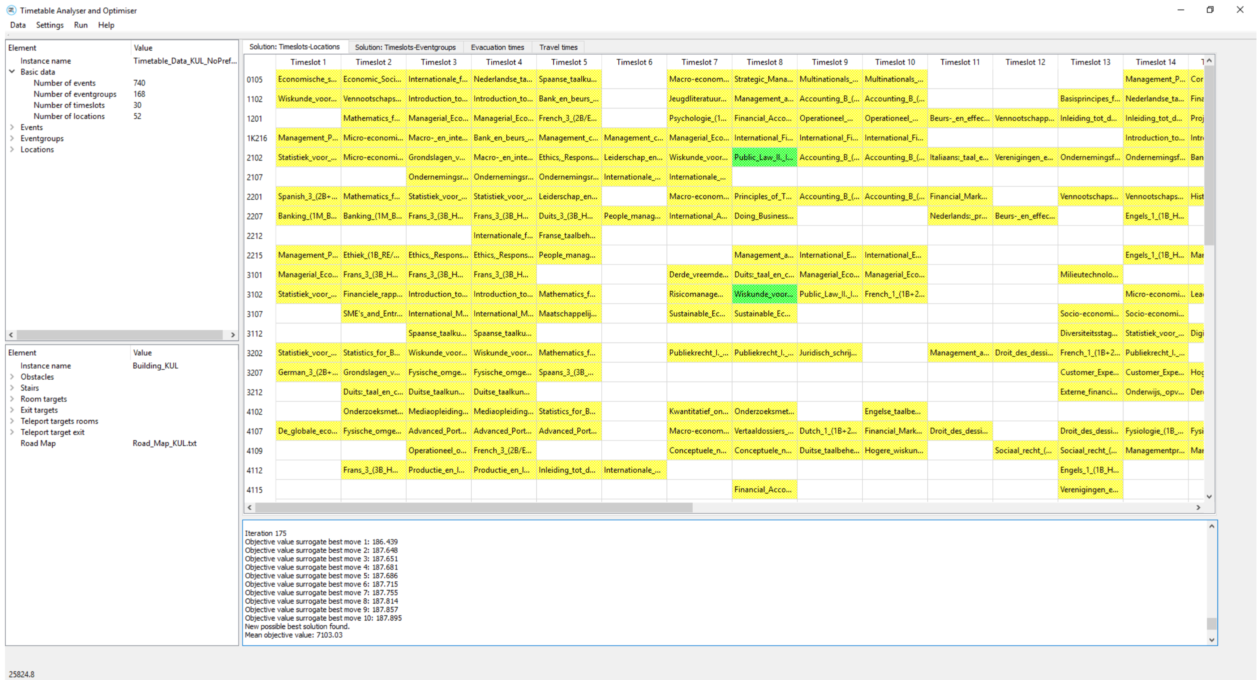


Figure 4.8: GUI: visualisation during algorithm run.

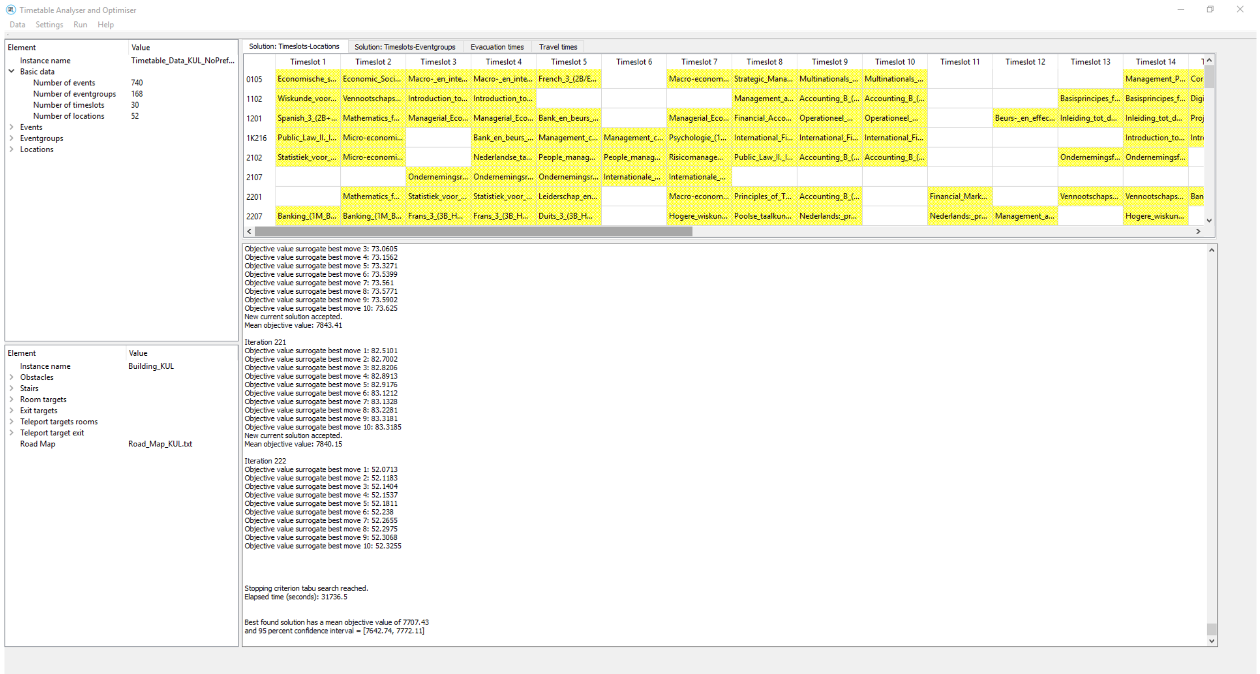


Figure 4.9: GUI: visualisation of the best found solution when optimising the evacuation times without taking the various scheduling preferences of the Planning Department into account.

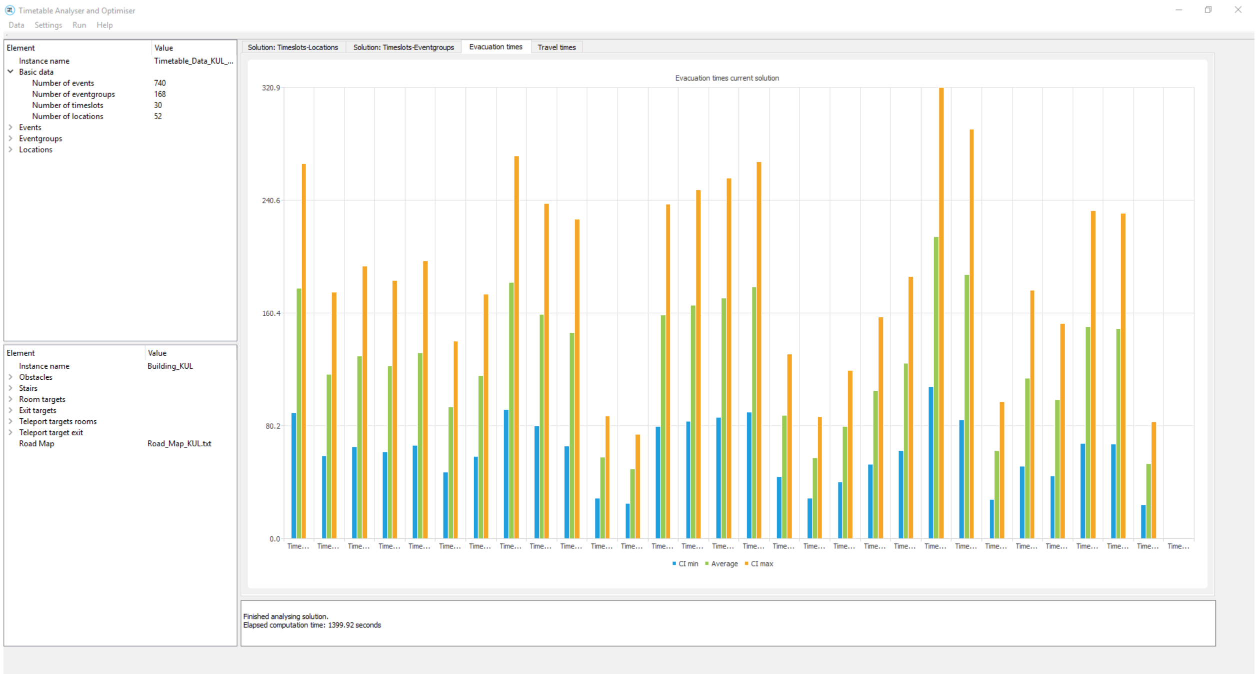


Figure 4.10: GUI: visualisation of the egress times for the initial solution of the KU Leuven instance.

Figure 4.11: GUI: visualisation of the travel times for the initial solution of the KU Leuven instance.

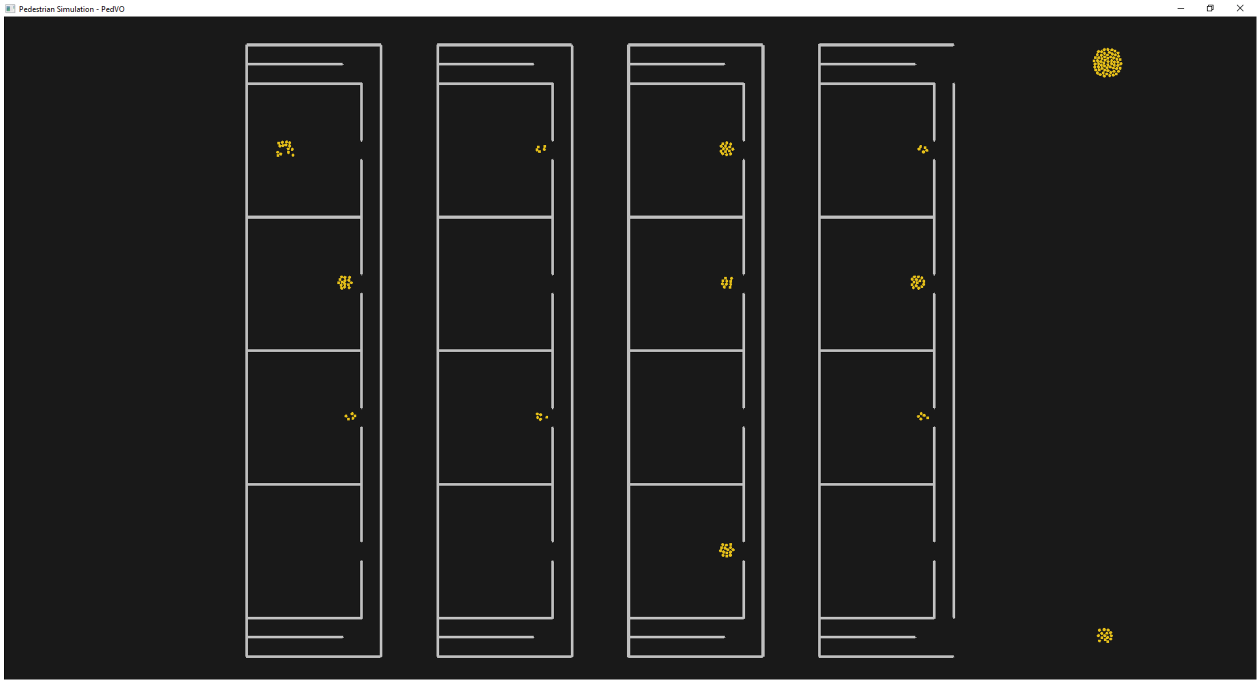


Figure 4.12: GUI: start of the simulation for flows between consecutive timeslots for a random test instance and building B-16-1 of Figure B.5.

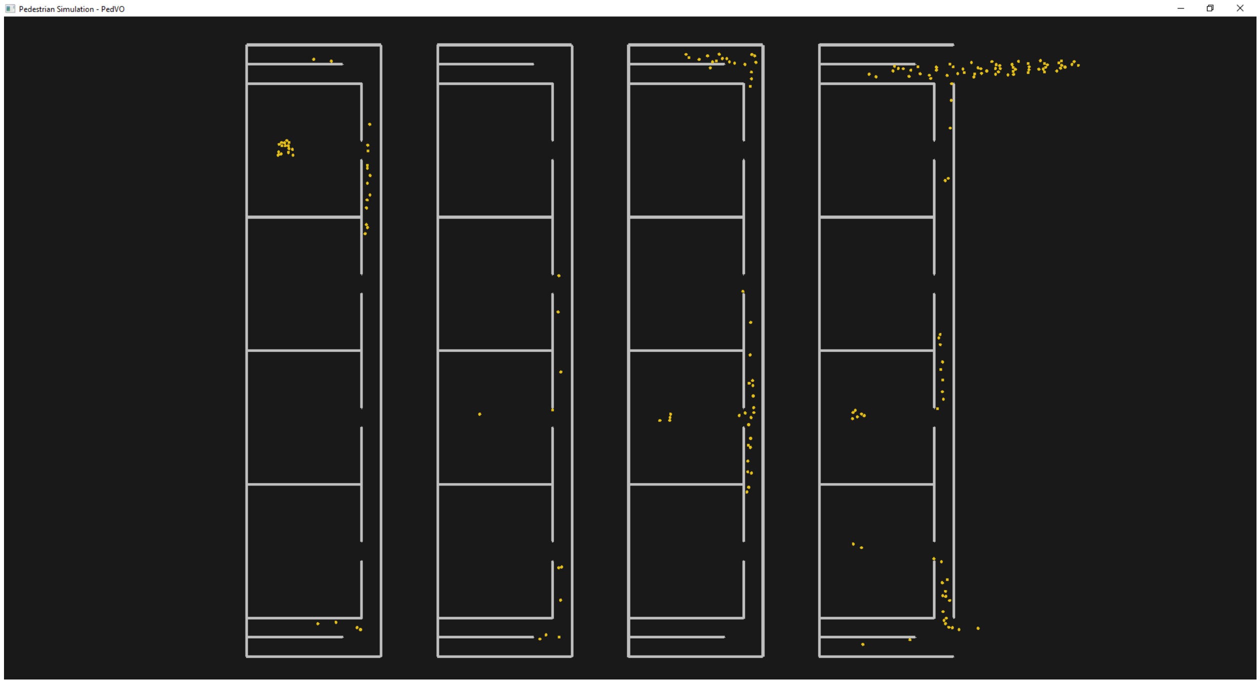


Figure 4.13: GUI: during the simulation for flows between consecutive timeslots for a random test instance and building B-16-1 of Figure B.5.

Figure 4.14: GUI: end of the simulation for flows between consecutive timeslots for a random test instance and building B-16-1 of Figure B.5.

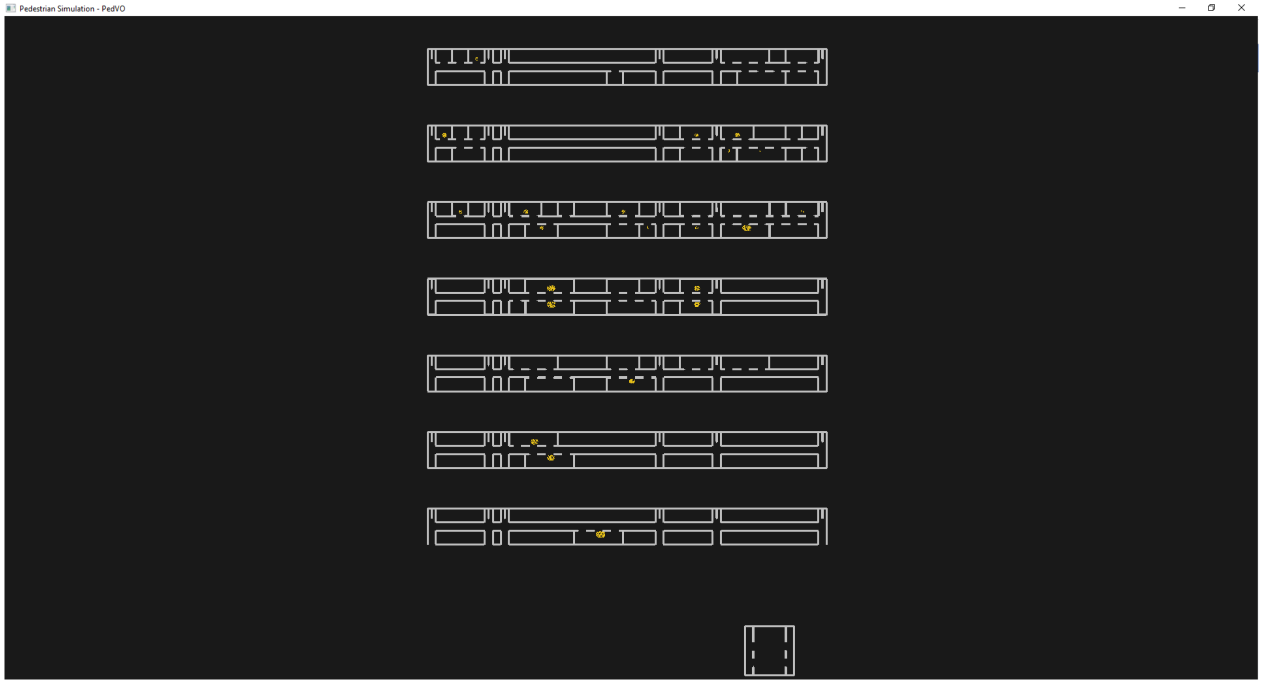


Figure 4.15: GUI: start of the simulation for an evacuation in the first timeslot for the KU Leuven timetable.

