FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Conflict solution in cellular evacuation model
Student:	Matej Šutý
Supervisor:	Ing. Pavel Hrabák, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2021/2022

Instructions

Despite their simplicity cellular models are able to capture key phenomena of collective motion. There is a simple definition of decision rules for movement and interaction. The goal of the thesis is to improve the conflict-solution rules in the floor-field cellular model suggested in [2]. Based on numerical simulations, the sensitivity analysis of model parameters should be performed.

Detailed instructions:
1. Get acquainted with the concept of floor-field cellular models for pedestrian flow.
2. Perform research of conflict-solution (more agents entering one cell) in cellular models.
3. Implement the cellular model from [2] with improved rules for conflict-solution.
4. Perform a series of simulations of ...
5. Get acquainted with the principles of sensitivity analysis and appropriate software for its performance, e.g OptisLang [4].
6. Based on the simulations perform a sensitivity analysis of introduced parameters on key observables as total evacuation time or time spent in the room.

–

1. Schadschneider, A., Chowdhury, D. and Nishinari, K. (2010) Stochastic transport in complex systems: from molecules to vehicles. Elsevier.2. Hrabák, P.; Bukáček, M. (2017) Influence of Agents Heterogeneity in Cellular Model of Evacuation. Journal of Computational Science. 2017(21), 486-493.
3. Li, Y., Chen, M., Dou, Z., Zheng, X., Cheng, Y., Mebarki, A. (2019) A review of cellular automata models for crowd evacuation. Physica A: Statistical Mechanics and its Applications 526,120752.
4. Most, T., Will, J. (2011) Sensitivity analysis using the metamodel of optimal prognosis, in: Proceedings of 8th Weimar Optimization and Stochastic Days 2011

Electronically approved by Ing. Karel Klouda, Ph.D. on 17 December 2020 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Conflict solution in cellular evacuation model

Matej Šutý

Department of applied mathematics
Supervisor: Ing. Pavel Hrabák, Ph.D.

May 13, 2021

Acknowledgements

First and foremost, I want to thank God for always being with me in all the hardships and successes. *For everything God created is good, and nothing is to be rejected if it is received with thanksgiving . . .*

I am very grateful for all the support my loving family has given me here and in Slovakia. This thesis would not be possible without the help and thoughtful guidance of my supervisor, Pavel Hrabák, whom I want to thank dearly.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 13, 2021 . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Šutý, Matej. *Conflict solution in cellular evacuation model.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2021.

Abstrakt

Agentní celulární modely mohou být použity pro simulaci evakuace lidí z místnosti. Akce a interakce heterogenních agentů vytváří skupinový pohyb a zachycují tak komplexní jevy v chování chodců. V této práci je představen multiagentní celularní model založený na floor-field modelu. Ten je rožšířen o novou strategii řešení konfliktů, kdy se jeden nebo víc agentů snaží vstoupit na tutéž buňku. Agenti a model maji různé parametry, které ovlivňují řešení konfliktů. Na těchto vstupních parametrech je provedena citlivostní analýza, která objasňuje vliv jednotlivých parametrů na rozptyl výstupních hodnot.

Klíčová slova multiagentní systém, celulární model, agresivita, řešení konfliktů, simulace evakuace, citlivostní analýza

Abstract

Agent-based cellular models can be used to simulate the process of evacuation of people from a room. The actions and interactions of heterogeneous agents create collective motion and capture complex phenomena of pedestrian dynamics. This thesis presents a multi-agent cellular model based on floor-field model and is extended by a new strategy for solving conflicts when two or more agents attempt to enter the same cell. The agents and the model have various parameters that influence the conflict solution. A sensitivity analysis on these parameters is performed that reveals the individual contribution of variance in the results.

Keywords multi-agent system, cellular model, aggressivity, conflict solution, evacuation simulation, sensitivity analysis

Contents

List of Figures

List of Tables

Introduction

The ability to predict and model the movement of pedestrians during evacuation is valuable knowledge. Understanding human behavior was always challenging, yet even individual activities such as movement are difficult to determine reliably. Pedestrian movement can be examined by physics, sociology, or psychology, but these methods fall short in complicated scenarios.

Increasing demands on safety procedures in buildings or during various events, such as evacuation, or demonstration, call for answers on pedestrian dynamics. Some available solutions provide precise simulations on how people move and interact, and others focus on real-time results and adaptation to a fast-changing environment.

The environment and setting are specific for each building, location, and various groups of people. It is essential to understand the influence of individual parameters on the process with so many possible settings. Simulating the process on a whole parametric space is very slow and could be sped up using only the significant parameters selected by sensitivity analysis.

This careful selection of input parameters for the evacuation model can produce results faster and more precisely by providing insight into the importance of individual parameters. Understanding the evacuation process better, we can overcome critical situations that might lead to the loss of lives.

An experiment of pedestrian dynamics in a room was performed at Czech technical university in Prague. The experiment was supervised by Pavel Hrabák, and the results were used to create a model for simulating the pedestrian movement. I was offered to extend the capabilities of the model, to and perform further research to simulate the evacuations more precisely. The precision depends on key observable results, such as total evacuation time and the flow of pedestrians through an exit.

The multi-agent model I implemented in this thesis is inspired by floor-field cellular model. Cellular models are surprisingly capable of capturing complex phenomena and processes with only a simple set of rules.

This thesis focuses on improving the feature of conflict solution and provides a new approach to the movement of agents in the model. The improvements are linked to the sensitivity parameters of agents to the environment. The final contribution of this thesis is the sensitivity analysis of input parameters to the variance in observable quantities, namely the total evacuation time.

This thesis is organized as follows: Chapter 1 focuses on the research of current approaches and trends used to model pedestrian dynamics. The distinction between microscopic and macroscopic models is explained. The inspiration for the model used in this thesis is mentioned and briefly explained, as well as several current models for pedestrian dynamics are described.

The essential contribution of this thesis dwells in the research of sensitivity analysis (SA). The exact form of SA used in this thesis is highlighted.

The following Chapter 2 explains the theory and the practice of the model. The first part in Section 2.1 provides the theoretical fundamentals for the evacuation model — the core components of the model and the mathematical methods. Later in Section 2.2 the software tools used in this research are mentioned . The particular implementation of components in the evacuation model is described.

The approach to sensitivity analysis is described in Chapter 3, where necessary components of the analysis are explained (input parameters, observable quantitities, methods of SA).

The following Chapter 4 shows the results of analysis on the simulation data. The spotlight is on global sensitivity analysis of individual input variables and their contribution to the variance of observable quantities. The individual contributions present a better understanding of how the evacuation is simulated.

In the end, new discoveries are explained and summarized. Results and discoveries of this thesis can be further researched and analyzed.

Goal

This thesis aims to capture phenomena of collective motion of pedestrians in a room using a multi-agent cellular model, improve the solution of conflicts, and analyze the impact of input parameters on the process. An inspiration for the simulation model is the floor-field model that presents fine resolution of individual agents that interact with each other and the environment. A new and improved set of rules for solving conflicts is used. Rules are affected by parameters of aggresssivity and sensitivity of agents to the environment. A number of simulations are executed, and sensitivity analysis of input parameters is performed. The outcome of the analysis presents the influence of individual input parameters. Key observable quantities are total evacuation time and flow of pedestrians through main exit. Sensitivity analysis of parameters shows a connection between variance in input parameters and variance in observables.

Chapter 1

State-of-the-art

The study of pedestrian dynamics is a hot topic frequently researched throughout the world (e.g. in Munich [3], Jülich [4], Tokyo [5], Clayton [6]). With it's own history dating back to 1975 [7], this field took off in the last three decades.

1.1 Pedestrian dynamics

Nowadays there are three major approaches to modeling the evacuation process [4]:

- Fluid-dynamics.
- Artificial intelligence.
- Agent-based models (ABM).

Fluid-dynamics In general the models using fluid-dynamics, for example model using smoothed particle hydrodynamics (SPH) [6], focus on the macroscopic collective behavior such as line formation, jamming or flow oscillations. They are based on various mathematical equations — Euler equations, Eikonal-type equations, or they use computational fluid dynamics. These models have fast calculations and can provide real-time results with lower precision. This is especially useful for crowd management or large simulations. They show an overview of the process with low or zero resolution on the individual pedestrians. These results, however, are suitable for civil engineering, for example in urban planning. Macroscopic models are simpler than microscopic models and produce results more quickly while retaining reasonable predictions. They are used for modeling large crowds or general flowsa and these models can be used for creating quick concepts where flexibility is the key.

Artificial intelligence Sometimes only one approach is not enough and models using artificial intelligence present themselves as a viable answer. These models can consist of different components that pull together, each focusing on one specific area. For example, wildfire evacuation model [8], consist of conceptual model, choice model and traffic model that are orchestrated by AI algorithm. Their use is rather narrow and depends heavily on data they operate with. The models using AI are hard to summarize because the field of AI is very broad.