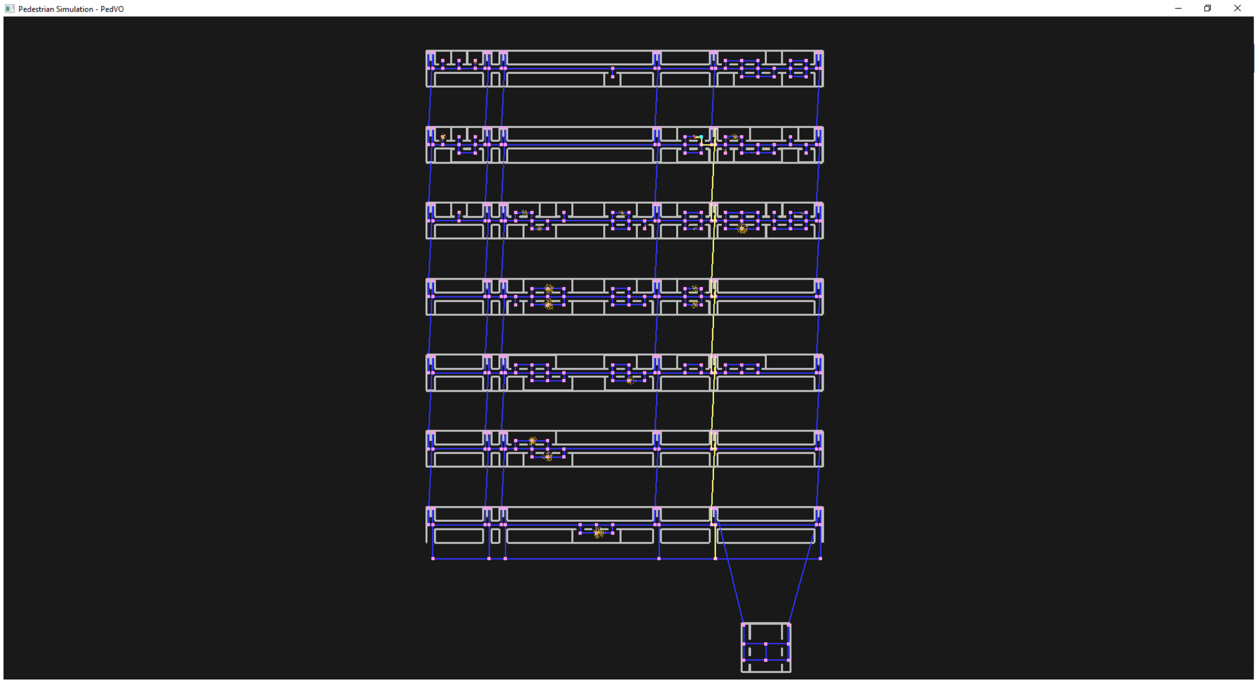


Figure 4.16: GUI: road map for the KU Leuven building.

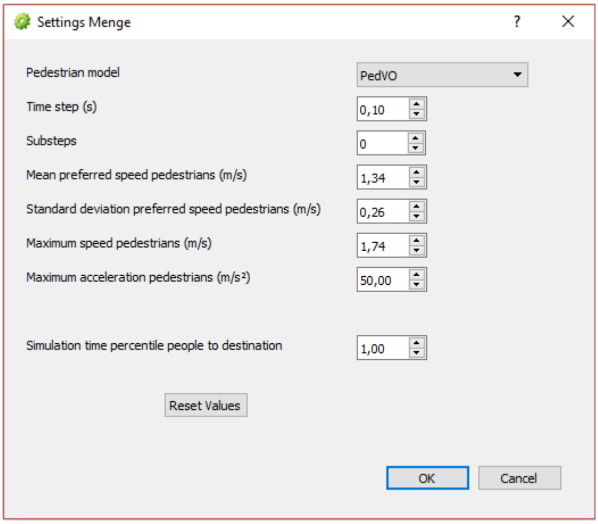


Figure 4.17: GUI: dialog to choose the pedestrian model and the simulation parameters.

4.10.3 Results

We only report the results for the case where we optimise the evacuation times. In a first step, we ran the algorithm taking only the room capacities and the room types (i.e., regular room, PC room, language room) into account. The result of this algorithm run is shown in Figure 4.9. Compared to the current solution proposed by the Planning Department that has a total mean evacuation time of 8369 seconds, the solution found by the tabu search heuristic is a considerable improvement with a total mean evacuation time of 7707 seconds.

However, as was mentioned in Section 4.10.1, when we showed this solution to the Planning Department they indicated that it violates various scheduling preferences. We then adjusted the list of feasible lecture-room combinations and reran the algorithm. However, in this case the heuristic did not find a solution that is a significant improvement compared to the solution proposed by the Planning Department. This can mean that either our algorithm fails on this instance or that the solution of the Planning Department is already very good taking into account all the other scheduling constraints that need to be met.

4.11 Conclusions and future research

This chapter presents a surrogate-based tabu search heuristic to assign events to rooms in a generic timetabling problem to improve the resulting people flows. The people flows are simulated using the microscopic Menge simulator. The tabu search uses a simple neighbourhood move that randomly selects a timeslot and evaluates all feasible swaps of events between rooms in that timeslot. Instead of evaluating all candidate solutions in each iteration with the computationally expensive Menge simulator, a surrogate model is used to quickly estimate the objective value of a candidate solution.

The heuristic is applied two different problem types to show that it can be used to tackle a wide variety of problems where timetabling decisions have an impact on people flows. The first example is an evacuation problem, where all people attending events in timeslot t need to be evacuated to safety. The second example considers the problem of optimising people flows between events in consecutive timeslots.

For both problems, eight different combinations of machine learning technique and kernel type are compared to fit the best surrogate model. The predictions of the surrogate models are also validated on all candidate solutions in three different iterations for four problem instances. Extensive computational tests show that the heuristic succeeds in finding solutions with significantly improved people flows for both problem types.

Next, the heuristic is implemented in a scheduling tool with a GUI and applied to the timetabling problem at KU Leuven Campus Brussels. We focused on the objective function where we want to minimize the evacuation times over all timeslots. When the heuristic only needs to take the type and capacity of the rooms into account, it succeeds in finding a solution with a significantly improved evacuation times. However, when all scheduling preferences listed by the Planning Department are taken into account as hard constraints, the model does not find a solution that is better than the current solution proposed by the Planning Department. Still, these results show that the model can be applied to large problem instances with many pedestrians without problems. We believe that this software tool is valuable in illustrating how the algorithm works and in visualising the proposed solution to the different stakeholders involved in the project.

A first direction for future research is the extension of the model to include a constraint requiring one or more events (in different timeslots) to be planned in the same room. For example, in a conference schedule, talks on the same topic should be planned sequentially in the same room. In this case, swapping sessions between rooms is probably more realistic than swapping individual events. Secondly, the use of a more sophisticated identification criterion might improve the probability of returning the truly best solution encountered during the search process.

Another valuable future research direction is to validate the model in practice by performing real-life experiments, as the model remains only theoretical. Unfortunately, due to the complexity and scale of the problem, this is not easy to achieve in practice. Additionally, the current problem has made abstractions of constraints that might be present in the real-life problem. While the model can accommodate some scheduling constraints by indicating which events can(not) be assigned to which rooms, this is not possible for all types of scheduling constraints, such as requiring that two or more lectures are planned in the same room. To encourage implementation in practice, the model could be extended to accommodate these requirements.

Chapter 5

Comparison of the pedestrian models of Chapter 3 and Chapter 4

5.1 Introduction

In the previous two chapters we have developed two different scheduling methods to build timetables to improve travel times between events in consecutive timeslots or evacuation times in the event of an emergency. In Chapter 3, we have used a macroscopic network model to describe the people flows. This macroscopic model treats the crowd as a whole. By way of contrast, in Chapter 4, we have used the microscopic Menge simulator to describe the people flows. In this microscopic model, each individual is represented as a separate entity. Up until now, we have mostly focused on the computational properties of the different optimisation models, but we have not compared both pedestrian models in detail.

In this chapter, we compare the results predicted by both pedestrian models. We also investigate the robustness of both models. The remainder of this chapter is structured as follows. In Section 5.2, we compare how both models evaluate different timetable solutions. Then in Section 5.3, we analyse the robustness of the network model in greater detail, followed by an analysis of the robustness of the Menge simulator in Section 5.4. Next, Section 5.5 briefly discusses the modelling power of each of the two models. Finally, Section 5.6 concludes the chapter.

5.2 Comparison of both models

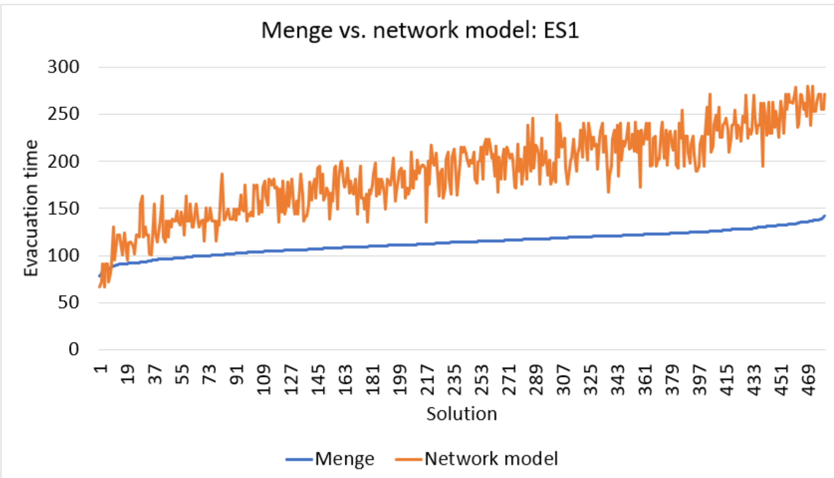
We compare the results of the Menge simulator of Chapter 4 with the results of the network model of Chapter 3. We use exhaustive search on timetable instances ES1, ES2, ES3, ES4, and ES5 and building B-8-2 and minimise the evacuation time. The objective values calculated by the Menge simulator are based on 10 simulation replications.

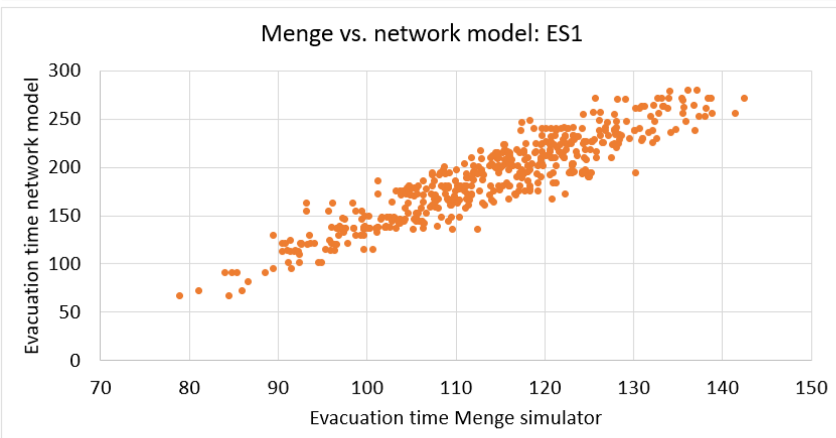
Figure 5.1 shows the results for instance ES1. While the correlation between the objective values calculated by both models for every solution in the solution space is high (0.92), there is nevertheless considerable variation in valuations of individual solutions. This is clear from the first graph of Figure 5.1, where solutions are sorted according to increasing objective values calculated with the Menge simulator. The objective values calculated by the network model vary widely between adjacent solutions. This can also be seen in the the second graph of Figure 5.1, where solutions that have the same objective value according to one model, can have different valuations according to the other model. The same trends can be observed for the other instances, as is shown in Figure 5.2.

We can take a closer look at the individual solutions. Table 5.1 compares

Figure 5.1: Comparison between the network model of Chapter 3 and the Menge simulator of Chapter 4 for instance ES1.

The tests use exhaustive search on timetable instance ES1 in combination with building B-8-2. For the Menge simulator, 10 simulation replications were used to calculate the objective values. The first figure shows the objective values calculated by both models for every solution in the solution space. The solutions are sorted according to increasing objective values calculated by the Menge simulator. The second figure shows the correlations between objective values calculated by both models.





the best three solutions identified by the network model with the best three solutions identified by the Menge simulator for instances ES1, ES2, ES3, ES4, and ES5 in combination with building B-8-2. Let $a-b-c-d$ denote a solution for instance ES1 where event 1 is assigned to room a, event 2 to room b, event 3 to room c and event 4 to room d. The network model ranks solution R12 – R21 – R22 – R11 and solution R12 – R22 – R21 – R11 as the best solutions with both an evacuation time of 66.8 seconds. This intuitively makes sense as these rooms are closest to the exit of the building (see Figure B.4). The best solution according to Menge is solution R12 – R22 – R21 – R11 with an evacuation time of 78.9 seconds, while solution R12 – R21 – R22 – R11 has an evacuation time of 84.5 seconds (the 4th best solution according to Menge). So both models actually agree on which solution is the best one for instance ES1.

From the results of Table 5.1, three observations can be made. First, in general, solutions that have a good ranking according to one model also have a good ranking according to the other model. Second, for every problem instance, many solutions in the solution space have more or less the same objective values. This is one explanation for the patterns in Figures 5.1 and 5.2. Third, the evacuation time predicted by the network model seems to be determined in large part by the distance of the longest route used by people evacuating the building. For example, for instance ES2, the network model predicts that solution R42–R12–R22–R32–R11 has a significantly higher evacuation time than the best solutions, because one group of people needs to travel from the fourth floor to the exit of the building. By contrast, the Menge simulator predicts that that same solution has more or less the same evacuation time as the best solutions. The results of the Menge simulator clearly depend more on the interactions of pedestrians and the resulting congestion, than on the distance of the longest route.

Figure 5.2: Comparison between the network model of Chapter 3 and the Menge simulator of Chapter 4 for instances ES2, ES3, ES4, and ES5.

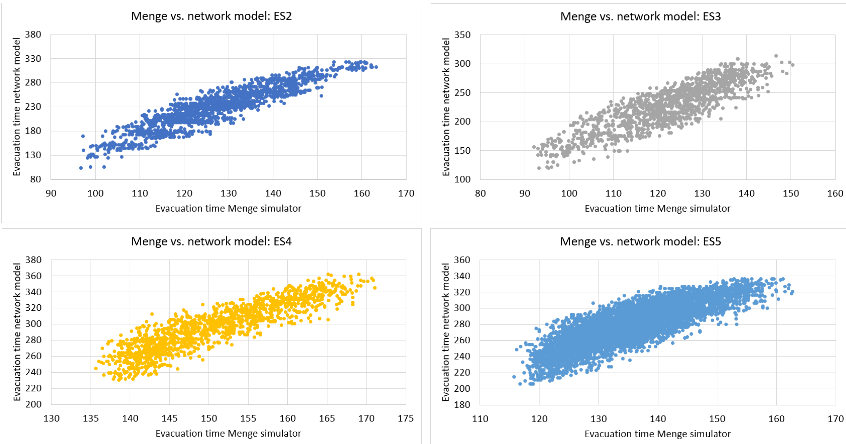


Table 5.1: Comparison of the best solutions identified by the network model of Chapter 3 and the Menge simulator of Chapter 4 for different problem instances.

The tests use exhaustive search on timetable instances ES1, ES2, ES3, ES4, and ES5 in combination with building B-8-2. For the Menge simulator, 10 simulation replications were used to calculate the objective values. 'Size' indicates the number of solutions in the solution space. The solution representation $a - b - ... - n$ means that event 1 is assigned to room a, event 2 to room b and event |E| is assigned to room n. For every instance, the best three solutions according to the network model and the best three solutions according to the Menge simulator are listed and compared. The solutions are sorted according to increasing objective values calculated by the network model.

Instance	Size	Solution	Network model		Menge	
			ET	Rank	ET	Rank
ES1	480	R12-R22-R21-R11	66.8	1	78.9	1
		R12-R21-R22-R11	66.8	1	84.5	4
		R12-R32-R21-R11	72.6	3	81.1	2
		R12-R32-R22-R11	72.6	3	85.9	7

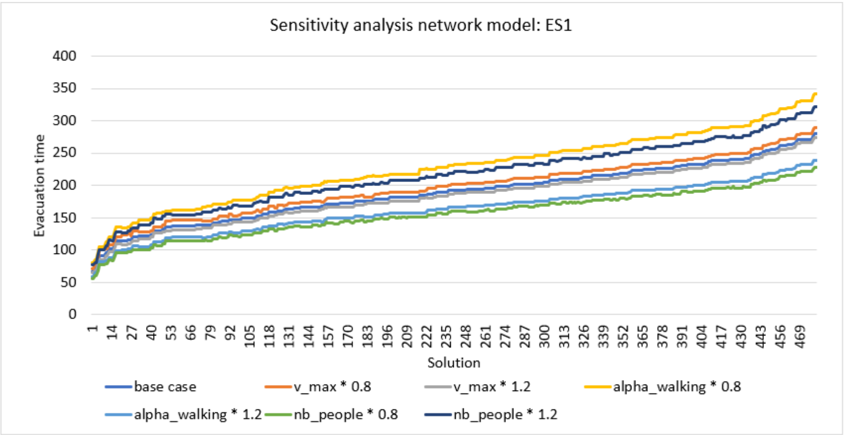
Continued on next page

Table 5.1 – continued from previous page

Instance	Size	Solution	Network model		Menge	
			ET	Rank	ET	Rank
		R12-R41-R22-R11	91.4	6	84.0	3
ES2	1608	R21-R12-R22-R32-R11	103.4	1	96.8	1
		R12-R11-R22-R32-R21	105.8	2	98.9	6
		R11-R12-R22-R32-R21	105.8	2	101.9	27
		R21-R12-R31-R32-R11	140.6	21	97.3	3
		R42-R12-R22-R32-R11	169.4	125	97.2	2
ES3	1379	R11-R31-R12-R21-R22-R32	120	1	93.2	4
		R11-R31-R12-R22-R21-R32	120	1	95.4	19
		R21-R11-R12-R22-R31-R32	121.4	3	95.0	14
		R22-R11-R12-R21-R31-R32	121.4	3	96.0	24
		R22-R11-R12-R21-R42-R31	142.6	23	93.0	2
		R21-R11-R12-R22-R42-R41	153.4	52	93.0	3
		R42-R11-R12-R21-R31-R22	156.2	56	92.1	1
ES4	1386	R11-R42-R21-R32-R22-R12-R31	230.8	1	138.6	37
		R12-R42-R11-R21-R22-R32-R31	232	2	137.8	20
		R11-R42-R12-R21-R22-R32-R31	232	2	139.4	69
		R21-R42-R11-R32-R22-R12-R31	232	2	140.3	118
		R11-R31-R12-R21-R22-R32-R41	245.2	43	135.7	1
		R12-R42-R21-R11-R41-R22-R31	252.6	93	136.4	3
		R22-R42-R21-R11-R41-R12-R32	253.8	103	136.0	2
ES5	5668	R12-R22-R11-R32-R42-R31-R41-R21	206.2	1	116.8	3
		R12-R22-R11-R31-R42-R32-R41-R21	206.2	1	118.5	22
		R11-R22-R12-R31-R42-R32-R41-R21	206.2	1	118.7	28
		R11-R22-R12-R32-R42-R31-R41-R21	206.2	1	118.9	35
		R11-R22-R12-R32-R41-R42-R31-R21	215.2	30	115.7	1
		R11-R32-R12-R21-R31-R42-R22-R41	248.8	702	116.1	2

Figure 5.3: Sensitivity analysis for the network model of Chapter 3.

The tests use exhaustive search on timetable instance ES1 in combination with building B-8-2. Three parameters are changed, viz. v_{max} (the maximum free walking speed), α (the scaling parameter in equation (3.1) for the relationship between walking speed and density), and the number of people in each event.



5.3 Robustness network model Chapter 3

We test the robustness of the network model of Chapter 3 by changing three parameters of the model. The parameters that are changed are v_{max} (the maximum free walking speed in zero density), α (the scaling parameter in equation (3.1) for the relationship between walking speed and density), and the number of people that attend each event. We use exhaustive search on timetable instance ES1 in combination with building B-8-2 and minimise the evacuation time. The results are displayed in Figure 5.3.

Each of the parameters has an impact on the absolute size of the objective values for each solution, but the rank order of the solutions remains the same for virtually all solutions in the solution space for every change in parameter values. This implies that the network model is quite robust, since measurement or estimation errors in the parameters do not have an impact on the ability of the optimisation model to identify good solutions.

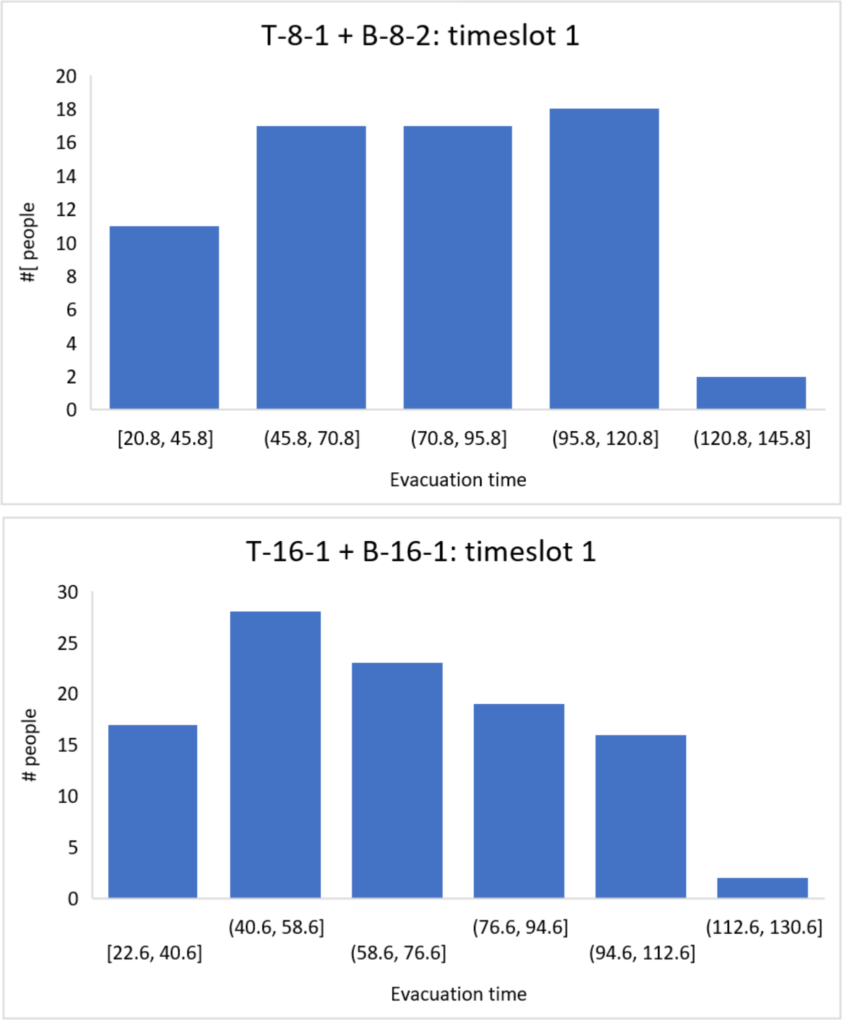
5.4 Robustness Menge simulator Chapter 4

We also test the robustness of the Menge simulator. First, in Section 5.4.1 we investigate the impact of changing the definition of the evacuation or travel time, based on the number of people that have reached their destination. Next, in Section 5.4.2 a sensitivity analysis is performed on the parameters of the Menge simulator and the number of people in the problem instance.

5.4.1 The impact of the definition of the evacuation or travel time

In Chapter 4, we always used the time that the last person arrives at his or her destination as the evacuation or travel time for the simulation. However, because a microscopic simulator represents every person as a separate entity, we can also define the evacuation or travel time as the time when a given percentage of the people in the simulation have reached their destination. Figure 5.4 shows the distribution of the times when the different agents in the simulation reach the exit of the building during an evacuation. For instance T-8-1 with building B-8-2, the average evacuation time for timeslot 1 is 75.9 seconds, while the evacuation time of the last person is 135 seconds. For instance T-16-1 with building B-16-1, the average evacuation time for timeslot 1 is 66.7 seconds, while the evacuation time of the last person is 121 seconds.

Figure 5.4: The distribution of the times when the different agents in the simulation reach the exit of the building.



The coefficients of variation (COV) for the evacuation or travel times as a function of the percentage of people that have reached their destination over different simulation runs are shown in Figure 5.5. It is clear that the COV is much larger if the evacuation or travel time is defined as the time when the last person has reached his or her destination, compared to when it is defined as the time when 50 or 95 percent of people have reached their destination. We could thus reduce noise during the optimisation by opting to define the evacuation or travel time as the time when e.g. 95 percent of people have reached their destination instead of 100 percent.

We again compare the predicted ranking of all solutions in the solution space of instance ES1. If we define the evacuation time as the time when 95 percent of people have reached safety, Menge predicts solution R12 – R21 – R22 – R11 to be the best solution with an evacuation time of 72 seconds, which is also one of the two best solutions according to the network model. If the evacuation time is defined as the time when 50 percent of people have reached safety, the two best solutions according to the network model have predicted evacuation times of 49.5 and 49.6 seconds according to Menge, while the best solution has an evacuation time of 48.1 seconds.

5.4.2 Sensitivity analysis parameters

Next, we analyse the impact of changing three parameters of the model, namely the mean preferred free walking speed, the standard deviation of the preferred free walking speed, and the number of people in the problem instance. We again use exhaustive search on timetable instance ES1 in combination with building B-8-2 and minimise the evacuation time. Figure 5.6 displays the results for the mean preferred free walking speed, Figure 5.7 shows the results for the standard deviation of the mean pre-

Figure 5.5: The coefficients of variation for the evacuation or travel time as a function of the number of people that have reached their destination.

'ET' refers to the evacuation time in a given timeslot. 'TT' refers to the travel time between two consecutive timeslots.

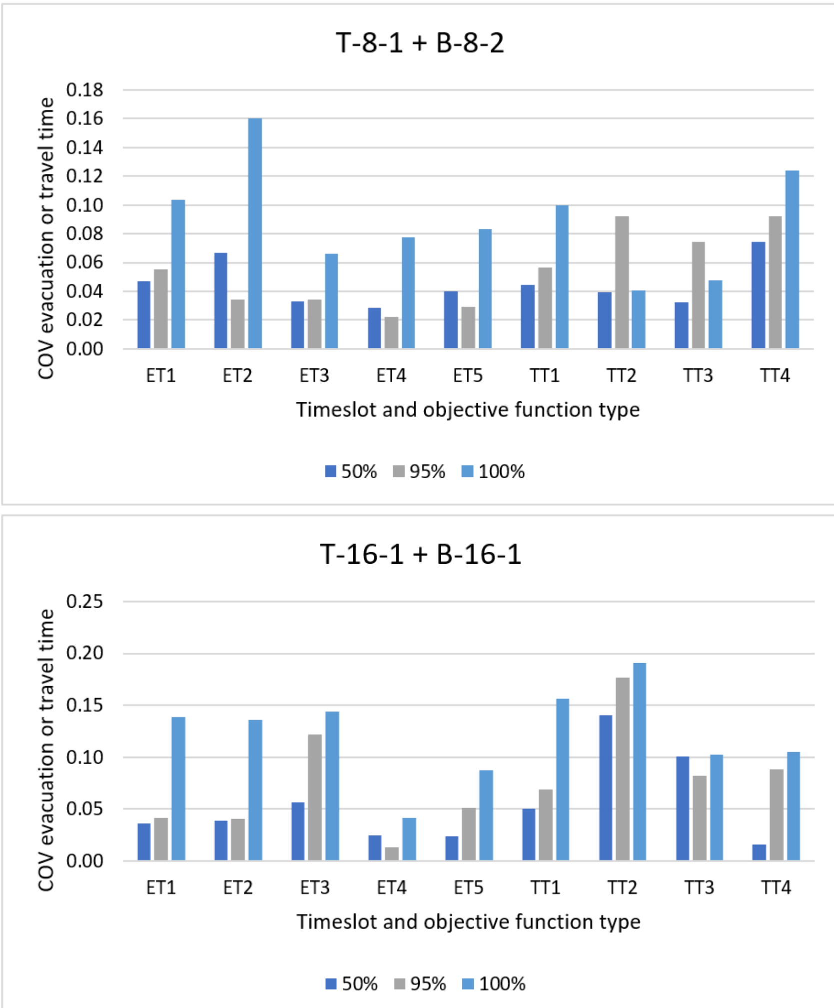
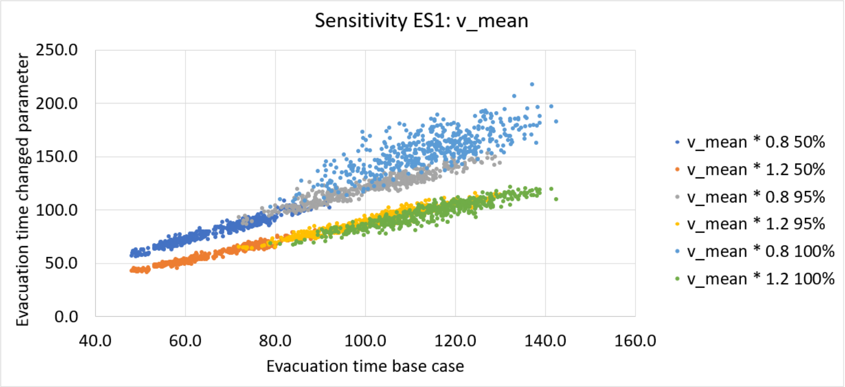


Figure 5.6: Sensitivity analysis of the Menge simulator for the mean preferred free walking speed.



ferred free walking speed, and Figure 5.8 illustrates the results for the number of people in the instance.

In contrast to the network model of Chapter 3, the Menge simulator of Chapter 4 is considerably more sensitive to changes in parameters. Not only the absolute value of the solutions is impacted, but also their relative ranking. This holds true for changes to each of the three parameters. Moreover, the sensitivity is much larger for the case where the evacuation time is defined as the time when the last person reaches safety, than for the case where the evacuation time is defined as the time when either 50 or 95 percent of people have reached safety.

5.5 Modelling power

The Menge simulator of Chapter 4 can easily be used to analyse more complex scenarios. We consider the following two scenarios:

1. **Include incoming fire-fighters.** This can be done by adding

Figure 5.7: Sensitivity analysis of the Menge simulator for the standard deviation of the preferred free walking speed.

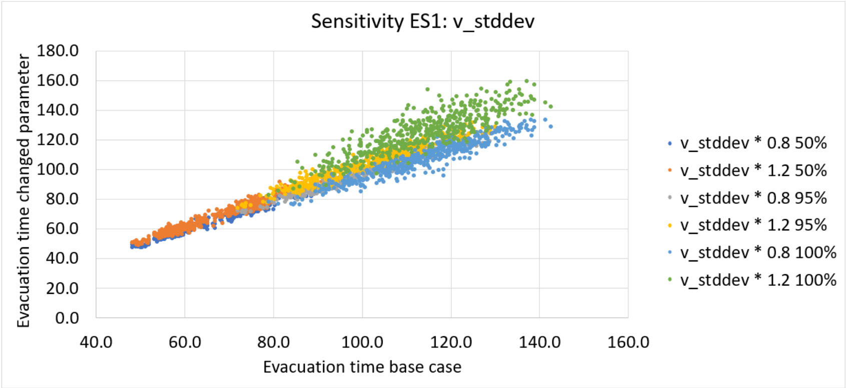
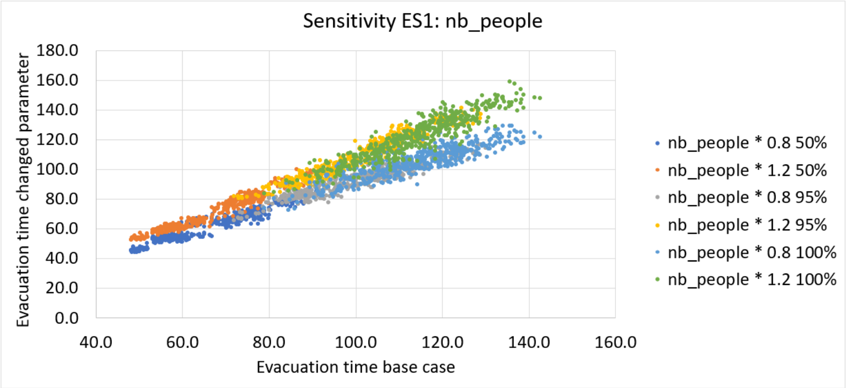


Figure 5.8: Sensitivity analysis of the Menge simulator for the number of people in the problem instance.



one or more *agent groups* that have as their *goal* the area(s) of the building where there is fire.

2. **Inability to use a certain stairwell or hallway of the building due to fire.** This can be achieved by removing one or more arcs in the *road map*. The wayfinding algorithm will then calculate a different route towards the exit for each agent, so that the area of the building where there is a fire, is avoided. Unfortunately, the road map implementation in Menge is static, so this can only be set at the start of the simulation. It is not possible to dynamically change the road map, based on how the fire is spreading through the building. It is possible, however, to define a different roadmap for every (group of) pedestrian(s), so that for every group of pedestrians different possible route choices are available.

These possibilities are illustrated in Figures 5.9 - 5.14. In the base case building B-16-1 is evacuated in a standard way where both exits and stairwells of the building can be used. The initial distribution of people over the building is shown in Figure 5.9. The evacuation process for the base case is shown in Figure 5.10. In a second scenario, we assume that the upper stairwells and exit of the building cannot be used due to a fire. We also assume that a group of firefighters travels to each floor of the building to extinguish the fires. Figure 5.11 shows the middle of the evacuation process. The yellow circles represent people that leave the building, while the green circles represent firefighters who enter the building. Figure 5.12 shows a later moment when most of the people have reached safety, while the firefighters have reached their designated area. In a third scenario, we assume that firefighters cannot use the stairwell where there is fire, but instead use the other stairwell to travel to each floor. Then the flow of firefighters crosses the flow of people evacuating the building, as can be seen in Figure 5.13. In Figure 5.14, we see the same scenario at a later time, when the firefighters have almost reached their designated areas.

It is more difficult to include such scenarios into the network model. We can, however, model the inability to use certain stairwells or hallways in a building in the same way, by removing certain arcs from the network.

Figure 5.9: Evacuation base case: start.