Agent-based models (ABM) Pedestrian dynamics models using agents express the concept of artificial societies [9]. The essence of the models are individual agents that are distinguishable and substantiate the heterogeneity of the whole system. These agents can operate on various levels of autonomy and mimic the behavior of people during evacuation, demonstration, shopping or other scenarios.

With fine resolution of individual agents come the increased requirements for computation and rather narrow use (e.g. ABM unable to demonstrate the maneuver of first-aid person [3]). They are used mostly for research of specific phenomena at microscopic level or for situations that rely on social interactions. ABM can operate on discrete [10], continuous [11] or combined [12] space. This space is mostly two-dimensional but there are models that can operate with inclined planes or stairs, even elevators and public transport [13].

Microscopic model individually analyses each pedestrian (agent) in the process. Agents create many interactions and this enables more precision at the cost of time-consuming computation. These models are used for discovering new phenomena or used as tool to better understand the processes within.

Other available solutions There are many other approaches for microscopic modeling of pedestrian dynamics:

The hybrid mathematical model ODE, that is based on the Helbing social-force discrete model, employs *classical Newtonian laws of point mechanics, as the motion of each individual is described by an ordinary differential equation (ODE)* [14].

One commercially available agent-based evacuation simulator, *Pathfinder*, is also a hybrid approach, that provides immense features: 3D modeling, contour plots, assisted evacuations for people with special needs, Monte Carlo simulations and many more. It was designed to accommodate tens of thousands of occupants. This model employs several components (agents, artificial intelligence and triangulated mesh) that calculate minimal-cost direction of each pedestrian based on rules of *steering* technique. Extensive explanation can be found in [12].

The open source evacuation simulator framework JuPedSim [15] utilizes three main modules, loosely connected, that can work independently. JuPedSim is an answer for research of evacuations because it provides already finished auxiliary components — visualization module and reporting module. The simulation module can be modified, to some degree, to fit the needs of individual research. At the moment the simulation model employs two force-based models: the Generalized Centrifugal Force Model and the continuous physical force Gompertz model which also uses differential equations. This framework is still being developed. The *floor-field* model in this thesis relies on different approach (conflict solution) to model the movement of agents and the current solution implemented in JuPedSim can't be utilized or customized.

The microscopic pedestrian and crowd dynamics can also be simulated by Vadere, an open source framework developed by the research group of Gerta Köster at the Department of Computer Science and Mathematics at the Munich University of Applied Sciences [3]. Compared to JuPedSim, Vadere allows only two-dimensional systems. Vadere is based on Optimal Steps Model (OSM) that utilizes several layers: locomotion, individual and interaction. The locomotion layer uses continuous geometry and models the way in which people move, according to the dynamic navigation field that is updated at each time step. The OSM introduces real human step that adjusts to various situations, e.g. the steps are shorter when pedestrian is confronted with dense crowd and slows down. What follows is the individual layer, that describes the characteristics of each person - space requirements, desired speed, individual plans... Some of these parameters are assigned default values, which are derived from studies of the pedestrian flow [16]. The last layer is the interaction layer that models social and environment interactions.

Selvek from the Faculty of information technologies at Czech technical university in Prague proposed algorithm LC-MAE (Local Cooperative Multi Agent Evacuation). His algorithm assumes several types of agents with different levels of rationality, who move on a undirected graph from endangered part into the safe part as quickly as possible. It is achieved by local planning of agents and their paths. Compared to solutions using network flow and centralized planning, his solution produces results worse only by a small factor [17].

The goal of this thesis is different from the research of Selvek, as the cellular evacuation model in this thesis does not aim to calculate the fastest or most optimal evacuation plan, but rather models the behavior of pedestrian during evacuation based on input parameters.. The contribution of this thesis lies in the study of microscopic processes within the model, and SA of individual input parameters.

1.2 Cellular automaton

A cellular automaton (CA) is a complex mathematic or computational system with discrete elements (e.g. agents), that can be in finite number of states. These elements are updated according to a simple set of rules, such as interacting with other elements or environment (e.g. floor-field) [18]. CA needs to be initialized, and because it abides by strict rules, it is deterministic in behavior, when it's not using any stochastic selection. The variance in results, due to stochastic selection, can mimic the randomness in human behavior and can be a desired feature of the model. The abstract system of CA can emulate very diverse processes. CA has been used, for example, to emulate a dynamic universe in *Game of life* [19]. This famous example demonstrates CA's capability to produce highly complex behavior with very simple rules [20].

1.2.1 Floor field CA

The inspiration for model used in this thesis is the floor field cellular model described by Katsuhiro Nishinari and Andreas Schadschneider in Article [21]. Main characteristic of this model is in the use of several floor fields — *static*, *dynamic*, *fire hazard*, ... These can be explained as two-dimensional lattices made of discrete cells that hold some value and abide to certain rules. Pedestrians in the evacuation, who are agents with individual parameters, move in the rectangular two-dimensional and interact with these lattices. Agents are updated in synchronous intervals but thanks to their individual and heterogeneous period of movement, they can have different speed on the grid. *"For the case of evacuation processes, the static floor field S describes the shortest distance to an exit door. The field strength S_{ij} is set inversely proportional to the distance from the door. The dynamic floor field D is a virtual trace left by the pedestrians similar to the pheromone in chemotaxis [22]. It has its own dynamics, namely diffusion and decay, which leads to broadening, dilution and finally vanishing of the trace. At $t = 0$ for all sites (i, j) of the lattice the dynamic field is zero, i.e., $D_{ij} = 0$. Whenever a particle jumps from site (i, j) to one of the neighboring cells, D at the origin cell is increased by one"* [21].

The update rules of said CA are:

1. The dynamic field D is updated according to decay rules.
2. For each agent the transition probability p_{ij} (in this thesis called *attractivity*) for a move to neighbor cell is calculated.
3. Each agent chooses randomly a target cell based on probability p_{ij}.
4. When two or more agents attempt to enter the same cell, conflict occurs and needs to be solved.
5. Agents move to their target cell if possible and update dynamic field D.

Detailed description of the procedures and components is in the Section 2.1. The course of the simulation is affected by a set of parameters which are explained in said section as well. These parameters will by analyzed by means of sensitivity analysis, described below in Section 1.3.

1.3 Sensitivity analysis

The research experience on this topic calls for a efficient computation and careful selection of input parameters. The stochastic selection used in simulation adds uncertainty to the model — the output varies for the same input parameters. As such, it is an expected feature because it imitates the randomness of human behavior.

"Aleatory uncertainty, also called stochastic or variable uncertainty, refers to uncertainty that cannot be reduced by more exhaustive measurements or a better model. Epistemic uncertainty, or subjective uncertainty, on the other hand, refers to uncertainty that can be reduced" [23]. Sensitivity analysis used in this thesis analyzes the epistemic uncertainty of results that is affected by input parameters with the OptiSLang software.

Current research of evacuation models also focuses on sensitivity analysis of uncertain input parameters and their contribution to simulation output. In Article [24], Köster et al. use SA to study the influence of two input parameters, number of protesters and standard deviation of their free-flow speed, on the length of the protest march. Their method of SA computes Sobol' indices with methods based on polynomial chaos expansions. Other research was performed by Claudio Feliciani and Katsuhiro Nishinari who present a method to quantify the amount of congestion in pedestrian crowds, which is estimated on the velocity vector field obtained from video recordings of moving crowds. To find the best parameter combination of the vector field, methods of SA were utilized [25].

There are several other ways to calculate the influence of input parameters, detailed overview can be found in Article [26].

Chapter 2

Realisation

2.1 Model definition

Evacuation, core topic of this thesis, is the process of moving people from a place, possibly dangerous, to somewhere safe. The evacuation model in this thesis represents the people in evacuation as agents who move on a grid that acts as a room or other area. The grid is two-dimensional, rectangular, and consists of discreet cells. Cells can be occupied by agents who move from one cell to another in 8 directions or stay in the same cell. The movement in the model is strictly discrete due to the nature of the cellular model. Movements are executed simultaneously, which leads to conflicts when two or more agents attempt to enter the same cell. Only one agent can be present in a cell at a time.

The agents are somewhat autonomous and react to the environment. They generally move in the direction to the exit. In this paper, only grid with one exit is considered. Because the exit is a cell, only one agent can leave the room at every synchronous update of the simulation. There is an interaction with agents in adjacent cells.