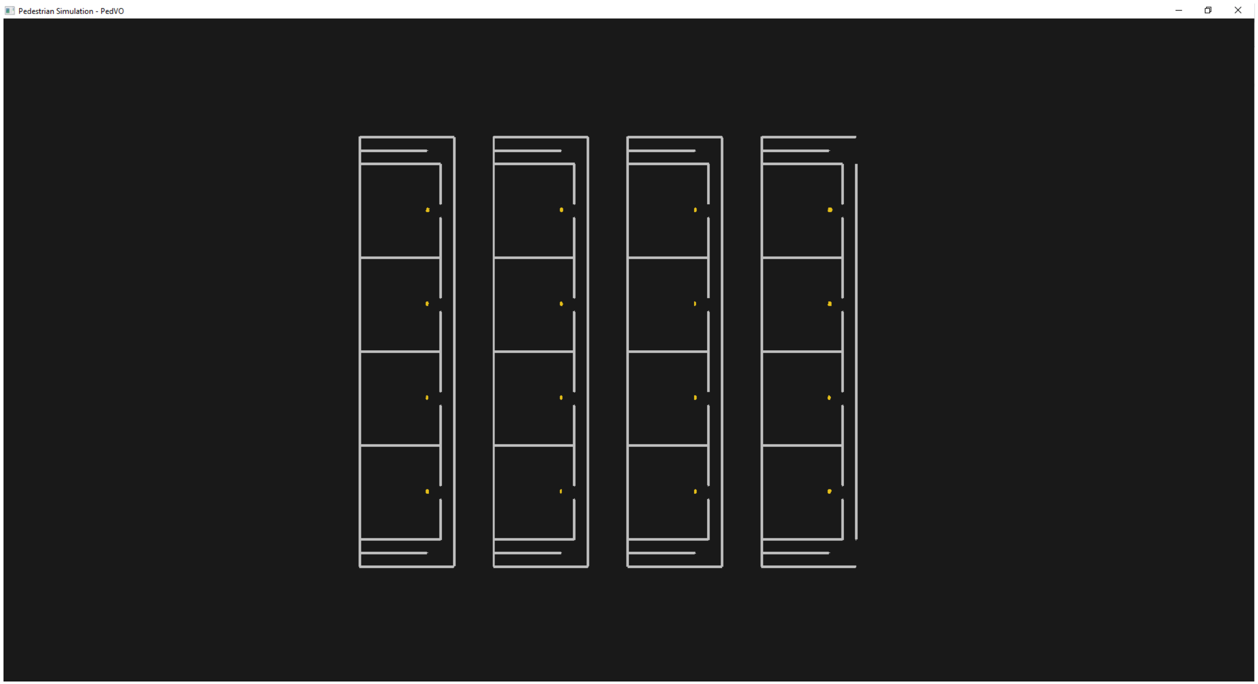


Figure 5.10: Evacuation base case: middle.

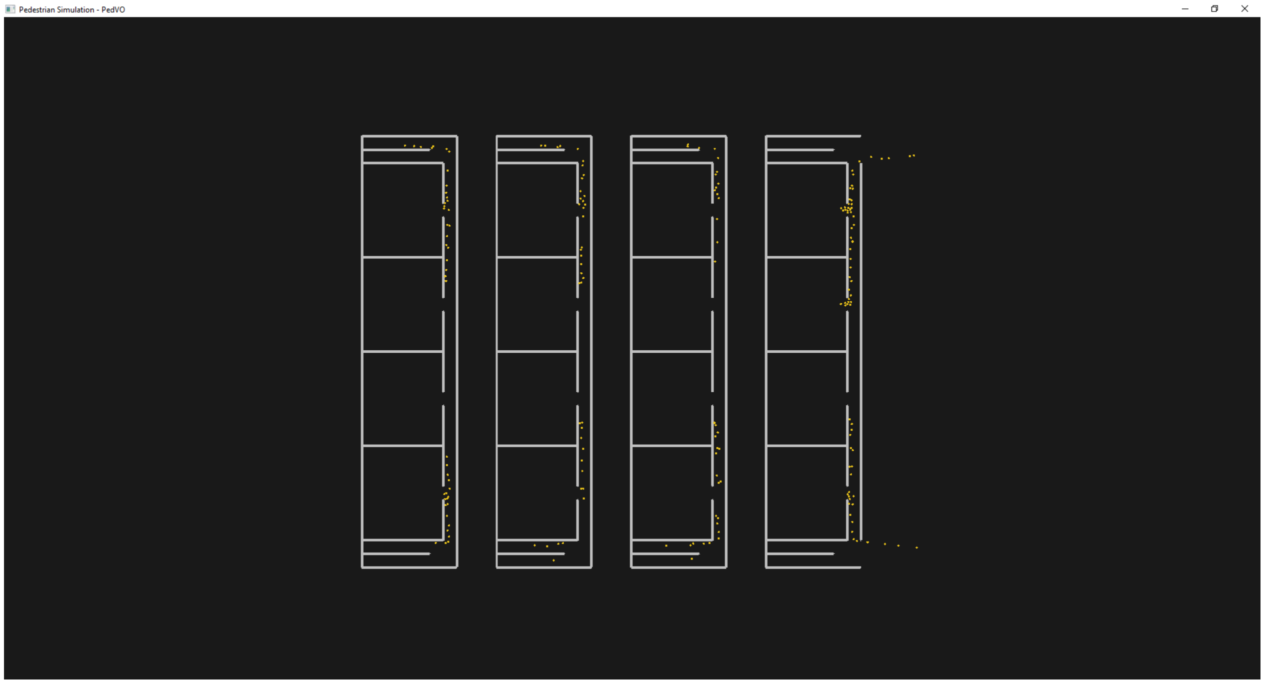


Figure 5.11: Evacuation fire upper stairwell with firefighters (1): middle.

Figure 5.12: Evacuation fire upper stairwell with firefighters (1): end.

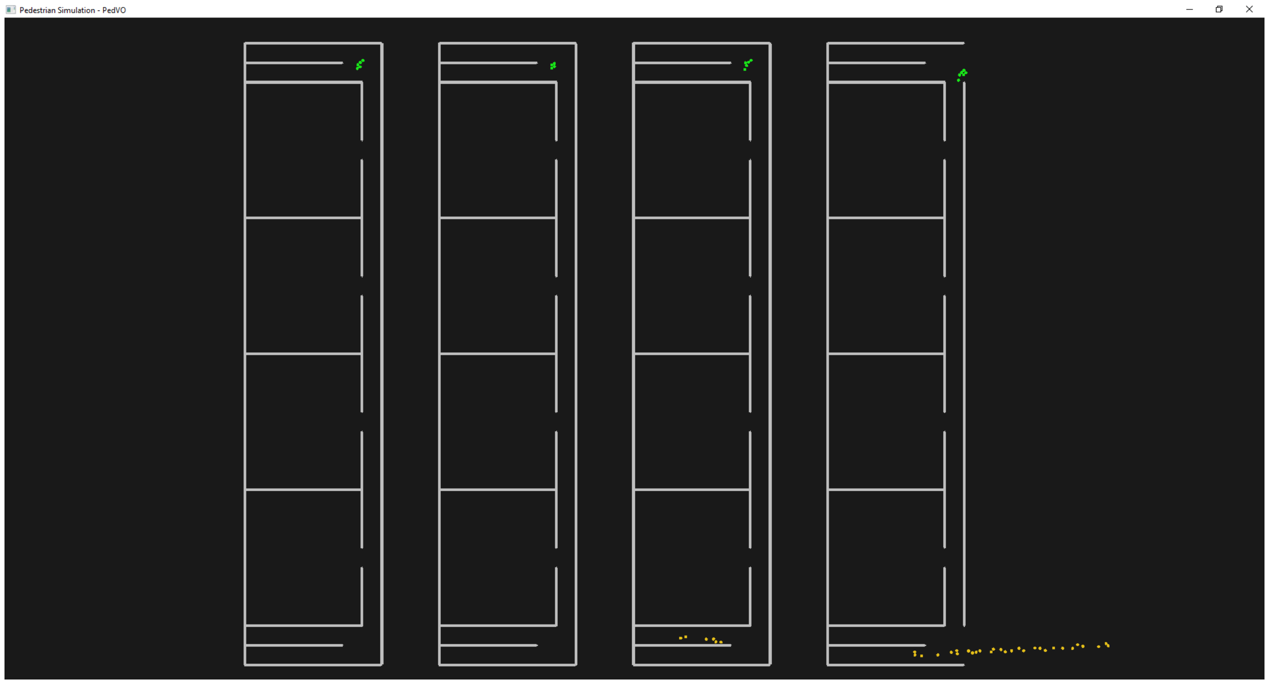


Figure 5.13: Evacuation fire upper stairwell with firefighters (2): middle.

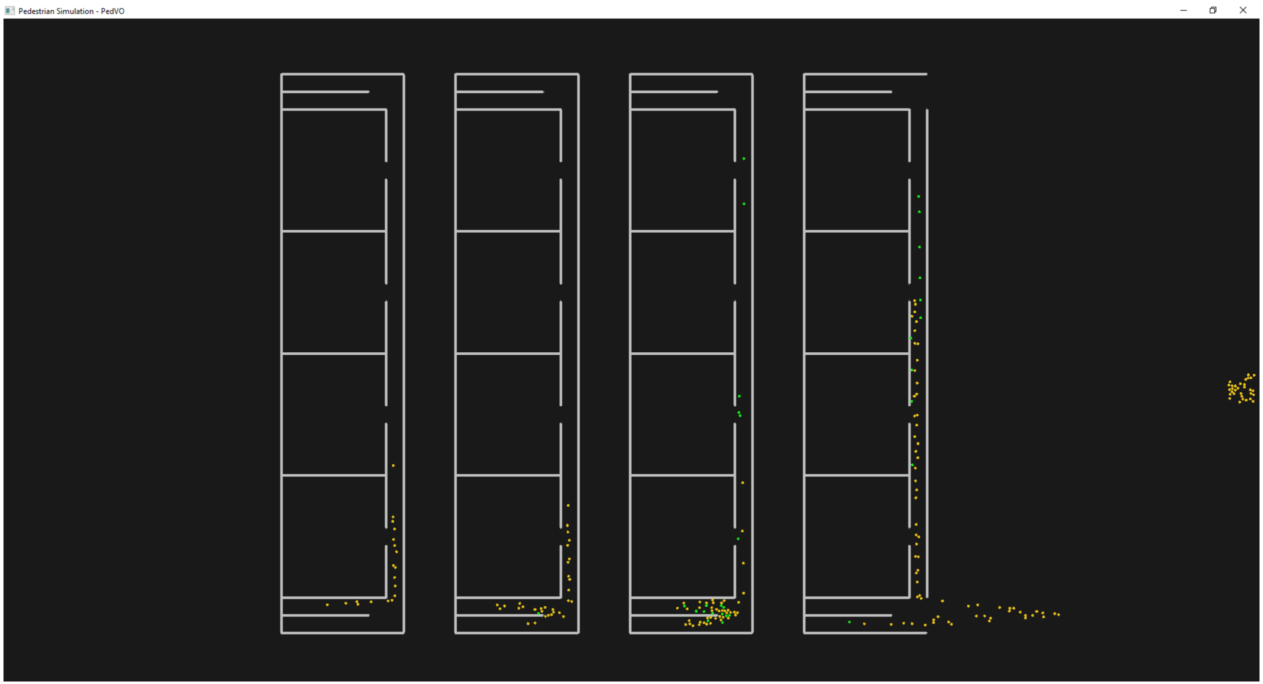
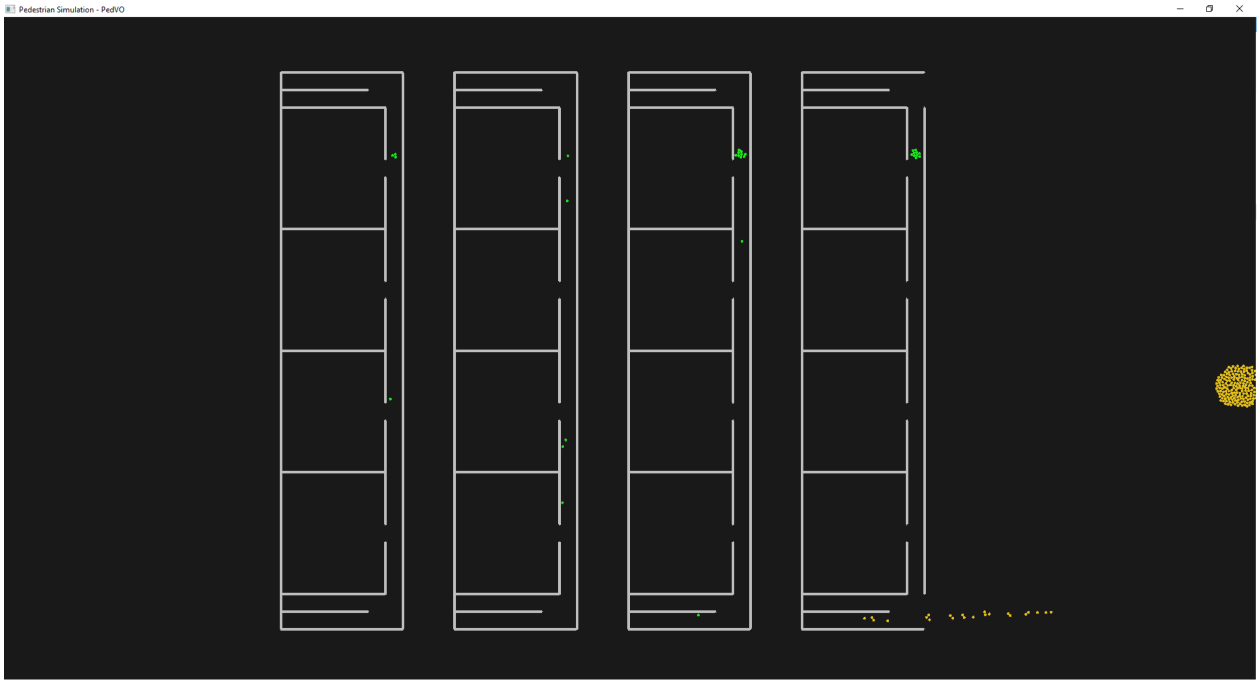


Figure 5.14: Evacuation fire upper stairwell with firefighters (2): end.



5.6 Discussion and conclusion

In this chapter we have compared the network model of Chapter 3 and the microscopic Menge simulator of Chapter 4. To apply the former model to a problem instance, we need to represent a building as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where the nodes $\mathcal{N}$ represent origins, destinations, and junctions, and the arcs $\mathcal{A}$ represent hallways and stairwells in the building. We need to enumerate all possible paths in the graph and list the arcs that are on each of those paths. For every origin-destination pair (i.e., going from one room to another room or going from one room to one of the exits), we need to indicate which paths one can take and the percentage of people that are expected to take each of those paths. We also need to specify the length and surface area of each arc (i.e., the physical space in the building that the arc represents) and whether the arc represents stairs or not. To use the latter model, we need to specify the coordinates of all obstacles such as walls on their actual scale. We also need to define goals for the rooms and exits in the building. Finally, we have to define either a road map, a navigation mesh, or a potential field for pedestrians to find their way through the building from their origin to their destination. In our implementation, we have used a road map, which is a graph comparable to the network model of Chapter 3.

Concerning the calibration of the models, our implementation of the network model requires the specification of three parameters, namely v_{max}, i.e. the maximum free walking speed of pedestrians in zero density, α, i.e. the scaling factor for the relationship between walking speed and density (see equation 3.19), and γ, i.e. the correction term for the walking speed on stairs compared to horizontal surfaces. For the Menge simulator on the other hand, each of the different pedestrian models has different parameters that need to be specified. However, the most important parameters that are shared by all models are the mean of the preferred free walking speed and the standard deviation of the preferred free walking speed. We have used data from the literature for these parameter settings (Helbing and Molnár, 1995). However, it is not clear that the parameter values obtained from these studies are the values of the population in our problem. The values for the students at KU Leuven Campus Brussels could be different from the values in the literature. Indeed, even in the literature there are large differences between values of different studies (see Section 2.3.1). Ideally, experiments should be conducted to obtain the real-world values applicable to our problem. However, such a study is outside the scope of this thesis.

Apart from the choice of parameter values, other assumptions could have an impact on how closely the results of the models reflect the real world. The assumption of which routes students use can have a large effect on the resulting travel or evacuation times. In the network model, we need to predict the percentage of students that use each of the possible paths to go from location A to location B. In the Menge simulator, the standard wayfinding algorithm assumes that people always take the shortest route. This assumption is clearly not realistic, as observations have shown that during evacuations people often take the same route they used to enter the building or follow other people during an emergency, although shorter or less crowded routes are available (e.g., Helbing et al., 2005). Again, empirical studies should be conducted to observe the actual behaviour of the population in our problem setting. Other assumptions that can impact the results, are the number of people assumed to be in the building at any given time and whether people all start walking at the same time or at different times.

The previous sections have shown that both models generally agree on the quality of the best solutions, but that there is considerable variation in evaluation for the other solutions. Moreover, the network model is much less sensitive to parameter settings or assumptions on the number of people in the building than the microscopic simulator. On the other hand, the microscopic simulator is more flexible and has a greater modelling power. For every real-world application, one should question whether the increase in modelling detail offsets the increase in computational complexity and sensitivity to parameter values and assumptions.

Chapter 6

Conclusion

The study of pedestrian walking behaviour and crowd dynamics is interesting for a variety of reasons. The design of safe and comfortable infrastructure for pedestrian traffic is increasingly important in modern cities because of growing traffic congestions and environmental concerns. Additionally, the importance of crowd management has been revealed by major accidents in the past, such as the Innovation fire in Brussels on 22 May 1967, where 251 people were killed; the Heysel Stadium disaster on 29 May 1985, where 39 people were killed; and the crowd disaster at the 2010 Love Parade, where 21 people died from suffocation.

In Chapter 2, we have presented a review of the use of optimisation models for pedestrian evacuation and design problems. The articles are classified according to the problem type that is studied, the level of model realism, and the modelling or solution technique. To substantiate the classification criteria and to provide a background for the reader, relevant empirical research and descriptive models (e.g., social-force and cellular automata models) are discussed. We concluded that most of the recent models explicitly include pedestrian dynamics, specifically congestion, but more attention should be given to calibration and implementation of the proposed models. Furthermore, optimisation models could benefit from including some of the modelling techniques used in descriptive models.

Chapter 2 also showed that most optimisation models have focused on different types of evacuation problems, such as the evacuation of a room, a building, or an aircraft. Only a few models consider design problems (e.g., determining the optimal positions and size of emergency exits, positioning of obstacles in front of bottlenecks to reduce clogging) or crowd management under normal conditions. To the best of our knowledge, no models consider the impact of scheduling decisions in a timetable on people flows between consecutive events or on the evacuation process in the event of an emergency. This thesis has aimed to fill the gap. We have presented two different models to optimise the people flows in a timetable.

Chapter 3 has presented a two-stage integer programming approach for building a university course timetable that aims at minimising the travel times between lectures in consecutive timeslots and the evacuation times in the event of an emergency. The first stage minimises the violation of the teacher and educational preferences by assigning lectures to timeslots and rooms. The second stage reassigns classrooms to lectures of the timetable of the first stage and minimises the travel or egress times. The student flows are modelled using a network model, which is a macroscopic model where the building is represented as a graph and people 'flow' through the arcs from node to node. The network model takes congestion into account by assuming density-dependent travel times through the arcs, based on the fundamental diagram of pedestrian walking behaviour. The conceptual model is applied to the dataset of the Faculty of Economics and Business of the KU Leuven Campus Brussels and is tested and validated with 7 adapted instances from the literature. In contrast to a monolithic model, the two-stage model consistently succeeds in finding quality feasible solutions with significantly reduced travel and egress times. However, the two-stage MIP approach is not able to solve the large KU Leuven instance. Moreover, due to its hierarchical nature, the two-stage approach can only find a single endpoint of the Pareto-front between the optimisation of the scheduling preferences and the minimisation of the travel or egress times. To deal with these problems, a metaheuristic approach is developed based on a synchronous parallel simulated annealing implemenation. Results show that parallelisation improves the efficiency of the heuristic significantly and that it is able to improve upon the solutions found by the two-stage MIP approach.

In contrast to Chapter 3 that focuses on the university course timetabling problem, Chapter 4 focuses on a generic timetabling problem, to arrive at a generic, flexible model for timetabling problems with people flows. The microscopic Menge simulator developed by Curtis et al. (2016) is implemented in a surrogate-based tabu search heuristic to iteratively find solutions with improved people flows. Both the evacuation time in each timeslot as well as the travel time between events in each pair of consecutive timeslots can be included in the objective function. However, to reduce the complexity and to keep the model generic, only the assignment of events to rooms is considered. The assignment of events to timeslots is considered fixed. Instead of evaluating every candidate solution with the computationally expensive Menge simulator, a surrogate model is used to filter good candidate solutions from bad candidate solutions. Eight combinations of machine learning technique and kernel type are tested on a training set as well as during algorithm run. Extensive computational tests show that the model can find timetables with significantly reduced people flows. Finally, we show an application of the model to the timetabling problem of KU Leuven Campus Brussel for the second semester of academic year 2018-2019. For this purpose, the model has been implemented in a scheduling tool with graphical user interface

(GUI). We believe the GUI can help to show people involved in the planning process how the algorithm works and persuade them of its value.

In Chapter 5, we have compared the results of the two types of pedestrian models in greater detail. The robustness of the models with respect to the choice of parameter values is investigated. The results show that the microscopic model has greater modelling power than the network model, but is also much more sensitive to changes in parameter values and assumptions. For every specific application of the algorithms to a real-world problem, the question is whether the increase in modelling realism offsets the difficulty of calibration and the increase in computational complexity. If the parameter values and assumptions used in the pedestrian models do not reflect real-world behaviour then there is no value in using a more complex microscopic simulator. Ideally, parameter values and behavioural assumptions used in the model should be based on actual observations related to the problem at hand. However, such endeavour is outside of the scope of this thesis. In this thesis, we have focused on the computational properties of the two types of models and have developed different heuristic optimisation techniques that can be used to build timetables with improved travel times between events in consecutive timeslots and egress times in the event of an emergency.

Continuum models (see Section 2.4.1) are a type of models that can provide a middle ground between network models and microscopic simulation models. These models offer greater detail than the network model, but still describe the crowd as a whole using average quantities such as the speed and density at a given time and location. As such, they are less sensitive to the choice of modelling assumptions and parameter values than the microscopic models. One example is the model of Hughes (2002). Many microscopic models usually model the trajectories of N pedestrians with a system of N ordinary differential equations that link the position of each pedestrian as a function of time with his or her walking speed as a function of time. By contrast, Hughes (2002) models crowds with a single partial differential equation based on the conservation of flow, which states that the decrease in density in a given location over a specific time should equal the net outflow of people from that location over that specific time. Continuum and microscopic simulation models both need to be solved by numerical techniques, such as the well-known Runge-Kutta method. We could thus replace the Menge simulator in our surrogate-based tabu search heuristic with a continuum model. A valuable future research direction can be to compare both pedestrian models used in this thesis with a continuum model.

Appendix A

The timetable instances

Table A.1: The timetable instances.

|E| refers to the number of events (or lectures) in the instance, |T| the number of timeslots, and |R| the number of available rooms.

Instance	|E|	|T|	|R|	Details	Chapters
FEB2012	396	30	56	Real-world data	3
KUL2018	740	30	52	Real-world data	4
comp01	160	30	7	Real-world data from literature	3
comp02	283	25	16	Real-world data from literature	3
comp03	251	25	16	Real-world data from literature	3
comp04	286	25	18	Real-world data from literature	3
comp05	152	36	9	Real-world data from literature	3
comp06	361	25	18	Real-world data from literature	3
comp07	434	25	20	Real-world data from literature	3
ML	21	3	8	Randomly generated	3
T-8-1	24	5	8	Randomly generated	4
T-8-2	26	5	8	Randomly generated	4
T-8-3	24	5	8	Randomly generated	4
T-8-4	25	5	8	Randomly generated	4

Continued on next page

Table A.1 – continued from previous page

Instance	|E|	|T|	|R|	Details	Chapters
T-8-5	32	5	8	Randomly generated	4
T-16-1	57	5	16	Randomly generated	4
T-16-2	48	5	16	Randomly generated	4
T-16-3	48	5	16	Randomly generated	4
T-16-4	42	5	16	Randomly generated	4
T-16-5	44	5	16	Randomly generated	4
ES1	4	1	8	Randomly generated	4, 5
ES2	5	1	8	Randomly generated	4, 5
ES3	6	1	8	Randomly generated	4, 5
ES4	7	1	8	Randomly generated	4, 5
ES5	8	1	8	Randomly generated	4, 5

Appendix B

The building layouts

Figure B.1: Building B-8-1.

A building consisting of eight rooms divided over two floors. The blue-striped area indicates stairs going down and the red-striped area indicates stairs going up.

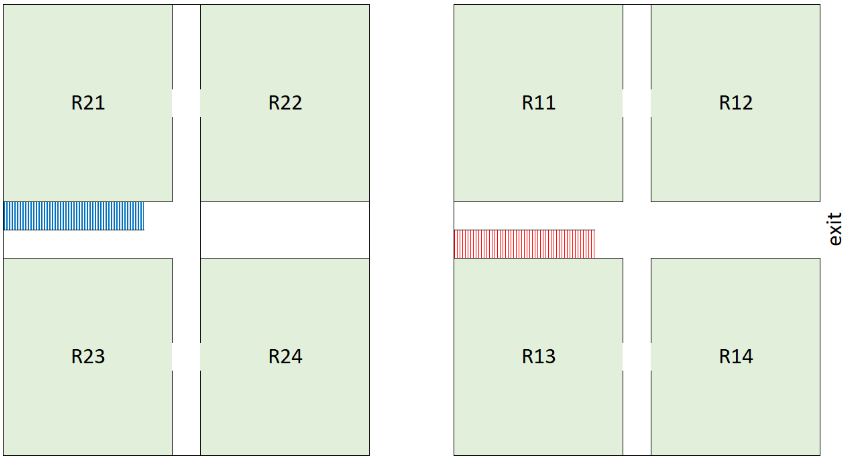


Figure B.2: The network model representation for building B-8-1.

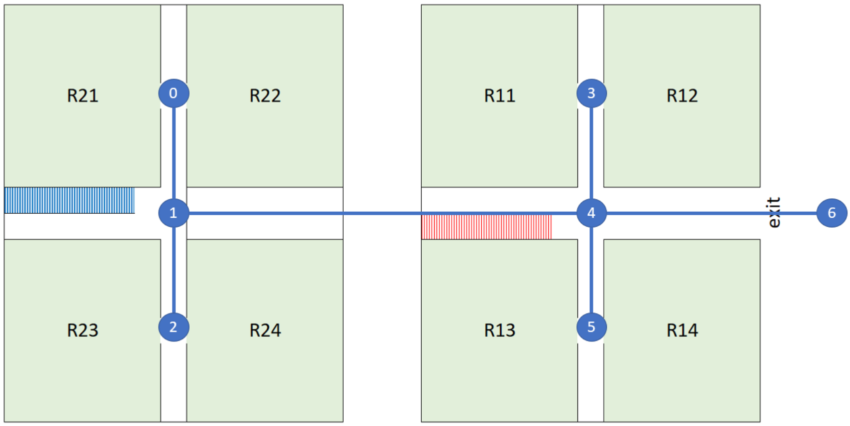


Figure B.3: Building B-8-2.

A building consisting of eight rooms divided over four floors. The blue-striped areas indicate stairs going down and the red-striped areas indicate stairs going up.

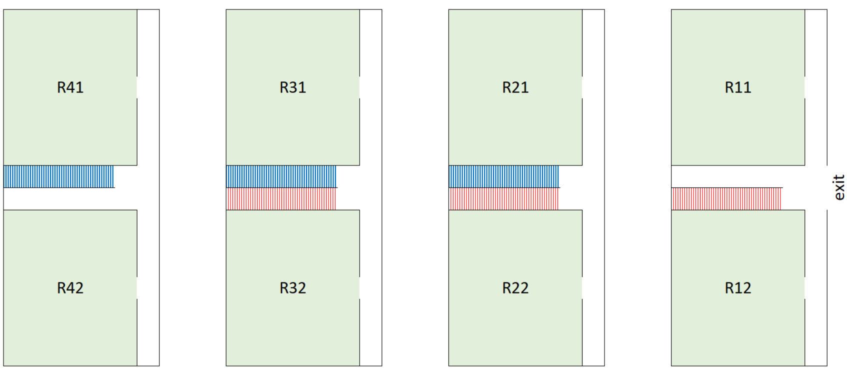


Figure B.4: The network model representation for building B-8-2.

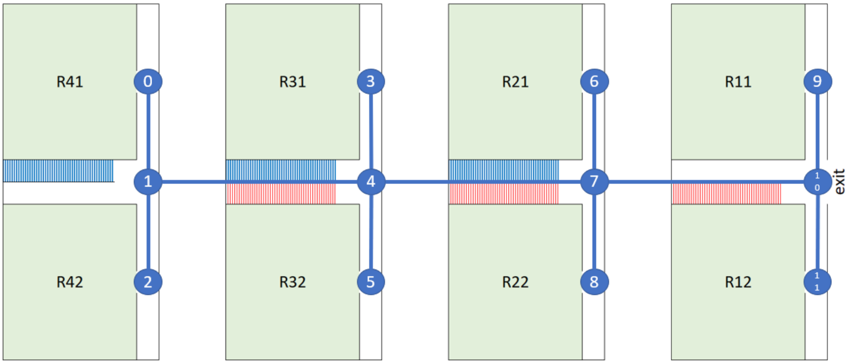


Figure B.5: Building B-16-1.

A building consisting of sixteen rooms divided over four floors and two stairwells. The blue-striped areas indicate stairs going down and the red-striped areas indicate stairs going up.

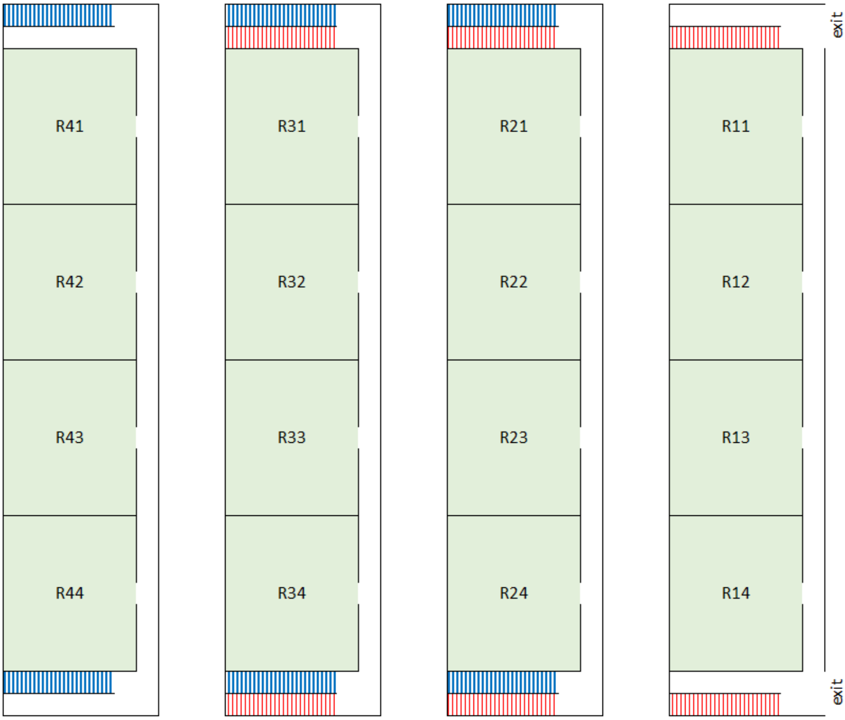


Figure B.6: The network model representation for building B-16-1.

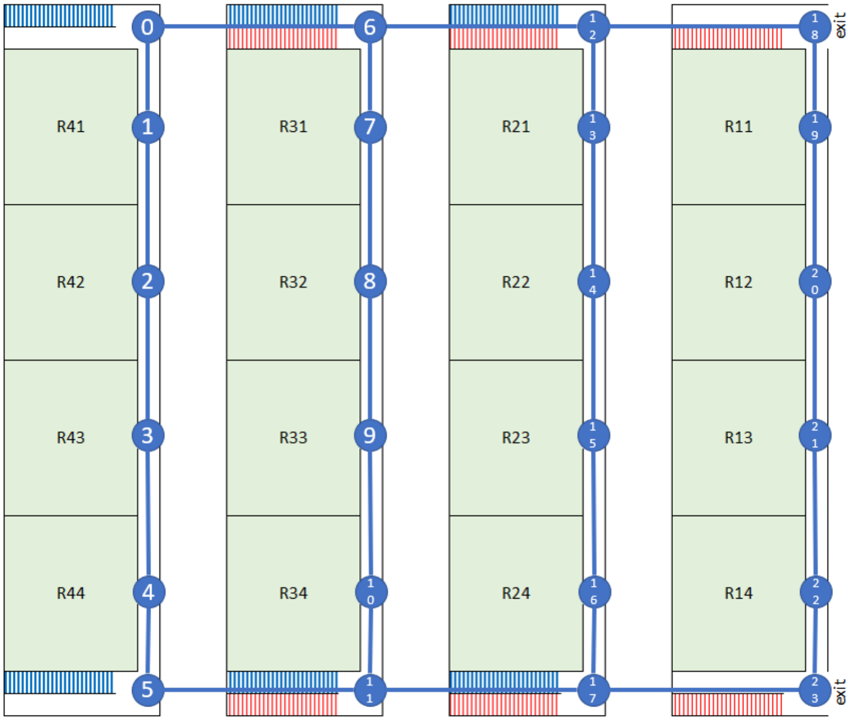


Figure B.7: Building B-16-2.

A building consisting of sixteen rooms divided over four floors and three stairwells. The blue-striped areas indicate stairs going down and the red-striped areas indicate stairs going up.

Figure B.8: The network model representation for building B-16-2.