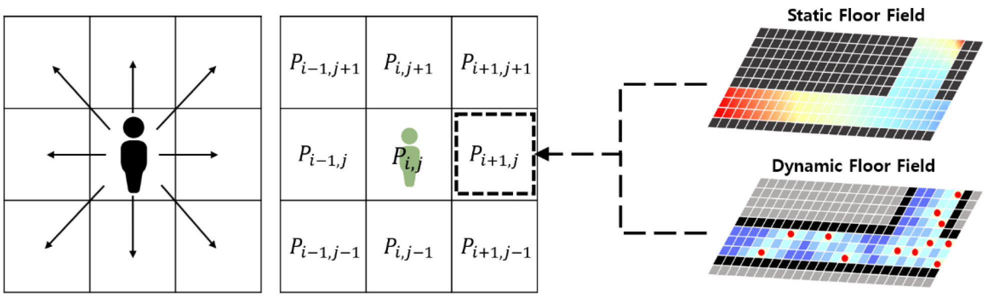


Figure 2.1: Example of cellular floor-field model utilizing *static* and *dynamic* field, where agent can move to cells in Moore neighbourhood. *Taken from [1].*

The collective motion of agents is reproduced using various strategies: choosing destination cell and solving conflicts, and using heterogeneity in the parameters. The strategies are affected by parameters of aggressivity and sensitivity of each agent and by parameter of friction. The parameter of aggressivity plays a role in situations when two or more agents attempt to enter the same cell.

2.1.1 Components

AGENT

Model is populated with n agents which are placed to unique positions on the grid. Agents have specific parameters such as aggressivity $\gamma \in [0, 1]$, inner time period $\tau \in \mathbb{R}$ and sensitivity parameters: to *static* potential $k_S \in \mathbb{R}$, to occupancy $k_O \in [0, 1]$, to diagonal movement $k_D \in [0, 1]$.

There are various ways to assign aggressivity γ to agents:

- groups of agents with the same γ
- stochastically choose γ from a probability distribution

Each agent can move to 8 adjacent cells in the Moore neighborhood [28] or decide to stay in his cell as is shown in Figure 2.2. The preference of the destination cell is determined by the strategy for choosing the destination cell described in Section 2.1.2.

GRID

The space in which agents move is a rectangular two-dimensional grid $G \subset \mathbb{Z}^2$ of width W and height H. Grid consists of cells, which position is (y_1, y_2) : $y_1 \in \{0, 1, \ldots, W - 1\}$, $y_2 \in \{0, 1, \ldots, H - 1\}$. Agents cannot leave the grid otherwise than through the exit placed on the border of grid. Corresponding to real world the grid is *not* spherical and it is not possible for agents to move from a cell on border and appear in a cell on the opposite side.

EXIT

Agents can leave the grid only through exit cell e placed on the border of the grid. Exit has the lowest static potential $S(e) = 0$. In this thesis, only one exit is considered. The moment when an agent moves to exit, he is removed from the grid. In each time-frame, only one agent can move to exit.

STEP

Evacuation is simulated in a sequence of m steps $\{s_1, \ldots, s_m\}$ executed by model. Each step consists of several phases (see Section 2.2.2): selection of destination cells, conflict solution, agent movement $\ldots$In this thesis, terms step and epoch are used interchangeably. The term step is used mostly during theoretical explanation of the model and it's components, while epoch is mentioned in simulations and explanation of various phenomena during evacuations.

TIME FRAME

Continuous real life duration T of evacuation is divided to k time-frames $\mathbf{t} = \{t_1, \ldots, t_k\}$. One step s_i of model takes $l = [t_i, t_{i+1}]$ time. Time-frame t_i covers one step s_i of the evacuation process. Agents move in two-dimensional grid in 8 directions. In this thesis, all agents have uniform inner time period τ, which is the duration of nominal vertical or horizontal movement. The movement in diagonal direction takes $\sqrt{2}$ times longer. To allow for more periodic update, $\sqrt{2}$ is approximated by rational constant $\frac{3}{2}$ [29] resulting in duration of diagonal movement equal to $\frac{3}{2} \cdot \tau$. Because all agents have identical τ, one step of the model takes $l = \tau$ time. The model implementation can assign heterogeneous τ to agents, but this would not contribute to the goal of this thesis. Detailed description of heterogeneity in velocity of agents and the impact on evacuation can be found in [29].

CELLS

Cell y is an element of grid G in position $(y_1, y_2) :\ y_1 \in \{0, 1, \ldots, W - 1\}, y_2 \in \{0, 1, \ldots, H - 1\}$. Each cell can be occupied at most by one agent at a given time-frame. Agents can move in 8 directions to cells in Moore neighborhood or decide to stay in the same cell. The choice of Von Neumann neighborhood was rejected, because it does not allow agents to execute diagonal movement, which is influenced by k_D and thus essential. Cell y holds the value of static potential $S(y) \in \mathbb{R}$. For cell y and exit e in grid G is $S(y) = L_1(y, e)$ where $L_1(y, e)$ is Manhattan distance [30] from cell y to exit e. The function of static potential could be different, for example shortest distance to exit calculated with Dijkstra algorithm. In this thesis rectangular room without obstacles is considered, so L_1 distance metric calculates shortest distance to exit.

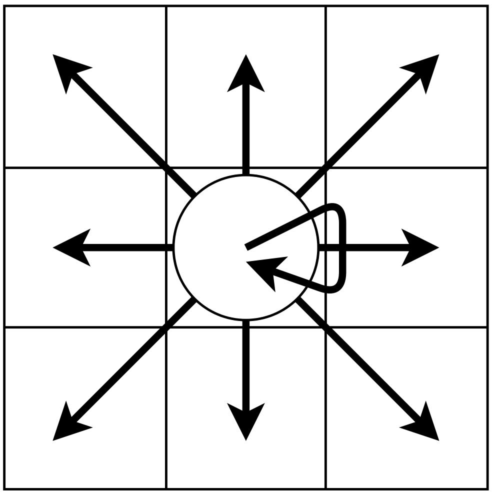


Figure 2.2: Agent, white circle in the middle, can move in 8 directions according to Moore neighborhood or decide to stay in his cell. The probability of moving to individual cell is calculated by strategy for choosing destination cell. Agents can't enter cell occupied by other agent or leave the grid.

2.1.2 Methods

STRATEGY FOR CHOOSING DESTINATION CELL

In each step of the simulation, all agents calculate the attractivity of individual cells in their neighborhood N. N is a set of adjacent cells to the cell occupied by agent, and the occupied cell as well. Attractiveness or attractivity is used to calculate probability $P \in [0, 1]$, which is the probability of agent selecting this cell as his preferred destination cell. The selection is executed in stochastic manner. Value of attractiveness depends on strategy for choosing destination cell.

In the equations below can be found following members: $S(y)$ is a function of static potential of cell y, where $S(y) \geq 0$. Associated parameter k_S is the sensitivity of agent to the static potential of a cell. $O(y)$ is the indicator of occupied cell y. When y is occupied by an agent, $O(y) = 1$, otherwise it is zero. Associated parameter k_O is the sensitivity of agent to the occupancy of cell. $D(y)$ is the indicator of diagonal motion from agent's origin cell x to destination cell y. When y can be entered from x by diagonal motion, $D(y) = 1$, otherwise it is zero. Associated parameter k_D is the sensitivity of agent to diagonal motion.

- **Strategy A:**

  Strategy A was described by Pavel Hrabák and Marek Bukáček in Article [2]. Probability P of agent, who is in cell x, moving to adjacent cell $y \in N$ is:

$$P(y \leftarrow x \mid N) = \frac{\exp(-k_S S(y))(1 - k_O O(y))(1 - k_D D(y))}{\sum_{z \in N} \exp(-k_S S(z))(1 - k_O O(z)(1 - k_D D(z))} \tag{2.1}$$

  The nominator is the attractivity of cell y to agent, which is normalized with attractivites of other cells in N and thus can be used as probability.

- **Strategy B:**

  Strategy B, introduced in this thesis, is more affected by the sensitivity to the occupancy of cells k_O. Probability P of agent moving to cell y from cell x, is calculated from P_O and P_S.

$$P(y \leftarrow x \mid N) = k_O P_O(y) + (1 - k_O) P_S(y) \tag{2.2}$$

  The term P_O focuses on occupancy of cell. Notice the missing parameter k_O in $(1 - O(y))$ which is different from strategy A. Term P_O is normalized across P_O of neighbor cells from N.

$$P_O(y) = \frac{\exp(-k_S S(y))(1 - O(y))(1 - k_D D(y))}{\sum_{z \in N} \exp(-k_S S(z))(1 - O(z)(1 - k_D D(z))} \tag{2.3}$$

  The term P_S takes into account the static potential $S(y)$ and makes the agent move in the correct direction to the exit, and the indicator of diagonal motion $D(y)$. Term P_S is also normalized across neighbor cells from N.

$$P_S(y) = \frac{\exp(-k_S S(y))(1 - k_D D(y))}{\sum_{z \in N} \exp(-k_S S(z))(1 - k_D D(z))} \tag{2.4}$$

Figure 2.3 presents a situation, where agent A has blocked path to *exit*, where agent B stands in the way. Cells are colored and numbered in correspondence to the graph legend in Figure 2.4. Agent A calculates the probabilities of transition to cells $0, \ldots, 8$ based on *strategy for choosing destination cell.*

Figure 2.4 shows the differences in probability distribution of moving to cells in neighborhood N of agent A. This 100% stacked graph depends on the parameter of sensitivity to occupancy k_O on the x-axis. With lower k_O, the agent is more likely to enter an occupied cell. It can be seen, in both strategies, that in low values, agent A is more likely to enter cell 7 (occupied by agent B), whereas, in high values, he prefers to stay in his cell. The new proposed strategy A in Subfigure (a) shows a more linear decomposition of probabilities than in strategy B in Subfigure (b). With strategy B the value of k_O can be interpreted in more clear way. More figures with different values for parameter of sensitivity to static field k_S are found in Appendix B.2.