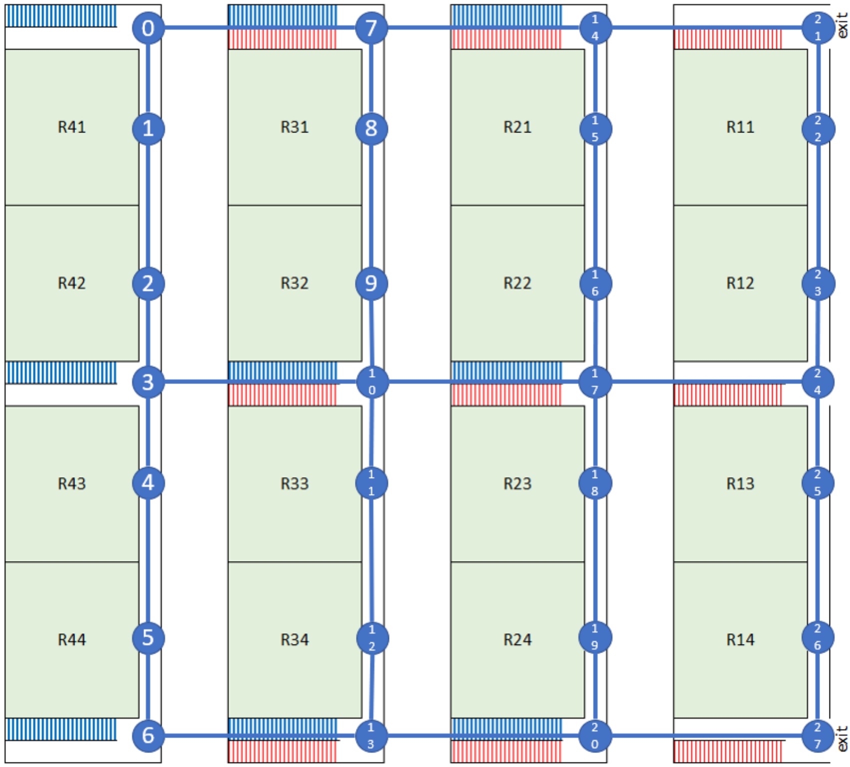


Figure B.9: Building B-20.

A building consisting of eight rooms divided over four floors and three stairwells. The blue-striped areas indicate stairs going down and the red-striped areas indicate stairs going up.

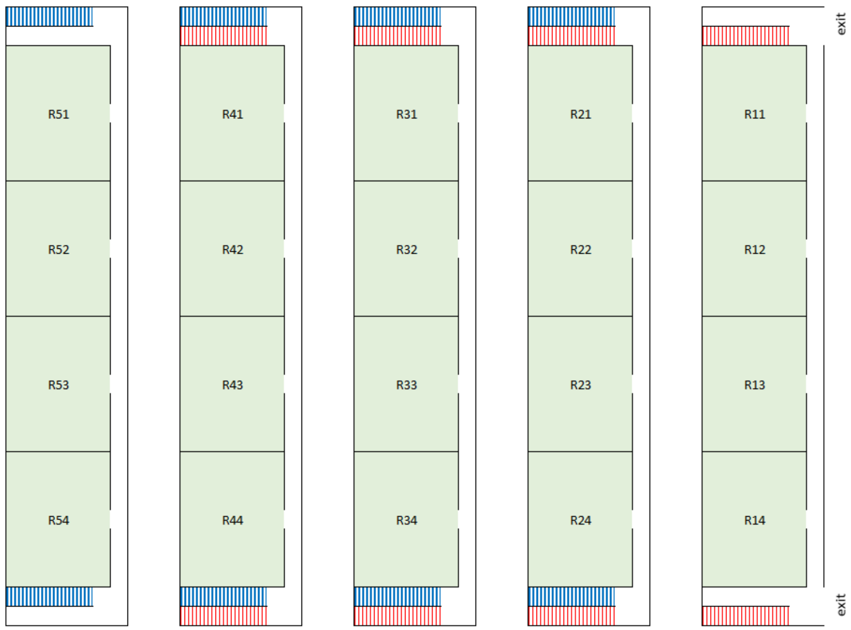


Figure B.10: The network model representation for building B-20.

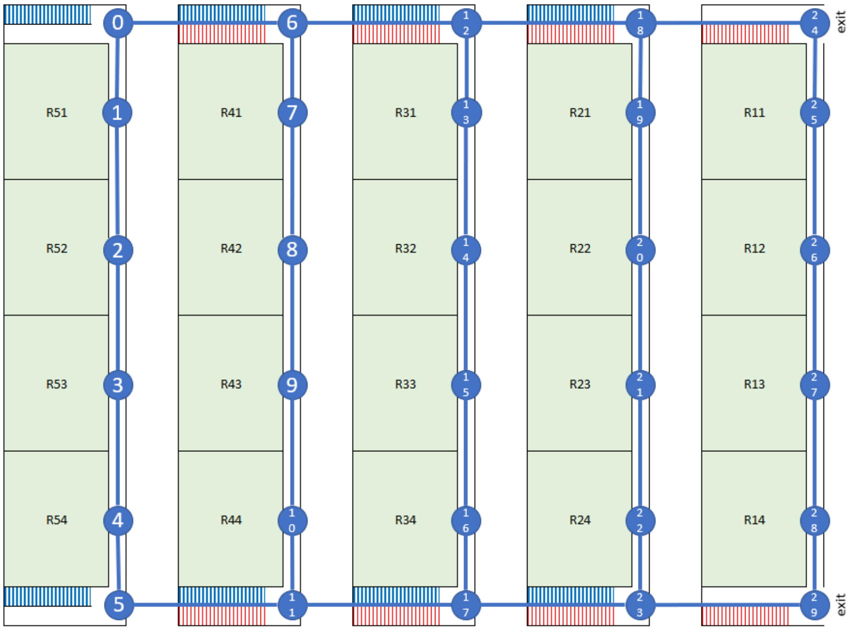


Figure B.11: The building of KU Leuven Campus Brussels.

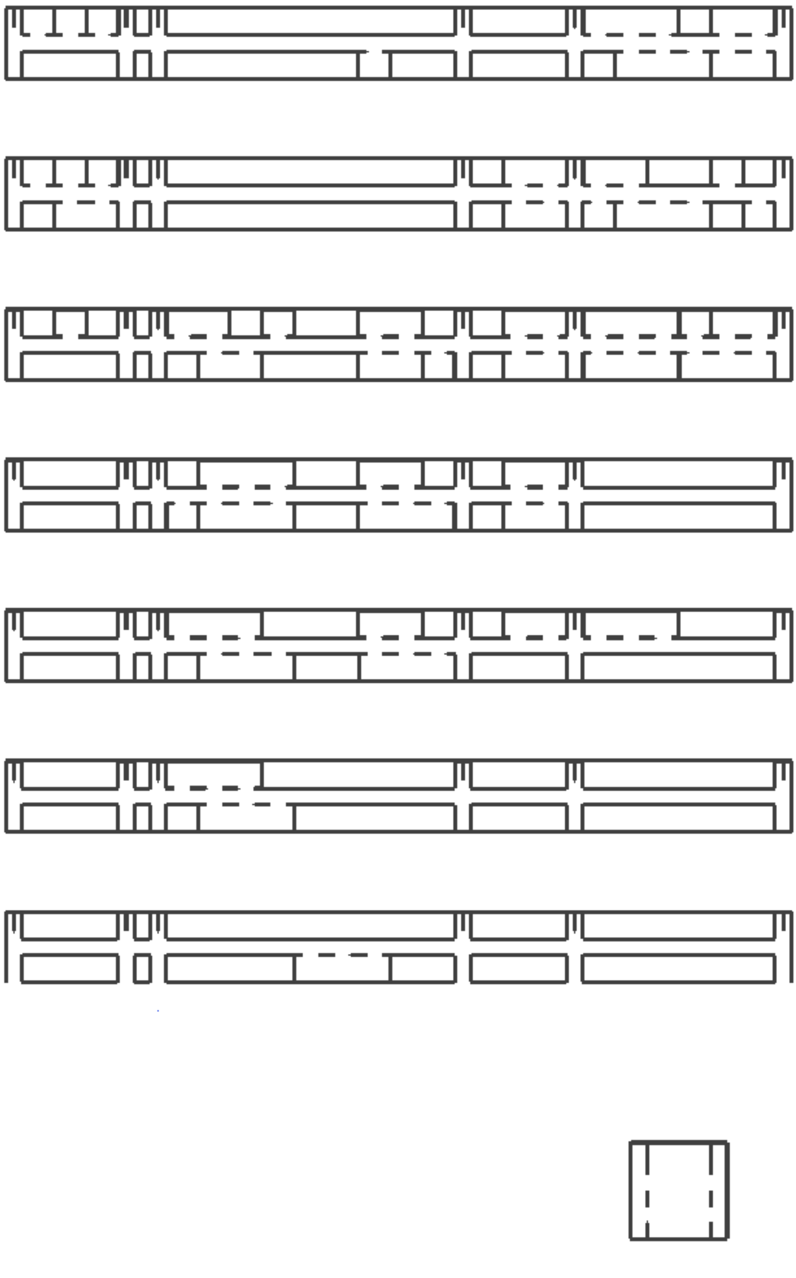
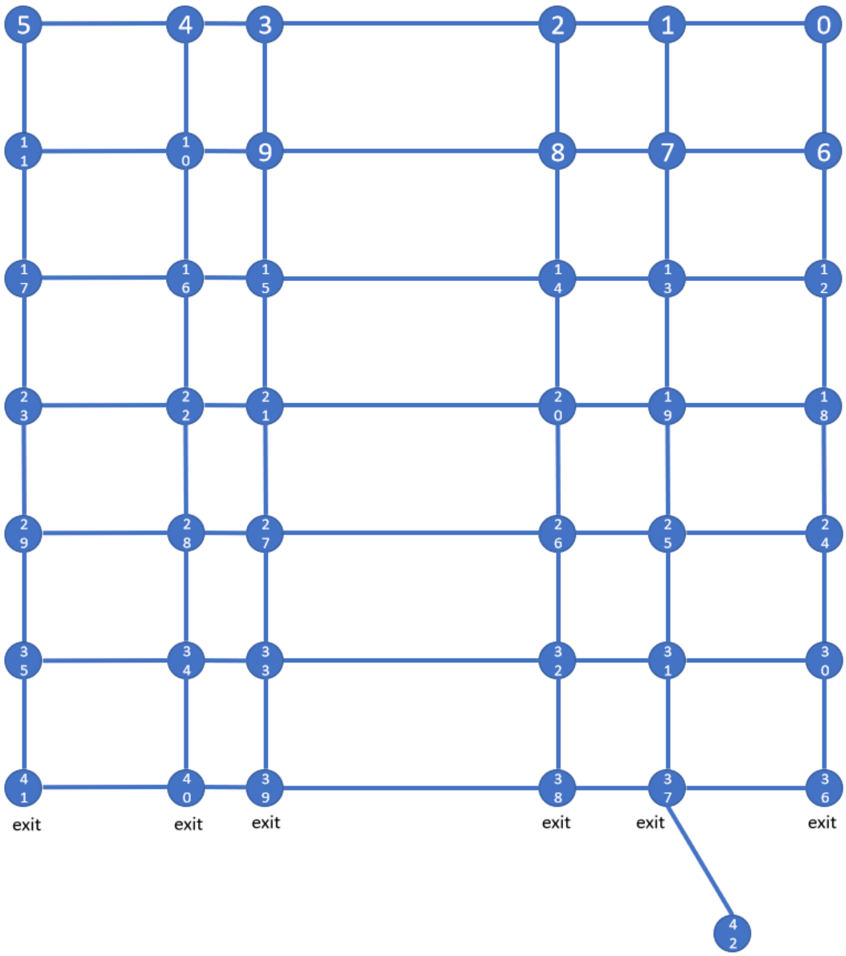


Figure B.12: The network model representation for the building of KU Leuven Campus Brussels.



List of Publications

Articles in internationally reviewed journals

Vermuyten, H., Namorado Rosa, J., Marques, I., Beliën, J., Barbosa-Póvoa, A. (2018). Integrated staff scheduling at a medical emergency service: An optimisation approach. *Expert Systems With Applications, 112*, 62-76. doi:10.1016/j.eswa.2018.06.017.

Vermuyten, H., Beliën, J., De Boeck, L., Reniers, G., Wauters, T. (2016). A review of optimisation models for pedestrian evacuation and design problems. *Safety Science, 87*, 167-178. doi:10.1016/j.ssci.2016.04.001.

Vermuyten, H., Lemmens, S., Marques, I., Beliën, J. (2016). Developing compact course timetables with optimized student flows. *European Journal of Operational Research, 251*(2), 651-661. doi:10.1016/j.ejor.2015.11.028.

Papers at international conferences and symposia, published in full in proceedings

Vermuyten H., Namorado Rosa J., Marques I., Beliën J., Barbosa-Póvoa A. (2019). A Column Generation-Based Diving Heuristic for Staff Scheduling at an Emergency Medical Service. In: Alves M., Almeida J., Oliveira J., Pinto A. (eds) Operational Research. IO 2018. Springer Proceedings in Mathematics & Statistics, vol 278. Springer, Cham. doi:10.1007/978-3-030-10731-4_16.

Meeting abstracts, presented at international conferences and symposia, published or not published in proceedings or journals

Vermuyten, H., Beliën, J., De Boeck, L., Wauters, T. (2019). A surrogate-based tabu search heuristic for room scheduling to improve people flows. Annual conference of the Belgian Operational Research Society (ORBEL), Hasselt, Belgium, 7-8 February 2019.

Vermuyten, H., Beliën, J., De Boeck, L., Wauters, T. (2018). A Surrogate-based Tabu Search Heuristic to Optimise the People Flows in a Timetable. INFORMS Annual Meeting, Phoenix, Arizona, USA, 4-7 November 2018.

Vermuyten H., Namorado Rosa, J., Marques, I., Beliën, J., Barbosa-Póvoa, A. (2018). Staff Scheduling at a Medical Emergency Service: a case study at Instituto Nacional de Emergência Médica. Annual conference of the Belgian Operational Research Society (ORBEL), Liège, Belgium, 1-2 February 2018.

Vermuyten H., Marques I., Barbosa-Póvoa A., Namorado Rosa J. (2017). Integrated Staff Scheduling at a Medical Emergency Service: An Optimization Approach. INFORMS Annual Meeting, Houston, Texas, USA, 22-25 October 2017.

Vermuyten H., Namorado Rosa J., Marques I., Barbosa-Póvoa A. (2017). Optimizing staff scheduling: a case at INEM. The XVIII Congress of the Portuguese Association of Operational Research (IO2017). Valença, Portugal, 28-30 June 2017.

Vermuyten H., Beliën J., Wauters T. (2017). The impact of timetabling on the efficient evacuation of a building. Annual conference of the Belgian Operational Research Society (ORBEL), Brussels, Belgium, 2-3 February 2017.

Vermuyten H., Beliën J., Wauters T. (2016). The Impact of Timetabling on the Efficient Evacuation of a Facility in the Event of an Emergency. European Conference On Operational Research (EURO). Poznań, Poland, 3-6 July 2016.

Vermuyten H., Lemmens S., Marques I., Beliën J. (2016). Developing compact course timetables with optimized student flows. Annual conference of the Belgian Operational Research Society (ORBEL). Louvain-la-Neuve, Belgium, 28-29 January 2016.

Vermuyten H. (2015). An Extension of the Stochastic Dynamic Lot-Size Model of Vargas to a Model with Uncertain Production. INFORMS Annual Meeting, Philadelphia, Pennsylvania, USA, 1-4 November 2015.

Vermuyten H., Beliën J., Marques I., Lemmens S. (2015). A two-stage model for optimizing the student flow of a university course timetabling problem. European Conference On Operational Research (EURO). Glasgow, UK, 12-15 July 2015.

List of Figures

List of Tables

Bibliography

Abdelghany, A., Abdelghany, K., Mahmassani, H., and Alhalabi, W. (2014). Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities. *European Journal of Operational Research*, 237(3):1105–1118. doi:10.1016/j.ejor.2014.02.054.

Al-Yakoob, S. M. and Sherali, H. D. (2007). A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations. *European Journal of Operational Research*, 180(3):1028–1044. doi:10.1016/j.ejor.2006.04.035.

Al-Yakoob, S. M., Sherali, H. D., and Al-Jazzaf, M. (2010). A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science*, 7(1):19–46. doi:10.1007/s10287-007-0066-8.

Aladag, C., Hocaoglu, G., and Basaran, M. (2009). The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem. *Expert Systems with Applications*, 36:12349–12356. doi:10.1016/j.eswa.2009.04.051.

Antonini, G., Bierlaire, M., and Weber, M. (2006). Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological*, 40(8):667–687. doi:10.1016/j.trb.2005.09.006.

Appert-Rolland, C., Degond, P., and Motsch, S. (2011). Two-way multi-lane traffic model for pedestrians in corridors. *Networks and Heterogeneous Media*, 6:351–381. doi:10.3934/nhm.2011.6.351.

Babaei, H., Karimpour, J., and Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86(C):43–59. doi:10.1016/j.cie.2014.11.010.

Badoni, R., Gupta, D., and Mishra, P. (2014). A new hybrid algorithm for university course timetabling problem using events based on groupings of students. *Computers & Industrial Engineering*, 78:12–25. doi:10.1016/j.cie.2014.09.020.

Badri, M. a. (1996). A two-stage multiobjective scheduling model for [faculty-course-time] assignments. *European Journal of Operational Research*, 94(1):16–28. doi:10.1016/0377-2217(95)00204-9.

Bakuli, D. L. and Smith, J. M. (1996). Resource allocation in state-dependent emergency evacuation networks. *European Journal of Operational Research*, 89(3):543–555. doi:10.1016/0377-2217(94)00230-4.

Bellio, R., Di Gaspero, L., and Schaerf, A. (2012). Design and statistical analysis of a hybrid local search algorithm for course timetabling. *Journal of Scheduling*, pages 1–13. doi:10.1007/s10951-011-0224-2.

Bellomo, N., Piccoli, B., and Tosin, A. (2012). Modeling crowd dynamics from a complex system viewpoint. *Mathematical Models and Methods in Applied Sciences*, 22(2):1230004. doi:10.1142/S0218202512300049.

Berseth, G., Usman, M., Haworth, B., Kapadia, M., and Faloutsos, P. (2015). Environment optimization for crowd evacuation. *Computer Animation and Virtual Worlds*, 26(3-4):377–386. doi:10.1002/cav.1652.

Blue, V. J. and Adler, J. L. (2001). Cellular automata microsimulation for modeling bi-directional pedestrian walkways. *Transportation Research Part B: Methodological*, 35(3):293–312. doi:10.1016/S0191-2615(99)00052-1.

Bonutti, A., De Cesco, F., Di Gaspero, L., and Schaerf, A. (2012). Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research*, 194(1):59–70. doi:10.1007/s10479-010-0707-0.

Borrmann, A., Kneidl, A., Köster, G., Ruzika, S., and Thiemann, M. (2012). Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models. *Safety Science*, 50(8):1695–1703. doi:10.1016/j.ssci.2011.12.021.

Boyce, D., Janson, B., and Eash, R. (1981). The effect on equilibrium trip assignment of different link congestion functions. *Transportation Research Part A: General*, 15(3):223–232. doi:10.1016/0191-2607(81)90003-0.

Brown, J. (1979). The knapsack sharing problem. *Operations Research*, 27(2):341–355. doi:10.1287/opre.27.2.341.

Burke, E., MacCarthy, B., Petrovic, S., and Qu, R. (2006a). Multiple-retrieval case based reasoning for course timetabling problems. *Journal of the Operational Research Society*, 57:148–162. doi:10.1057/palgrave.jors.2601970.

Burke, E., Petrovic, S., and Qu, R. (2006b). Case based heuristic selection for timetabling problems. *Journal of Scheduling*, 9:115–132. doi:10.1007/s10951-006-6775-y.

Burke, E. K., Mareček, J., Parkes, A. J., and Rudová, H. (2010). Decomposition, reformulation, and diving in university course timetabling. *Computers & Operations Research*, 37(3):582–597. doi:10.1016/j.cor.2009.02.023.

Burke, E. K. and Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266–280. doi:10.1016/S0377-2217(02)00069-3.

Carrasco, M. and Pato, M. (2004). A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. *European Journal of Operational Research*, 153:65–79. doi:10.1016/S0377-2217(03)00099-7.

Carter, M. and Laporte, G. (1996). Recent developments in practical examination timetabling. In Burke, E. and Ross, P., editors, *Practice and theory of automated timetabling I. PATAT 1995. Lecture Notes in Computer Science, vol. 1153*, pages 1–21. Springer: Berlin, Heidelberg. doi:10/bthb6p.

Carter, M. and Laporte, G. (1998). Recent developments in practical course timetabling. In Burke, E. and Carter, M., editors, *Practice and theory of automated timetabling II. PATAT 1997. Lecture Notes in Computer Science, vol. 1408*, pages 3–19. Springer: Berlin, Heidelberg. doi:10.1007/BFb0055878.

Cepolina, E. M. (2005). A methodology for defining building evacuation routes. *Civil Engineering and Environmental Systems*, 22(1):29–47. doi:10.1080/10286600500049946.

Cepolina, E. M. (2009). Phased evacuation: An optimisation model which takes into account the capacity drop phenomenon in pedestrian flows. *Fire Safety Journal*, 44(4):532–544. doi:10.1016/j.firesaf.2008.11.002.

Chalmet, L. G., Francis, R. L., and Saunders, P. B. (1982). Network models for building evacuation. *Management Science*, 28(1):86–105. doi:10.1287/mnsc.28.1.86.

Chen, P. H. and Feng, F. (2009). A fast flow control algorithm for real-time emergency evacuation in large indoor areas. *Fire Safety Journal*, 44(5):732–740. doi:10.1016/j.firesaf.2009.02.005.

Chiarandini, M., Socha, K., Biarattari, M., and Rossi-Doria, O. (2006). An effective hybrid algorithm for university course timetabling. *Journal of Scheduling*, 9:403–432. doi:10.1007/s10951-006-8495-8.

Choi, W., Hamacher, H., and Tufekci, S. (1988). Modeling of building evacuation problems by network flows with side constraints. *European Journal of Operational Research*, 35(1):98–110. doi:10.1016/0377-2217(88)90382-7.

Chooramun, N., Lawrence, P. J., and Galea, E. R. (2012). An agent based evacuation model utilising hybrid space discretisation. *Safety Science*, 50(8):1685–1694. doi:10.1016/j.ssci.2011.12.022.

Chraibi, M., Seyfried, A., and Schadschneider, A. (2010). Generalized centrifugal-force model for pedestrian dynamics. *Physical Review E*, 82:046111. doi:10.1103/PhysRevE.82.046111.

Crooks, A., Castle, C., and Batty, M. (2008). Key challenges in agent-based modelling for geo-spatial simulation. *Computers, Environment and Urban Systems*, 32(6):417–430. doi:10.1016/j.compenvurbsys.2008.09.004.

Curtis, S. (2014). *Pedestrian Velocity Obstacles: Pedestrian Simulation Through Reasoning in Velocity Space*. PhD thesis, University of North Carolina at Chapel Hill.

Curtis, S., Best, A., and Manocha, D. (2016). Menge: A modular framework for simulating crowd movement. *Collective Dynamics*, 1:1–40. doi:10.17815/CD.2016.1.

Daskalaki, S., Birbas, T., and Housos, E. (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1):117–135. doi:10.1016/S0377-2217(03)00103-6.

De Causmaecker, P., Demeester, P., and Vanden Berghe, G. (2009). A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research*, 195:307–318. doi:10.1016/j.ejor.2008.01.043.

Deng, K., Chen, W., Mehta, P. G., and Meyn, S. P. (2008). Resource pooling for optimal evacuation of a large building. *Proceedings of the IEEE Conference on Decision and Control*, 47:5565–5570. doi:10.1109/CDC.2008.4739428.

Dimopoulou, M. and Miliotis, P. (2001). Implementation of a university course and examination timetabling system. *European Journal of Operational Research*, 130(1):202–213. doi:10.1016/S0377-2217(00)00052-7.

Ding, A. W. (2011). Implementing real-time grouping for fast egress in emergency. *Safety Science*, 49(10):1404–1411. doi:10.1016/j.ssci.2011.06.006.

Doheny, J. G. and Fraser, J. L. (1996). MOBEDIC - A decision modelling tool for emergency situations. *Expert Systems with Applications*, 10(1):17–27. doi:10.1016/0957-4174(95)00030-5.

Dorneles, A., de Araújo, O., and Buriol, L. (2014). A fix-and-optimize heuristic for the high school timetabling problem. *Computers & Operations Research*, 52(A):29–38. doi:10.1016/j.cor.2014.06.023.

Drager, K., Løvås, G., and Wiklund, J. (1992). Evacsim: a comprehensive evacuation simulation tool. In *Sullivan, J. (Ed.), Proceedings of the 1992 International Emergency Management and Engineering Conference*, pages 101–108.

Duives, D. C., Daamen, W., and Hoogendoorn, S. P. (2013). State-of-the-art crowd motion simulation models. *Transportation Research Part C: Emerging Technologies*, 37(1):193–209. doi:10.1016/j.trc.2013.02.005.

Fahy, R. F. (1994). EXIT 89 - An evacuation model for high rise buildings - model description and example applications. doi:10.3801/IAFSS.FSS.4-657.

Ferdoushi, T., Das, P., and Akhand, M. (2014). Highly constrained university course scheduling using modified hybrid particle swarm optimization. In *2013 International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–5. doi:10.1109/EICT.2014.6777887.

Ferreiro, A., García, J., López-Salas, J., and Vázquez, C. (2013). An efficient implementation of parallel simulated annealing in GPUs. *Journal of Global Optimization*, 57(3):863–890. doi:10.1007/s10898-012-9979-z.

Flötteröd, G. and Lämmel, G. (2015). Bidirectional pedestrian fundamental diagram. *Transportation Research Part B: Methodological*, 71(2015):194–212. doi:10.1016/j.trb.2014.11.001.

Fonseca, G., Santos, H., Toffolo, T., Brito, S., and Brazil, M. (2016a). GOAL solver: a hybrid local search based solver for high school timetabling. *Annals of Operations Research*, 239:77–97. doi:10.1007/s10479-014-1685-4.

Fonseca, G. H. G., Santos, H. G., and Carrano, E. G. (2016b). Late acceptance hill-climbing for high school timetabling. *Journal of Scheduling*, 19(4):453–465. doi:10.1007/s10951-015-0458-5.

Forrester, A. and Keane, A. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79. doi:10.1016/j.paerosci.2008.11.001.

Fruin, J. (1971). *Pedestrian Planning and Design.* Metropolitan Association of Urban Designers and Environmental Planners, New York.

Fu, M. (2001). Simulation optimization. In Peters, B., Smith, J., Medeiros, D., and Rohrer, M., editors, *Proceedings of the 2001 Winter Simulation Conference*, pages 53–61. doi:10.1109/WSC.2001.977246.

Fu, M., April, J., and Glover, F. (2005). Simulation optimization: a review, new developments, and applications. In Kuhl, M., Steiger, N., Armstrong, F., and Joines, J., editors, *Proceedings of the 2005 Winter Simulation Conference.* doi:10.1109/WSC.2005.1574242.

Galea, E. and Perez Galparsoro, J. (1994). A computer-based simulation model for the prediction of evacuation from mass-transport vehicles. *Fire Safety Journal*, 22:341–366. doi:10.1016/0379-7112(94)90040-X.

Galiza, R. and Ferreira, L. (2013). A methodology for determining equivalent factors in heterogeneous pedestrian flows. *Computers, Environment and Urban Systems*, 39:162–171. doi:10.1016/j.compenvurbsys.2012.08.003.

Geiger, M. (2012). Applying the threshold accepting metaheuristic to curriculum based course timetabling. *Annals of Operations Research*, 194:189–202. doi:10.1007/s10479-010-0703-4.

Georgoudas, I. G., Sirakoulis, G. C., and Andreadis, I. T. (2010). An anticipative crowd management system preventing clogging in exits during pedestrian evacuation processes. *IEEE Systems Journal*, 5(1):129–141. doi:10.1109/JSYST.2010.2090400.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549. doi:10.1016/0305-0548(86)90048-1.

Goldsman, D. and Nelson, B. (1998). Statistical screening, selection, and multiple comparison procedures in computer simulation. In Medeiros, D., Watson, E., Carson, J., and Manivannan, M., editors, *Proceedings of the 1998 Winter Simulation Conference*, pages 159–166. doi:10.1109/WSC.1998.744911.

Guéret, C., Jussien, N., Boizumault, P., and Prins, C. (1996). Building university timetables using constraint logic programming. In Burke, E. and Ross, P., editors, *Practice and theory of automated timetabling i*, pages 130–145. Springer, Berlin. doi:10/djwkxb.

Gunawan, A. and Kien Ming, N. (2012). A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem. *Computers & Operations Research*, 39:3074–3088. doi:10.1016/j.cor.2012.03.011.

Guo, R. Y., Huang, H. J., and Wong, S. C. (2011). Collection, spillback, and dissipation in pedestrian evacuation: A network-based method. *Transportation Research Part B: Methodological*, 45(3):490–506. doi:10.1016/j.trb.2010.09.009.

Guo, R. Y., Huang, H. J., and Wong, S. C. (2012). Route choice in pedestrian evacuation under conditions of good and zero visibility: Experimental and simulation results. *Transportation Research Part B: Methodological*, 46(6):669–686. doi:10.1016/j.trb.2012.01.002.

Gwynne, S., Galea, E., Owen, M., Lawrence, P., 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):741–749. doi:10.1016/S0360-1323(98)00057-2.

Hamacher, H. W. and Tjandra, S. A. (2002). Mathematical modelling of evacuation problems: a state of the art. In *Pedestrian and Evacuation Dynamics*, pages 227–266.

Hankin, B. and Wright, R. (1958). Passenger flow in subways. *OR*, 9(2):81–88. doi:10.2307/3006732.

Hänseler, F. S., Bierlaire, M., Farooq, B., and Mühlematter, T. (2014). A macroscopic loading model for time-varying pedestrian flows in public walking areas. *Transportation Research Part B: Methodological*, 69:60–80. doi:10.1016/j.trb.2014.08.003.

Hao, J. and Benlic, U. (2011). Lower bounds for the ITC-2007 curriculum-based course timetabling problem. *European Journal of Operational Research*, 212:464–472. doi:10.1016/j.ejor.2011.02.019.

He, G.-Q., Yang, Y., Chen, Z.-H., Gu, C.-H., and Pan, Z.-G. (2013). A review of behavior mechanisms and crowd evacuation animation in emergency exercises. *Journal of Zhejiang University - SCIENCE C (Computers & Electronics)*, 14(7):477–485. doi:10.1631/jzus.CIDE1301.

Helbing, D. (1991). A mathematical model for the behavior of pedestrians. *Behavioral Science*, 36(4):298–310. doi:10.1002/bs.3830360405.

Helbing, D., Buzna, L., Johansson, A., and Werner, T. (2005). Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Science*, 39(1):1–24. doi:10.1287/trsc.1040.0108.

Helbing, D., Farkas, I., and Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407:487–490. doi:10.1038/35035023.

Helbing, D. and Johansson, A. (2010). Pedestrian, crowd and evacuation dynamics: experiments, simulations, and design solutions. In

Encyclopedia of complexity and System Science, volume 16, pages 6476–6495. doi:10.1007/978-3-642-04504-2.

Helbing, D., Johansson, A., and Al-Abideen, H. Z. (2007). Dynamics of crowd disasters: An empirical study. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 75(4):1–7. doi:10.1103/PhysRevE.75.046109.

Helbing, D. and Molnár, P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286. doi:10.1103/PhysRevE.51.4282.

Hertz, A. (1991). Tabu search for large scale timetabling problems. *European Journal of Operational Research*, 54(1):39–47. doi:10.1016/0377-2217(91)90321-L.

Hong, L. J. and Nelson, B. L. (2009). A brief introduction to optimization via simulation. In Rossetti, M., Hill, R., Johansson, B., Dunkin, A., and Ingalls, R., editors, *Proceedings of the 2009 Winter Simulation Conference*, pages 75–85. doi:10.1109/WSC.2009.5429321.

Hoogendoorn, S. and Bovy, P. H. L. (2003). Simulation of pedestrian flows by optimal control and differential games. *Optimal Control Applications and Methods*, 24(3):153–172. doi:10.1002/oca.727.

Hoogendoorn, S. P. and Bovy, P. H. L. (2001). Generic gas-kinetic traffic systems modeling with applications to vehicular traffic flow. *Transportation Research Part B: Methodological*, 35(4):317–336. doi:10.1016/S0191-2615(99)00053-3.

Hoogendoorn, S. P. and Bovy, P. H. L. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological*, 38(2):169–190. doi:10.1016/S0191-2615(03)00007-9.

Hoogendoorn, S. P. and Daamen, W. (2005). Pedestrian behavior at bottlenecks. *Transportation Science*, 39(2):147–159. doi:10.1287/trsc.1040.0102.

Hoppe, B. and Tardos, E. (1994). Polynomial time algorithms for some evacuation problems. In *Proceedings of the Fifth Annual SIAM Symposium on Discrete Algorithms*, pages 433–441.

Hoppe, B. and Tardos, E. (2000). The quickest transshipment problem. *Mathematics of Operations Research*, 25(1):36–62. doi:10.1287/moor.25.1.36.15211.

Huang, L., Wong, S., Zhang, M., Shu, C.-W., and Lam, W. H. (2009). Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm. *Transportation Research Part B: Methodological*, 43(1):127–141. doi:10.1016/j.trb.2008.06.003.

Hughes, R. L. (2002). A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological*, 36(6):507–535. doi:10.1016/S0191-2615(01)00015-7.

Jalali, H. and Van Nieuwenhuyse, I. (2015). Simulation optimization in inventory replenishment: a classification. *IIE Transactions*, 47(11):1217–1235. doi:10.1080/0740817X.2015.1019162.

Johansson, A. and Helbing, D. (2005). Pedestrian flow optimization with a genetic algorithm based on Boolean grids. In *Pedestrian and evacuation dynamics*, pages 267–272. doi:10/d35gfr.

Johansson, A., Helbing, D., Al-Abideen, H. Z., and Al-Bosta, S. (2008). From crowd dynamics to crowd safety: A video-based analysis. *Advances in Complex Systems*, 11:479–527. doi:10.1142/S0219525908001854.

Kady, R. A. (2012). The development of a movement-density relationship for people going on four in evacuation. *Safety Science*, 50(2):253–258. doi:10.1016/j.ssci.2011.08.058.

Kalakou, S. and Moura, F. (2014). Bridging the gap in planning indoor pedestrian facilities. *Transport Reviews: A Transnational Transdisciplinary Journal*, 34(4):474–500. doi:10.1080/01441647.2014.915441.

Kalender, M., Kheiri, A., Özcan, E., and Burke, E. (2012). A greedy gradient-simulated annealing hyperheuristic for a curriculum-based course timetabling problem. pages 1–8. doi:10.1109/UKCI.2012.6335754.

Kang, J., Jeong, I.-J., and Kwun, J.-B. (2015). Optimal facility-final exit assignment algorithm for building complex evacuation. *Computers & Industrial Engineering*, 85:169–176. doi:10.1016/j.cie.2015.03.012.

Karamouzas, I., Heil, P., van Beek, P., and Overmars, M. H. (2009). A predictive collision avoidance model for pedestrian simulation. In Egges, A., Geraerts, R., and Overmars, M., editors, *Motion in Games*, pages 41–52. Springer: Berlin Heidelberg. doi:10/cgkn2s.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220:671–680. doi:10.1126/science.220.4598.671.