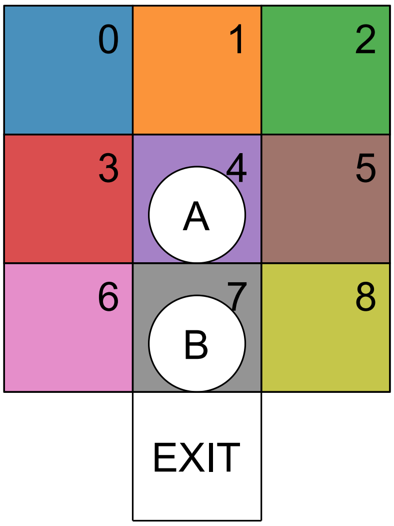
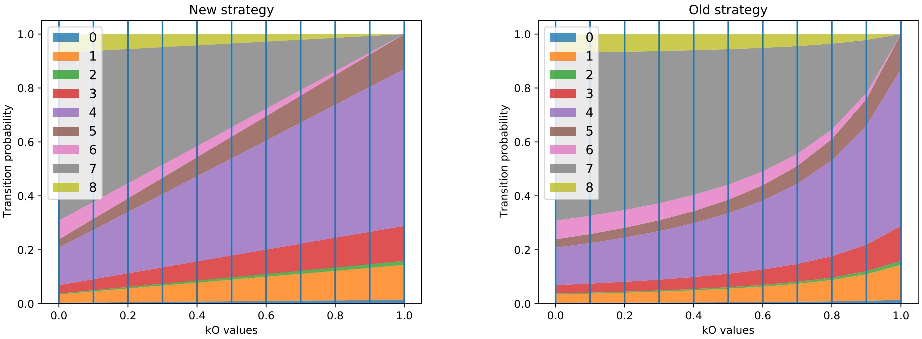


Figure 2.3: Situation of two agents A and B. Agent B is blocking the exit for the other agent.



(a) New strategy B.

(b) Old strategy A.

Figure 2.4: Comparison of strategies for choosing destination cell. Different colors show probabilities of moving to cells in neighborhood calculated for agent A in situation with blocked exit, see Figure 2.3. Numbering in legend corresponds to cell numbering in Figure 2.3, Parameters: $k_S = 1.5,\ k_D = 0.5$. On the left is the new *strategy B*, that produces more even probability distribution of moving to adjacent cells. This way the impact of k_O can be explained in a more clear way.

AGGRESSIVITY

Aggressivity $\gamma \in [0, 1]$ as agent parameter, defines the ability of agent to win conflict. This term was introduced in research by Pavel Hrabák and Marek Bukáček in Article [2]. Conflicts happen when two or more agents select same cell as their destination cell. This thesis provides two strategies for solving conflicts described below in Section 2.1.2.

There are various methods (see Section 2.1.1) to assign γ to n agents:

- assign aggressivity to all agents from a uniform distribution on interval $\gamma \in [x, y], x, y \in [0, 1] \wedge x \leq y$
- assign constant γ to disjunctive groups of agents
- combination of the previous two.

The γ values of agents are generally discretized to $h : h \leq n$ unique values. When only h values are assigned to the agents, the conflicts occur more frequently. If agents were assigned random values from continuous uniform interval, the probability of two agents having the same γ would be zero.

BONDS

In every step s_i at most one agent can be present in a cell x. The principle of bonds [29] is used in order to allow agent a_1 in cell y_1 to move to a cell y_2 occupied by agent a_2 in following step $s_{\mathrm{i}+1}$. Immediately when a_2 leaves y_2 to move to other cell, a_1 will enter now empty y_2. *This principle enables the motion in lines within one algorithm step, which is desired phenomenon of pedestrian flow* [29].

STRATEGY FOR SOLVING CONFLICTS

Conflicts happen when two or more agents choose cell y as their destination cell. There are two strategies for choosing winning agent.

- **Strategy A:**

  From agents $\{x_1, \ldots, x_k\}$ an agent x_j is chosen with $\gamma_j = \mathrm{argmax}_{1,\ldots,k}$. If no other agents has same the aggressivity, agent x_j wins the conflict immediately. If there are two or more agents with same highest aggressivity a parameter of friction μ affects the conflict. With probability $P = \mu(1 - \gamma_j)$ none of the agents will enter the cell and they will all remain in their positions. This event is called blocking event. With $\bar{P} = 1 - P$ agent x_j will enter the cell. Strategy A is explained in detail in Article [2]. In Figure 2.5 are shown three possible outcomes of a conflict.

Figure 2.5 shows all possible outcomes of conflict with three agents solved by *strategy A*. The upper left scheme is the initial position of agents who all attempt to enter the cell in the middle. Two agents have same γ_1 and the agent on the left has γ_2. First possible solution of this situation is shown in lower left corner when agent with $\gamma_2 > \gamma_1$ wins the solution and enters the cell.

On the right is depicted a situation with two agents of γ_2, and one with γ_1, who attempt to enter the cell in the middle. This situation, with strategy A, can result in one of the two agents with γ_2 entering the cell, as can be seen in the bottom row on the sides. Also, all of the agents could end up staying in their positions due to blocking occasion, as can be seen in the scheme in the middle of the bottom row.

- **Strategy B:**

  Strategy B is implemented to repress jamming. During the implementation and testing an unsolicited phenomena of jamming near bottleneck, due to low aggressivity of agents, was discovered. To prevent jamming the sensitivity parameter μ could be set to lower value but it will also lead to lower number of desired blocking occasions in other parts of the evacuation. When agents with low aggressivity $\epsilon \ll 1$ get to conflict situation, the probability of one agent winning the conflict is also very low: $P = \mu \cdot (1 - \epsilon)$. This strategy selects the winning agent a_i stochastically with probability P_i proportional to agent's respective aggressivity γ_i. Set of agents $\{a_1, \ldots, a_k\}$ with respective aggressivity parameters $\{\gamma_1, \ldots, \gamma_k\}$ produces for each agent a_i a probability $P_i = \frac{\gamma_i}{\sum_{j \in k} \gamma_j}$.

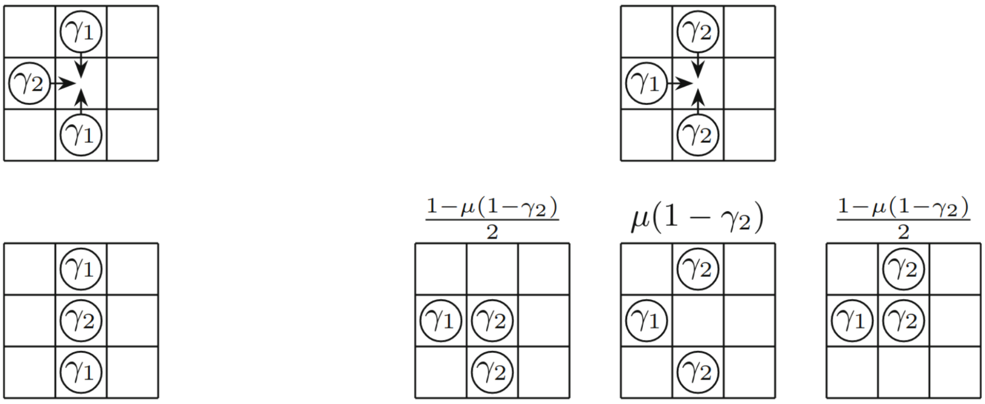


Figure 2.5: Conflict solution for $\gamma_1 < \gamma_2$. Left: more aggressive wins the conflict over two less aggressive agent. Right: the conflict of two more aggressive agents can resolve by the blocking of the movement. *Taken from* [2].

New proposed strategy B did not bring any new benefits as was noticed during the testing of the model. The friction parameter μ affects the creation of blocking occasion before the proportional probabilities of agents, based on their aggressivity, can be taken into account. With strategy B jamming still occurs. Because of this reason, further simulations performed on this model use strategy A.

The goal of this thesis was to perform research on conflict-solution and implement improved rules. The way in which conflicts are solved is significantly affected by agents themselves who participate in conflicts. Which agent, and where he gets in conflict, is hence part of the conflict solution and the new proposed *strategy A for choosing destination cell* focuses on it.

2.2 Implementation

2.2.1 Tools

Python

Python is very popular programming language among data scientists [31] with many packages that extend the functionality. The low learning curve and intuitive approach determined Python as adequate tool for this project. Due to popularity, practicality and frequent development of Python, implementation of this model can be further researched, modified and extended as is the wish of the author.

Mesa framework

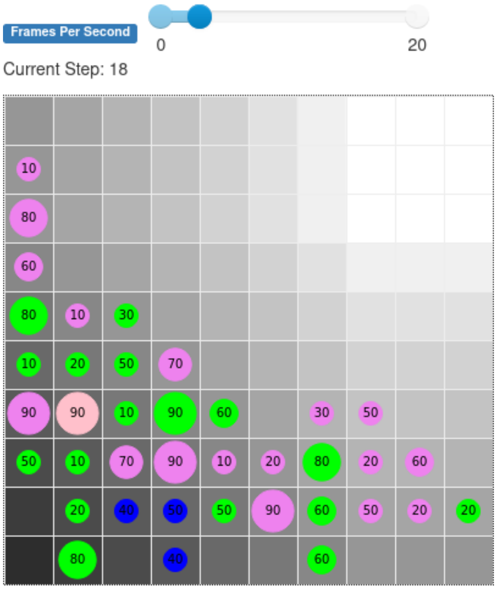


Figure 2.6: Vizualisation of evacuation in Mesa.

Agent-based modeling is a computational methodology used in social science, biology, and other fields, which involves simulating the behavior and interaction of many autonomous entities, or agents, over time [32]. My research of available agent-based modeling frameworks revealed Mesa as an up-to-date framework which is currently used in many projects [33, 34]. This framework allows broad customization: from discrete to continuous movement, synchronous or asynchronous update and various types of data collection. Data collections can be customized with popular data science packages [35]. Together Mesa, Python, and other packages (Numpy and Pandas [36], Matplotlib [37]) create a flexible and customizable environment that suits the needs of this thesis.

Mesa framework allows progressing in the simulation with different speed or by single steps, which is very helpful for examination of the course of the simulation. Overview of the situation can also be demonstrated with real-time graphs of data-collectors for various observed values. Figure 2.6 presents a web interface for executing simulations in Mesa framework. This snapshot depicts simulation with exit placed in the lower left corner and agents with different colors and numbers. The colors represent inner state of the agent. For example the agent, who lost conflict and remained in its position has blue color. The number is the aggressivity of agent, that is also visualized by bigger circle diameter for more aggressive agents.

2.2.2 Evacuation model

PModel

PModel is an object which executes all processes in the evacuation simulation. PModel provides an interface for creating a model with specific parameters and allows data exploration by accessing the data during and after the simulation. Each model is assigned a pseudo-random number generators (PRNG) [38] , that are initialized with a seed. Different seeds in simulations generalize the concept of evacuation while making it possible to replicate the results.

PAgent

PAgent represents a pedestrian agent in the simulation. Agents interact with the environment by sensing and moving. Each agent holds his set of parameters that calculate probabilities for destination cells. The strategy for selecting destination cell in Section 2.1.2 can be selected. The processes of movement in the model are strictly sequential. This approach allows line formation and homogeneous movement of all agents.

The Mesa framework allows visualizing the course of the simulation. In Figure 2.6 are displayed circles — *PAgents* — with a number showing the aggressivity $100 \cdot \gamma$ value and different colors, that represent inner state of agents. To better understand the terms *head* bond, blocked agent and exhausted agent, it is recommended to consult paragraph *PSchedule* in this Section, where said terms are explained.

- **Green** is the default color.
- **Red** is the color of an aggressive *PAgent* — his γ is higher than 0.5.
- **Violet**, the *head* bond agent did not move.
- **Blue** is the color of blocked *PAgent* — blocking occasion occurred.
- **Pink** is the color of exhausted *PAgent* — out of time-frame due to diagonal movements.

PCell

PCell is an atomic element in the grid that agent can enter. PCell determines the selection of destination cell for agents trying to enter this cell and resolves conflicts with stochastic probabilistic selection using PRNG G_2. Strategy A or B for solving conflicts in Section 2.1.2 can be selected. In Figure 2.6 we can see rectangular tiles — *PCells* — in different shades of grey. The darker the color, the lower the static potential $S(y)$. Black tile is the *exit*.

PSchedule

PSchedule orchestrates the sequence of steps in the simulation. Time in the model is split into time-frames. Agents moving at different speeds (for example because of diagonal movement) enter the sequence according to the duration of their movement [29]. Each movement, also remaining in its position, updates the time when their movement ends according to τ. The movement in diagonal motion takes $\sqrt{2}$ more time than orthogonal movement hence the agent's inner time period is increased by rationalized $\frac{3}{2}$ of nominal orthogonal movement. Nominal orthogonal movement has duration of 1 time-frame, the timespan of one step in the simulation. One step can be split into 4 phases: *start, scan, solve, step.* Object PSchedule executes phase *start* first for a group of all *PAgents* and when all *PAgents* finished their phase, it executes the same phase for all *PCells.* The remaining phases (*scan, solve, step*) are executed only by *PAgents.* This order of execution delegates the tasks of calculating probabilities of destination cells to *PAgents* and the task of selecting destination cell and solving conflict to *PCells.*

A detailed description of each phase:

1. ***Start***

   - *PAgent* sets all his states (*destination cell, color, head, tail)* to default. *PAgent* then calculates the attractivity of adjacents cells based on the *strategy for choosing destination cell.* The destination cell is chosen stochastically, and *PAgent* joins the queue of agents attempting to enter this cell.
   - *PCell* selects one agent from queue. If more than two agents are attempting to enter this cell, a conflict occurs. *PCell* solves the conflict based on the *strategy for solving conflicts* described in Section 2.1.2. *PCell* can create a blocking event when no agent enters this cell. In that case, all agents are assigned the blue color. Otherwise, it chooses a successful agent. That agent is then assigned this cell as his *destination cell* and creates a *head* bond to an agent occupying this cell if there is one. Also, the occupying agent creates a *tail* bond in direction to the successful agent. This is the principle of bonds from Section 2.1.2, which allows agents to form lines, and it is important for phase *scan.*

2. ***Scan***

   - If *PAgent* has no *head* bond, it means he can immediately enter his *destination cell*, because it is not occupied, and he can move to this cell. By moving to his *destination cell*, he left his former cell empty, and signaled the *tail* bond agent attempting to enter his former cell to move there. If the agent entering this phase has a *head* bond to an agent occupying his *destination cell,* he skips this phase and waits for a signal from the *head* bond agent. *PAgents* that move to an empty cell start a sequence of movements for agents in their line by freeing their former cell.

3. ***Solve***

    - When PAgent was assigned a *destination cell* by *PCell* but *head* bond agent in the *destination cell* did not move it is desired to record this event for statistics. The color of *PAgent* is set to violet. The time of the agent is updated. When the agent's time after moving is not in the interval of the next step, assign him the pink color to display him as exhausted.

4. ***Step***

    - When a *PAgent* reaches *exit*, he is removed from the grid and the schedule.

PGenerator

The deterministic run, useful for repeated runs, is achieved by generating the model with same initial conditions and same probabilistic selections during the evacuation.

Initial conditions (positions on the grid and generated parameter values of agents) are created by deterministic PRNG G_1 with seed s_1. Two simulations with different settings, for example different k_S values, can be performed with identical initital conditions, when using same seed s_1. This way it is assured, that different results of the simulation can be linked to difference in parameters, and not accounted to the randomness in placing the agents on the grid.

Secondary, the course of the evacuation is directly affected by the stochastic selection in choosing destination cells and in solving conflicts. It needs to be said, that it's a desired feature, because it can model the randomness in human behavior. To control the stochastic selection, generator G_2 with seed s_2 is introduced. Generator G_2 executes the stochastic selection in choosing destination cells and in solving conflicts. G_2 is isolated from generator G_1, they do not affect each other. Analysis of the model, for example SA, requires repeated runs with same initial conditions, and with the same set of input parameters. When a new seed s_2 is assigned to G_2 in each run, the outcome of the simulation is varied, and generalizes the concepts in the model. Using identical seed s_1 for G_1, the difference in results can be linked to difference in input parameters, and not to the randomness of the model, even though the randomness always increases variance in results.

Chapter 3

Sensitivity analysis

This chapter presents research based on quantitative and qualitative analysis of the input parameters. On one hand, the quantitative analysis measures the influence of the individual input parameters on the simulation using sensitivity analysis in OptiSLang software. On the other hand, qualitative analysis of the model explains the observed structures, that form during the simulation, and show how the parameters are linked together. The qualities of the model were researched by observing running simulations and by statistical methods of analysis.

3.1 Input parameters

The course of the simulation is affected by input parameters. Some input parameters, such as number of agents n, size of the room $W \times H$, or location of the *exit*, are not subject to calibration of the model as they correspond to the real world setting and can't be manipulated. The investigation of number of agents n, however, can show how the model responds to the congestion phenomenon — more agents are approaching the exit, respectively bottleneck, than are being evacuated, respectively egressed, through the exit.

Individual agents are assigned different aggressivity parameter γ. With higher value, the agent is more likely to win the conflict. The influence of γ is described in Section 2.1.2.