Kisko, T. M. and Francis, R. L. (1985). EVACNET+: A computer program to determine optimal building evacuation plans. *Fire Safety Journal*, 9(2):211–220. doi:10.1016/0379-7112(85)90009-8.

Lachapelle, A. and Wolfram, M. T. (2011). On a mean field game approach modeling congestion and aversion in pedestrian crowds.

Transportation Research Part B: Methodological, 45(10):1572–1589. doi:10.1016/j.trb.2011.07.011.

Lam, J. H., Yuen, J. K., Lee, E. W., and Lee, R. Y. (2014). Experimental study on upward movement in a high-rise building. *Safety Science*, 70:397–405. doi:10.1016/j.ssci.2014.07.011.

Langston, P. a., Masling, R., and Asmar, B. N. (2006). Crowd dynamics discrete element multi-circle model. *Safety Science*, 44(5):395–417. doi:10.1016/j.ssci.2005.11.007.

Law, A. and Kelton, W. (1991). *Simulation Modeling & Analysis*. McGraw-Hill, Inc.

Lee, D., Kim, H., Park, J.-H., and Park, B.-J. (2003). The current status and future issues in human evacuation from ships. *Safety Science*, 41(10):861–876. doi:10.1016/S0925-7535(02)00046-2.

Lee, S.-Y. and Lee, K. (1996). Synchronous and Asynchronous Parallel Simulated Annealing with Multiple Markov Chains. In *IEEE Transactions on Parallel and Distributed Systems*, volume 7, pages 993–1008. doi:10.1109/71.539732.

Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30(1):167–190. doi:10.1007/s00291-007-0097-0.

Li, N. and Xu, Y. (2014). Evacuation modeling from the control perspective and corresponding sequential-based optimal evacuation guidance. *IEEE Transactions on Control Systems Technology*, 22(3):1094–1102. doi:10.1109/TCST.2013.2272558.

Lim, G. J., Rungta, M., and Baharnemati, M. R. (2015). Reliability analysis of evacuation routes under capacity uncertainty of road links. *IIE Transactions*, 47(1):50–63. doi:10.1080/0740817X.2014.905736.

Liu, X.-D., Song, W.-G., and Lv, W. (2014). Empirical data for pedestrian counterflow through bottlenecks in the channel. *Transportation Research Procedia*, 2:34–42. doi:10.1016/j.trpro.2014.09.006.

Løvås, G. G. (1994). Modeling and simulation of pedestrian traffic flow. *Transportation Research Part B: Methodological*, 28(6):429–443. doi:10.1016/0191-2615(94)90013-2.

Løvås, G. G. (1995). On performance measures for evacuation systems. *European Journal of Operational Research*, 85(2):352–367. doi:10.1016/0377-2217(94)00054-G.

Løvås, G. G. (1998). Models of wayfinding in emergency evacuations. *European Journal of Operational Research*, 105(3):371–389. doi:10.1016/S0377-2217(97)00084-2.

Lü, Z. and Hao, J. (2010). Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200:235–244. doi:10.1016/j.ejor.2008.12.007.

Ma, J., Song, W. G., Tian, W., Lo, S. M., and Liao, G. X. (2012). Experimental study on an ultra high-rise building evacuation in China. *Safety Science*, 50(8):1665–1674. doi:10.1016/j.ssci.2011.12.018.

McCollum, B. (2007). A perspective on bridging the gap between theory and practice in university timetabling. In Burke, E. and Rudová, H., editors, *Practice and theory of automated timetabling VI*, pages 3–23. Springer: Berlin. doi:10/cdmq34.

Mercy, A. (2012). Het opstellen van lessenroosters met het oog op het minimaliseren van studentenstromen. Master's thesis, Hogeschool-Universiteit Brussel.

MirHassani, S. and Habibi, F. (2013). Solution approaches to the course timetabling problem. *Artificial Intelligence Review*, pages 1–17. doi:10.1007/s10462-011-9262-6.

Mōri, M. and Tsukaguchi, H. (1987). A new method for evaluation of level of service in pedestrian facilities. *Transportation Research Part A: General*, 21(3):223–234. doi:10.1016/0191-2607(87)90016-1.

Moussaïd, M., Helbing, D., Garnier, S., Johansson, A., Combe, M., and Theraulaz, G. (2009). Experimental study of the behavioural mechanisms underlying self-organization in human crowds. In *Proceedings of The Royal Society B*, volume 276, pages 2755–2762. doi:10.1098/rspb.2009.0405.

Muhdi, R., Gwynne, S., and Davis, J. (2009). The incorporation and validation of empirical crawling data into the buildingEXODUS model. *Safety Science*, 47(1):97–104. doi:10.1016/j.ssci.2007.12.003.

National Fire Protection Association (2016). Fire statistics: Printable fact sheets. Retrieved from `http://www.nfpa.org/news-and-research/fire-statistics-and-reports/fire-statistics/printable-fact-sheets`.

Nelson, H. and MacLennan, H. (1995). Emergency movement. In *DiNinno, P. et al. (Eds.), SFPE Handbook of Fire Protection Engineering, second Ed. National Fire Protection Association, Quincy, Mass.*

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):1086–1094. doi:10.1016/j.tre.2010.04.001.

Oeding, D. (1963). Verkehrsbelastung und Dimensionierung von Gehwegen und anderen Anlagen des Fugängerverkehrs. Technical report, Forschungsbericht 22, Technische Hochschule Braunschweig.

Ólafsson, S. and Jumi Kim, S. (2002). Simulation optimization. In Yücesan, E., Chen, C., Snowdon, J., and Charnes, J., editors, *Proceedings of the 2002 Winter Simulation Conference*, pages 79–84. doi:10.1109/WSC.2002.11728711.

Olsson, P. Å. and Regan, M. A. (2001). A comparison between actual and predicted evacuation times. *Safety Science*, 38(2):139–145. doi:10.1016/S0925-7535(00)00064-3.

Opasanon, S. and Miller-Hooks, E. (2009). The Safest Escape problem. *Journal of the Operational Research Society*, 60(12):1749–1758. doi:10.1057/jors.2008.122.

Papadimitriou, E., Yannis, G., and Golias, J. (2009). A critical assessment of pedestrian behaviour models. *Transportation Research Part F: Traffic Psychology and Behaviour*, 12(3):242–255. doi:10.1016/j.trf.2008.12.004.

Park, I., Jang, G. U., Park, S., and Lee, J. (2009). Time-dependent optimal routing in micro-scale emergency situation. *Proceedings - IEEE International Conference on Mobile Data Management*, 10:714–719. doi:10.1109/MDM.2009.122.

Patriksson, M. (2015). *The traffic assignment problem: models and methods*. Mineola, New York: Dover Publications, Inc.

Pereira, L. a., Duczmal, L. H., and Cruz, F. R. B. (2013). Congested emergency evacuation of a population using a finite automata approach. *Safety Science*, 51(1):267–272. doi:10.1016/j.ssci.2012.07.002.

Petrovic, S. and Burke, E. (2004). University timetabling. In Leung, J., editor, *Handbook of scheduling: Algorithms, models, and performance analysis*, pages 1–34. CRC Press, Boca Raton.

Phillips, A., Waterer, H., Ehrgott, M., and Ryan, D. (2015). Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research*, 53:42–53. doi:10.1016/j.cor.2014.07.012.

Polus, A., Schofer, J., and Ushpiz, A. (1983). Pedestrian flow and level of service. *Journal of Transportation Engineering*, 109(1):46–56. doi:10.1061/(ASCE)0733-947X(1983)109:1(46).

Pongcharoen, P., Promtet, W., Yenradee, P., and Hicks, C. (2008). Stochastic optimisation timetabling tool for university course scheduling. *International Journal of Production Economics*, 112(2):903–918. doi:10.1016/j.ijpe.2007.07.009.

Predtechenskii, V. and Milinskii, A. (1978). *Planning for Foot Traffic Flow in Buildings*. Amerind Publishing, New Dehli.

Pursals, S. C. and Garzón, F. G. (2009). Optimal building evacuation time considering evacuation routes. *European Journal of Operational Research*, 192(2):692–699. doi:10.1016/j.ejor.2007.10.004.

Qu, Y., Gao, Z., Xiao, Y., and Li, X. (2014). Modeling the pedestrian's movement and simulating evacuation dynamics on stairs. *Safety Science*, 70:189–201. doi:10.1016/j.ssci.2014.05.016.

Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K. (2005). Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28. doi:10.1016/j.paerosci.2005.02.001.

Redl, T. (2004). *A Study of University Timetabling that Blends Graph Coloring with the Satisfaction of Various Essential and Preferential Conditions*. PhD thesis, Department of Computational and Applied Mathematics, Rice University.

Ross, P., Corne, D., and Fang, H.-L. (1994). Improving evolutionary timetabling with delta evaluation and directed mutation. In *Proceed-*

ings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, PPSN III, pages 556–565, London, UK, UK. Springer-Verlag. doi:10/d9n98z.

Rudová, H., Müller, T., and Murray, K. (2011). Complex university course timetabling. *Journal of Scheduling*, 14:187–207. doi:10.1007/s10951-010-0171-3.

Santos, H., Uchoa, E., Ochi, L., and Maculan, N. (2012). Strong bounds with cut and column generation for class-teacher timetabling. *Annals of Operations Research*, 194(1):399–412. doi:10.1007/s10479-010-0709-y.

Schadschneider, A., Klingsch, W., Kluepfel, H., Kretz, T., Rogsch, C., and Seyfried, A. (2009). Evacuation dynamics: empirical results, modeling and applications. In *Encyclopedia of Complexity and System Science*, pages 517–550. doi:10/bfd967.

Schadschneider, A. and Seyfried, A. (2009). Empirical results for pedestrian dynamics and their implications for cellular automata models. In *Pedestrian Behavior - Models, Data Collection and Applications*, pages 27–44. doi:10.1108/9781848557512-002.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127. doi:10.1023/A:1006576209967.

Schimmelpfeng, K. and Helber, S. (2007). Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum*, 29:783–803. doi:10.1007/s00291-006-0074-z.

Selim, S. Z. and Al-Rabeh, a. H. (1991). On the modeling of pedestrian flow on the Jamarat Bridge. *Transportation Science*, 25(4):257–263. doi:10.1287/trsc.25.4.257.

Seyfried, A., Rupprecht, T., Passon, O., Steffen, B., Klingsch, W., and Boltes, M. (2009). New insights into pedestrian flow through bottlenecks. *Transportation Science*, 43(3):395–406. doi:10.1287/trsc.1090.0263.

Seyfried, A., Steffen, B., Klingsch, W., and Boltes, M. (2005). The fundamental diagram of pedestrian movement revisited. *Journal of Statistical Mechanics*, P10002:13. doi:10.1088/1742-5468/2005/10/P10002.

Shambour, M., Khader, A., Kheiri, A., and Özcan, E. (2013). A two stage approach for high school timetabling. In *ICONIP (1), LNCS 8226*, pages 66–73. doi:10/cvnv.

Sime, J. D. (1995). Crowd psychology and engineering. *Safety Science*, 21(1):1–14. doi:10.1016/0925-7535(96)81011-3.

Smith, R. A. and Lim, L. B. (1995). Experiments to investigate the level of 'comfortable' loads for people against crush barriers. *Safety Science*, 18(4):329–335. doi:10.1016/0925-7535(94)00052-5.

Spearpoint, M. and MacLennan, H. a. (2012). The effect of an ageing and less fit population on the ability of people to egress buildings. *Safety Science*, 50(8):1675–1684. doi:10.1016/j.ssci.2011.12.019.

Stanton, R. J. C. and Wanless, G. K. (1995). Pedestrian movement. *Safety Science*, 18(4):291–300. doi:10.1016/0925-7535(94)00037-4.

Talebi, K. and Smith, J. (1985). Stochastic network evacuation models. *Computers & Operations Research*, 12(6):559–577. doi:10.1016/0305-0548(85)90054-1.

Tavares, R. M. (2010). Design for horizontal escape in buildings: The use of the relative distance between exits as an alternative approach to the maximum travel distance. *Safety Science*, 48(10):1242–1247. doi:10.1016/j.ssci.2010.03.009.

Teknomo, K. (2002). *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*. PhD thesis.

Thompson, P. A. and Marchant, E. W. (1995a). A computer model for the evacuation of large building populations. *Fire Safety Journal*, 24(2):131–148. doi:10.1016/0379-7112(95)00019-P.

Thompson, P. A. and Marchant, E. W. (1995b). Computer and fluid modeling of evacuation. *Safety Science*, 18(4):277–289. doi:10.1016/0925-7535(94)00036-3.

Thompson, P. A. and Marchant, E. W. (1995c). Testing and application of the computer model 'SIMULEX'. *Fire Safety Journal*, 24(2):149–166. doi:10.1016/0379-7112(95)00020-T.

Tong, C. O. and Wong, S. C. (2000). A predictive dynamic traffic assignment model in congested capacity-constrained road networks. *Transportation Research Part B: Methodological*, 34(8):625–644. doi:10.1016/S0191-2615(99)00045-4.

Tuga, M., Berretta, R., and Mendes, A. (2007). A hybrid simulated annealing with kempe chain neighborhood for the university timetabling problem. In *6th IEEE/ACIS international conference on computer and information science*, pages 400–405. doi:10.1109/ICIS.2007.25.

Twarogowska, M., Goatin, P., and Duvigneau, R. (2014). Comparative Study of Macroscopic Pedestrian Models. *Transportation Research Procedia*, 2:477–485. doi:10.1016/j.trpro.2014.09.063.

van den Berg, J., Guy, S., Lin, M., and Manocha, D. (2011). Reciprocal n-Body Collision Avoidance. In *Pradalier C., Siegwart, R., Hirzinger, G. (eds) Robotics Research. Srpinger Tracts in Advanced Robotics, vol 70. Springer, Berlin, Heidelberg.* doi:10/bbzghd.

Van Heuverswyn, K. (2009). *Leven in de Risicomaatschappij, Deel II: Een kritische analyse van de Belgische wetgeving: welzijn op het werk, civiele veiligheid*. Sevesorisicos, Maklu/Garant: Antwerpen.

Venuti, F. and Bruno, L. (2007). An interpretative model of the pedestrian fundamental relation. *Comptes Rendus - Mecanique*, 335(4):194–200. doi:10.1016/j.crme.2007.03.008.

Vermuyten, H., Beliën, J., De Boeck, L., Reniers, G., and Wauters, T. (2016a). A review of optimisation models for pedestrian evacuation and design problems. *Safety Science*, 87:167–178. doi:10.1016/j.ssci.2016.04.001.

Vermuyten, H., Lemmens, S., Marques, I., and Beliën, J. (2016b). Developing compact course timetables with optimized student flows. *European Journal of Operational Research*, 251(2):651–661. doi:10.1016/j.ejor.2015.11.028.

Verraedt, P. (2018). KU Leuven Uurrooster exporteren. Retrieved from `https://verraedt.be/uurrooster_kuleuven`.

Wagner, N. and Agrawal, V. (2014). An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster. *Expert Systems with Applications*, 41(6):2807–2815. doi:10.1016/j.eswa.2013.10.013.

Wren, A. (1996). Scheduling, timetabling and rostering: A special relationship? In Burke, E. and Ross, P., editors, *Practice and theory of automated timetabling V. Lecture Notes in Computer Science*, pages 46–75. Springer: Berlin. doi:10/fpsxtq.

Yang, L., Rao, P., Zhu, K., Liu, S., and Zhan, X. (2012). Observation study of pedestrian flow on staircases with different dimensions under normal and emergency conditions. *Safety Science*, 50(5):1173–1179. doi:10.1016/j.ssci.2011.12.026.

Yin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70. doi:10.1016/j.swevo.2011.05.001.

Yuen, J. K. K. and Lee, E. W. M. (2012). The effect of overtaking behavior on unidirectional pedestrian flow. *Safety Science*, 50(8):1704–1714. doi:10.1016/j.ssci.2011.12.020.

Zanlungo, F., Ikeda, T., and Kanda, T. (2011). Social force model with explicit collision prediction. *EPL (Europhysics Letters)*, 93(6):68005. doi:10.1209/0295-5075/93/68005.

Zarboutis, N. and Marmaras, N. (2007). Design of formative evacuation plans using agent-based simulation. *Safety Science*, 45(9):920–940. doi:10.1016/j.ssci.2006.08.029.

Zhang, D., Liu, Y., M'Hallah, R., and Leung, S. (2010). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, 203:550–558. doi:10.1016/j.ejor.2009.09.014.

Zhao, H. and Gao, Z. (2010). Reserve capacity and exit choosing in pedestrian evacuation dynamics. *Journal of Physics A: Mathematical and Theoretical*, 43(10):105001. doi:10.1088/1751-8113/43/10/105001.

Zheng, X. and Liu, M. (2010). Forecasting model for pedestrian distribution under emergency evacuation. *Reliability Engineering & System Safety*, 95(11):1186–1192. doi:10.1016/j.ress.2010.07.005.

Zheng, X., Zhong, T., and Liu, M. (2009). Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, 44(3):437–445. doi:10.1016/j.buildenv.2008.04.002.

Doctoral Dissertations from the Faculty of Economics and Business

A list of doctoral dissertations from the Faculty of Economics and Business can be found at the following website: `https://www.kuleuven.be/doctoraatsverdediging/archief.htm`.

Department of Information Management, Modeling and Simulation, Brussels Campus
Warmoesberg 26
1000 BRUSSEL, Belgium
tel. +32 2 210 12 11
feb.brussel@kuleuven.be
www.kuleuven.be