Simulations in this thesis always assign γ to agents uniformly from the interval $[0, 1]$ with granularity $m = 10$, thus for all agents $\gamma_i \in \{0.0, 0.1, \ldots, 0.9, 1.0\}$. These values are generated by PRNG G_1, initialized by seed I, which also generates the positions of the agents on the grid. All simulations and iteration have seed $I = 1245$.

The sensitivity parameters ($k_S,\ k_D,\ k_O$) are also input parameters. These parameters are uniformly assigned to all agents (if not stated other), and they affect their behavior, for example their solution of conflicts. Both the aggressivity γ and sensitivity parameters can be assigned uniformly to agents in groups as can be found in simulation in Section 4.6.

The friction parameter μ is a global parameter of the model that affects the stochastic occurrence of blocking occasions from conflicts. All simulations in this thesis use the new *strategy B* for *choosing destination cell*, and the unchanged *strategy A* for *solving conflicts*, because *strategy B* was ruled out, as described in Section 2.1.2 and in Section 4.6.

3.1.1 Numeric intervals

In Equation (2.2) the sensitivity to occupancy k_O, the sensitivity to diagonal movement k_D, and in conflict solution, the friction parameter μ, can be set to values from interval $[0, 1]$. This interval can be studied using discrete values, for example $\{0.0, 0.2, \ldots, 1.0\}$, that cover the range. Contrary to this, the sensitivity to the *static field* k_S can be any positive real number and requires different approach as will be described later.

3.2 Observable quantities

Stochastic selection of destination cells and pseudo-randomness in solving conflicts presents a possibility to measure various values of agents and model statistically. Simulation of evacuation can be described as a sequence of epochs (or steps, that consist of four phases) in which agents move in the room or leave the room.

TOTAL EVACUATION TIME

Total evacuation time T_{TET} is a time span from the start of evacuation (epoch s_1) to the moment when last agent leaves the room (epoch s_k). Each epoch takes t time [39].

$$T_{\text{TET}} = k \cdot t \tag{3.1}$$

Agents start from different positions, so it is possible to measure speed v_i of agent a_i:

$$v_i = \frac{\sigma_i}{\Delta t_i} \tag{3.2}$$

σ_i is the number of movements of a_i, and Δt_i is the time span from the start of simulation to a moment of a_i leaving the room.

Average speed $\bar{v}$ of all agents $a_i \in A$:

$$\bar{v} = \frac{\sum_{a_i \in A} v_i}{|A|} \tag{3.3}$$

FLOW

Flow is the number of agents evacuating the room at a given time span. Total flow J_{total} represents the flow of agents from the start of evacuation to the end. Time span of J_{total} then depends on n which is the number of all agents and T which is the duration of simulation.

$$J_{\text{total}} = \frac{n}{T} \tag{3.4}$$

The particular flow of agents with m agents that left the room in time span from t_i to t_j:

$$\begin{aligned} \Delta t &= t_j - t_i \\ J_{ij} &= \frac{m}{\Delta t} \end{aligned} \tag{3.5}$$

3.3 Methods of analysis

Sensitivity analysis of input parameters on total evacuation time is the main contribution of this thesis. SA was performed on OptiSLang software developed by Dynardo GmbH in Germany. The graphs — 3D and COP — exported from OptiSLang have the information of their origin in the caption. Additional research, that started the moment when first versions of the evacuation model were tested and the behavior of the model was observed, consists of several quantitative and qualitative analyses. On one hand, quantitative analysis presents it's results in numbers and graphs, that root in data from simulations. On the other hand, qualitative analysis focuses on understanding the qualities of the model, on the connections between parameters and it also focuses on explaining the preconditions that lead to specific phenomena. Both domains, quantitative and qualitative, are aided by visuals created in Matplotlib or in Mesa.

3.3.1 SA using MOP

To measure the contribution of individual input parameters on the variance in T_{TET}, OptiSLang uses Metamodel of Optimal Prognosis (MOP). The metamodel includes Polynomial, MLS or Krigins model. At first, OptiSLang calculates the prognosis quality of each available model, (Polynomial, MLS, Kriging) using Coefficient of Prognosis (COP) and then chooses the one with the highest quality. Measuring the quality of each model takes a long time, hence some simulations were allowed to use only polynomial and MLS model, as these two models produce adequate results while keeping the computation time reasonable.

As a result of the MOP, we obtain an approximation model, which includes the important variables. Based on this meta-model, the total effect sensitivity indices, . . . , are used to quantify the variable importance. The variance contribution of a single input variable is quantified by the product of the CoP and the total effect sensitivity index estimated from the approximation model [40]. COP(X_i) is variance contribution of single input variable X_i, that shows how much X_i contributed to the approximated variance using MOP. When MOP has COP 60%, it means that it was able to capture 60% of the variance in observed quantity. Some parameters can be related to each other and the sum of individual parameters COP(X_i) can exceed total COP or even 100%. For example, in MOP with COP 60%, parameters X_1 and X_2 with COP(X_2) = 50% and COP(X_2) = 20% individually contributed with 50%, respectively with 20%, to variance in observed quantity.

3.4 Simulations

At first, the simulations were observed on the Mesa framework web interface, that is depicted in Figure 2.6. They were used mostly during implementation of the model and in testing the new strategies.

Later, batch simulations on wide range of intervals were performed. All simulations had the same model configuration of grid 15 × 15, exit placed at border $(0, 8)$ and same seed $I = 1245$ for PRNG G_1, which assigns individual values of aggressivity γ to agents and places them in the grid hence initial conditions for all simulations and iterations were identical. Each iteration had different seed J for PRNG G_2, that solves conflicts and selects destination cells. With different J seeds, the course of the simulation varies and can be generalized.

The parameter k_S can be any positive real number. The first simulations, and later SA of k_S on simulation S_2, showed that T_{TET} does not change significantly for values higher than 5, hence this value was set as the upper range of k_S. The reason for setting the lower limit to 1.5 is extensively explained in Section 4.2. The movement of agents with k_S lower than 1.5 is erratic.

In the Table 3.1 are input parameters of simulations S_2, S_3, S_4 for SA in OptiSLang. Other analyses have configurations described in the text.

		Simulation name		
Parameters	**Step**	S_2	S_3	S_4
iterations	-	3	2	2
k_S	0.1	[0.3, 5.0]	[1.5, 4.5]	{1.5, 3.0, 4.5}
k_O	0.1	[0.0, 1.0]	[0.0, 1.0]	[0.0, 1.0]
k_D	0.2	[0.0, 0.1]	[0.0, 1.0]	[0.0, 1.0]
μ	0.2	[0.0, 0.1]	[0.0, 1.0]	[0.0, 1.0]

Table 3.1: Settings for simulations in SA.

In the 3D and COP graphs exported from OptiSLang, the axes are renamed, for clarity, with capitals S, O, D, F, T, and mean $S = k_S$, $O = k_O$, $D = k_D$, $F = \mu$ and $T = T_{\text{TET}}$. 3D graphs approximate the T_{TET} with data from simulations and plot them on a plane. COP graphs shows how much individual parameters S, O, D, F contribute to the variance in T_{TET}. The number in percent at the top of COP (for example on the right in Figure 4.2) shows how much variance in T_{TET} was MOP able to capture.

Chapter **4**

Results

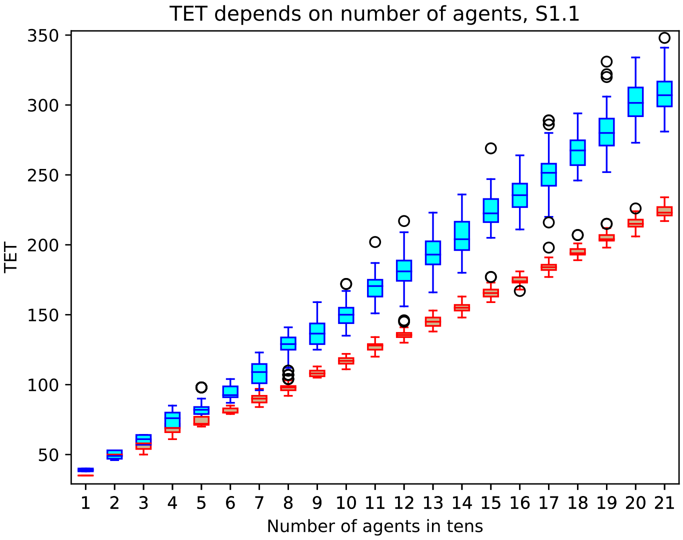
The results of SA in this chapter are explained by graphs from OptiSLang: the 3D plot of T_{TET} and COP of individual parameters. The discoveries of qualitative analysis, such as linear dependency of T_{TET} on number of agents n, are depicted by other graphs, for example histograms or boxplots.

4.1 Number of agents n

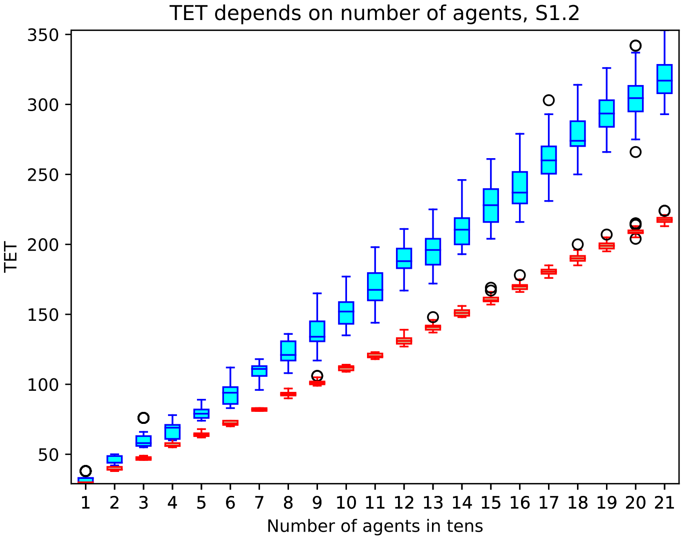
It was expected, that the increase of total evacuation time T_{TET} is linear and depends on the number of agents n. Figure 4.1 demonstrates the linear dependency of T_{TET} on n.

The sensitivity parameters for blue boxplots are $k_O = 0.9,\ k_D = 0.5,\ \mu = 0.9$ and for red boxplots $k_O = 0.9,\ k_D = 0.5,\ \mu = 0.1$. Each simulation was repeated 30 times with different seeds J for G_2. Horizontal axis x is the number of agents n in tens and vertical axis y is total evacuation time T_{TET}. The boxplots show the variance in T_{TET} for each n. There are two graphs with different $k_S \in \{1.5, 3.5\}$.

In both figures, the simulations with higher friction $\mu = 0.9$ show increasing variance of T_{TET}, because with more agents the conflicts occur more frequently and this results in more blocking occasions. Contrary to this, the boxplots in red, where μ is low, are rather consistent. These figures also demonstrate linear dependency of T_{TET} on number of agents. The slope of linear dependency is influenced by other parameters, just as the variance in T_{TET}.



(a) $k_S = 1.5$



(b) $k_S = 3.5$

Figure 4.1: Linear dependency of T_{TET} on number of agents. Higher friction, blue boxplots, increases variance of T_{TET}. Two graphs with different k_S show linear dependency of total evacuation time on n, regardless of k_S.

4.2 Sensitivity to static field k_S

The probability of choosing destination cell y is calculated from members P_O and P_S in the new strategy B for choosing destination cell. Notice P_S member in Equation 2.4, which is replicated, for clarity, below:

$$P_S(y) = \frac{\exp(-k_S S(y))(1 - k_D D(y))}{\sum_{z \in N} \exp(-k_S S(z))(1 - k_D D(z))}$$

For better demonstration, the member $(1 - k_D D(y))$ is left out from the edited equation $\bar{P_S}$, where y is destination cell from neighborhood N for agent A in cell x.

$$\begin{aligned} \bar{P_S}(y) &= \frac{\exp\left(-k_S S(y)\right) \cdot \exp\left(k_S S(x)\right)}{\sum_{z \in N} exp(-k_S S(z)) \cdot \exp\left(k_S S(x)\right)} \\ &= \frac{\exp\left(-k_S (S(y) - S(x))\right)}{\sum_{z \in N} exp(-k_S (S(z) - S(x)))} \end{aligned} \tag{4.1}$$

The equation above leads to discovery of member $\bar{P_S}$ being proportional to exponential difference in static potential:

$$\bar{P_S}(y) \propto \exp\left(-k_S (S(y) - S(x))\right) \tag{4.2}$$

For cell y, which is closest to exit and has lowest $S(y)$ from adjacent cells, the relative attractivity (the right hand side of Equation (4.2)) grows exponentially with increasing k_S.

For example, Figure 2.4 shows agent A in cell y_5 (cell will be noted x). A calculates P_S of cells y_7 and y_1, where $S(y_7) < S(x) < S(y_1)$. As k_S increases, $P_S(y_7)$ of cell closer to exit rapidly approaches 1. Opposite to this, $P_S(y_1)$ of cell with higher static potential decreases to zero with increasing k_S.

The influence of k_S can be very strong for high numbers, and marginal for values lower than 1. It was noticed during the testing phases of the model implementation, when the simulations with very low k_S values lasted very long. Visual examination of simulation exposed the erratic movement of agents that did not progress to the exit.

Figure 4.2 shows the 3D plot of approximated T_{TET}, from simulation S_2 in Table 3.1, exported from OptiSLang. The black dots are the values of T_{TET} from the simulations. *MLS* model was selectedy by MOP. Because of long computational time, *Krigin* model was not allowed. Vertical axis T is the T_{TET}, on horizontal axis F are values of friction μ and horizontal axis S holds the values of sensitivity to static field k_S. This graph shows very high T_{TET} for $k_S < 1.5$.

In the same figure, on the right, is a graph with COP of individual parameters. The COP(k_S) is prevalent and other parameters have marginal values of COP. This is because of undesired erratic movement of agents, that increases T_{TET} and it's variance, when k_S is lower than 1.5. In Figure 4.4 can be spotted how T_{TET} does not change much for values higher than 4.5. Because of these reasons, the interval of k_S was limited to $[1.5, 4.5]$ in further simulations.

In Figure 4.2, the influence of μ on axis F might seem marginal, but, as will be presented later, the friction parameter μ plays an important role in the course of the evacuation.

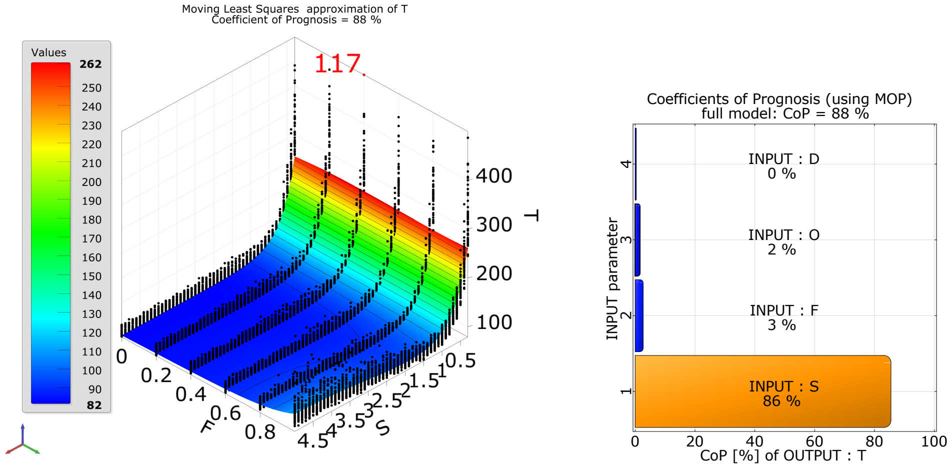


Figure 4.2: Left: 3D graph of total evacuation time, exported from OptiSLang. Right: COP of input parameters, exported from OptiSLang. Data from simulation S_2. For $k_S < 1.5$ the total evacuation time is erratic and meaningless, which lead to limit of the range of k_S to $[1.5, 4.5]$.

In simulations with $k_S \in [1.5, 4.5]$, other parameters influence the simulation more significantly, as can be seen in Figure 4.3 in COP graph on the right. According to COP, the most significant parameter is friction μ, that contributed with 48% to variance in observed quantity T_{TET}. Formerly, the most significant parameter was k_S, with COP(k_S) higher than 86%. Now, COP(k_S) dropped to 6%, on par with COP(k_D). COP of MOP on the top of the graph, is lower as well: was 88% and now is 67%. This can be explained with very high variance in T_{TET} for $k_S < 1.5$, which was attributed to k_S. In simulations with limited k_S, the variance in T_{TET} is lower.

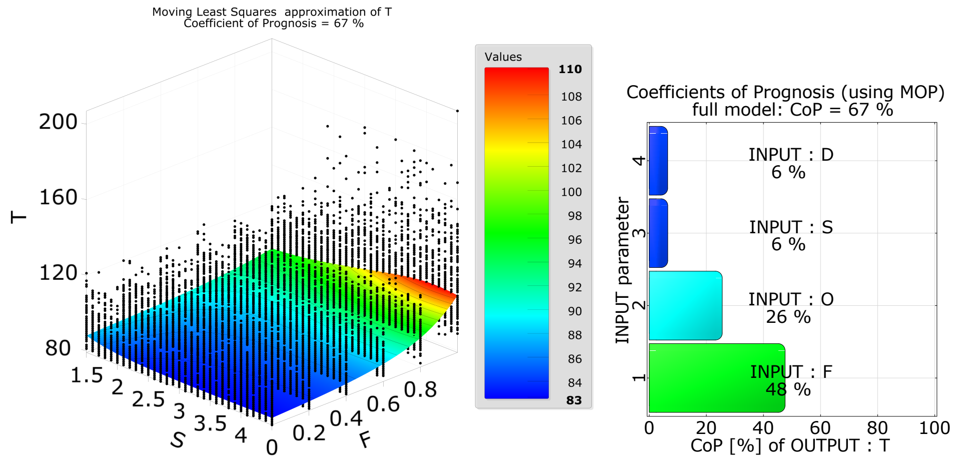


Figure 4.3: Left: 3D graph of T_{TET} from OptiSLang. Right: COP of input parameters, exported from OptiSLang. Data from simulation S_3. With limited range of k_S, the COP of individual parameters, on the right, is well distributed. *Note: The angle of view is different compared to Figure 4.2, axes F and S are switched.*

SA of simulations S_2 and S_3 was performed using *MLS* model for MOP. *Krigin* model was disabled because of high computational time. Further SA on simulations S_4 use *Krigin* model.

4.3 Sensitivity to occupancy k_O

Lower values of k_O allow the agent to choose an occupied cell as his target cell. At the start of the simulation, the agents are densely packed. The same applies to the congestion, when more agents are approaching the exit (or bottleneck in general), than are being egressed. In both situations, k_O plays a role, as agents near the exit, and agents on the border of the congestion cluster, can choose empty or occupied cells.

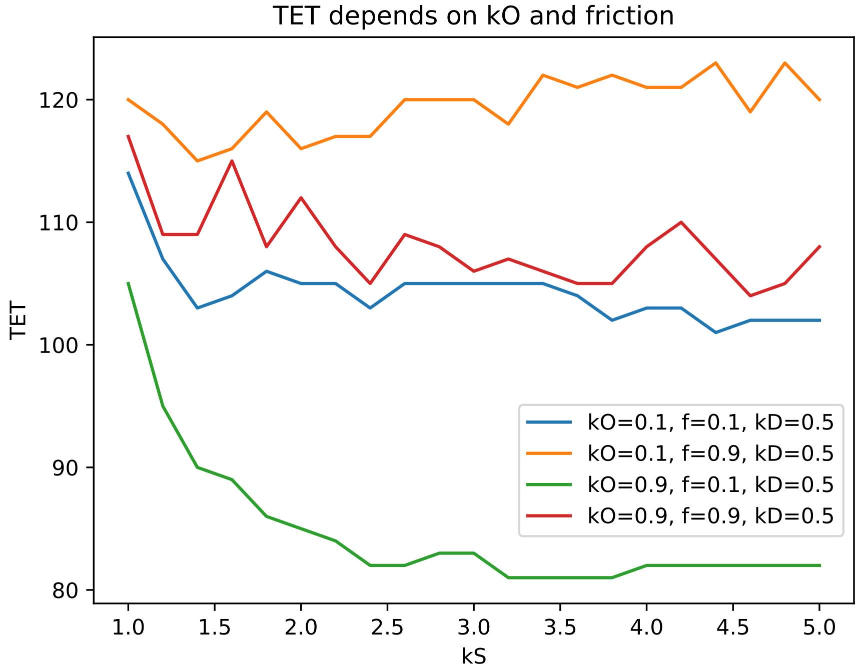


Figure 4.4: Different sets of parameters k_O, μ and how they affect T_{TET}. Vertical axis is averaged T_{TET} from three simulations and horizontal axis is variable k_S. Lower k_O and higher μ increase evacuation time.

The agents inside the cluster seldom move. Even if they choose an occupied cell, when $k_0 \neq 1$, they can't enter the cell, because in dense cluster the occupant of said cell can't move neither. With high k_O, agents prefer to stay in their cell, when adjacent cells are occupied. At the start of the evacuation, with high k_O, movement waves can be spotted, as is depicted in Figure B.1 in Appendix.

Counter to this, low k_O can create *queue* structures. Transition probabilities, influenced by the *strategy for choosing destination cell* and low k_O, are much higher for cells closer to exit — cells with lower *static* field values.

Agents are prone to ignore the empty cells on the sides of the queue and they prefer to stay in the queue. Even though the movement inside the queue is rather consistent, the total evacuation time increases. When a queue is formed, two clusters are formed as well, one at both ends of the queue. This can be noticed in Figure B.1 in Appendix. One cluster forms at the exit, and because two or more agents attempt to enter the exit at the same time, blocking occasions might occur and total evacuation time will be increased. The cluster at the other end fills the queue and the agents inside it are mostly stationary. If more agents were approaching the exit simultaneously, the total evacuation time would be lower.

Figure 4.4 shows the characteristics of four constant parameters sets with variable $k_S \in [1.0, 5.0]$ on *x axis*. Each set of parameters was simulated 100 times and the vertical *y axis* is the averaged T_{TET} from these simulations. Even though the lines are not very smooth, it needs to be said that the resolution of T_{TET} is high — T_{TET} is in range from 80 to 125. All sets have the same $k_D = 0.5$.

The green line shows how parameters $k_O = 0.9, \mu = 0.9$ are affected by k_S. The relative position of the green line, compared to others, shows that by allowing agents to overtake queue (high k_O) and with low number of conflicts (low μ), the evacuation times are the shortest (81 to 105 epochs). Contrary to this, with same k_O and higher $\mu = 0.9$, the evacuations take longer (105 to 117 epochs) and the results for various k_S vary a lot due to more frequent conflicts. The remaining two sets have low $k_0 = 0.1$, which minimizes overtaking, and forces the agents to stay in *queues.*This results in longer evacuations. The higher $\mu = 0.9$ in the orange line is rough. It should be noticed, that with increasing k_S, the evacuations take longer than in the beginning (115 epochs for low k_S and 125 epochs for higher k_S). In the end the total evacuation time is increased with lower k_O and it also increases with higher μ.

4.4 Sensitivity to diagonal movement k_D

Agent that moves to adjacent diagonal cells his time, when movement ends, by $\frac{3}{2}$ of nominal movement duration. At the start of the simulation, agents closest to the exit have a free path, and move rapidly towards exit. Movement in diagonal directions allows agents, that are not in a straight line from the *exit*, to get there with less movements. However, they are penalized by increased movement duration — after two diagonal movements the agents are *exhausted* and do not move. Especially at the start of the simulation, with low k_D, agents can move in waves, as can be seen in Figure B.1 in Appendix B.1. Diagonal movement is also useful when a queue or other congestion structure forms. The agents can overtake the queue when they move diagonally.

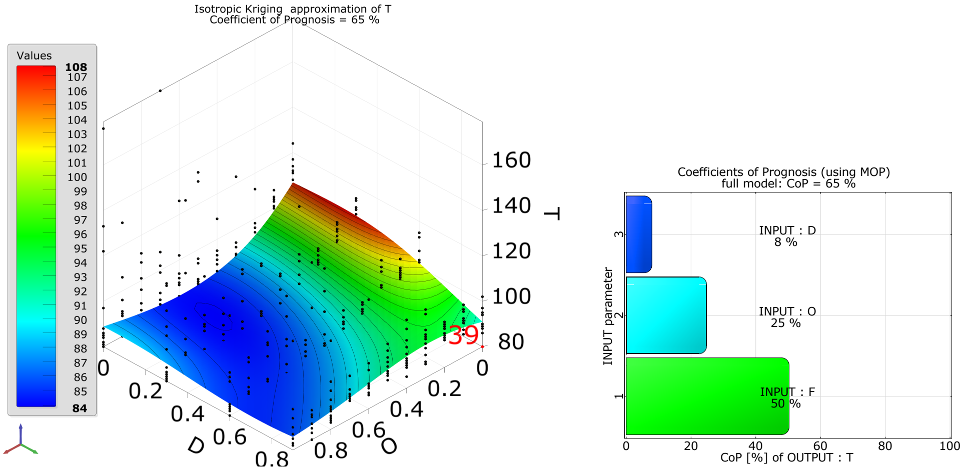


Figure 4.5: Left: 3D graph of T_{TET} from OptiSLang. Right: COP of input parameters, exported from OptiSLang. Data from simulation $S_4, k_S = 3.0$. Compared to other parameters, influence of k_D is marginal. *Note: MOP selected Krigin model.*

Even though the microscopic behavior of agents can be significantly affected by k_D, the general effect on the T_{TET} is marginal. Graph of COP in Figure 4.3 shows that COP(k_D) is very low (6%) with $k_S \in [1.5, 4.5]$. Figure 4.5 presents 3D graph and COP, exported from OptiSLang, of simulation S_4 with fixed $k_S = 3.0$, which is in the middle of k_S interval described before. It can be seen, that COP(k_D) is low and the prevalent input parameter is friction μ, with COP$(\mu) = 48\%$. The COP of MOP (65%) is similar to SA of simulation S_4 with other other values $k_S \in \{1.5, 4.5\}$. The quality of approximation is 65%, yet the sum of individual COP of input parameters exceeds this value.

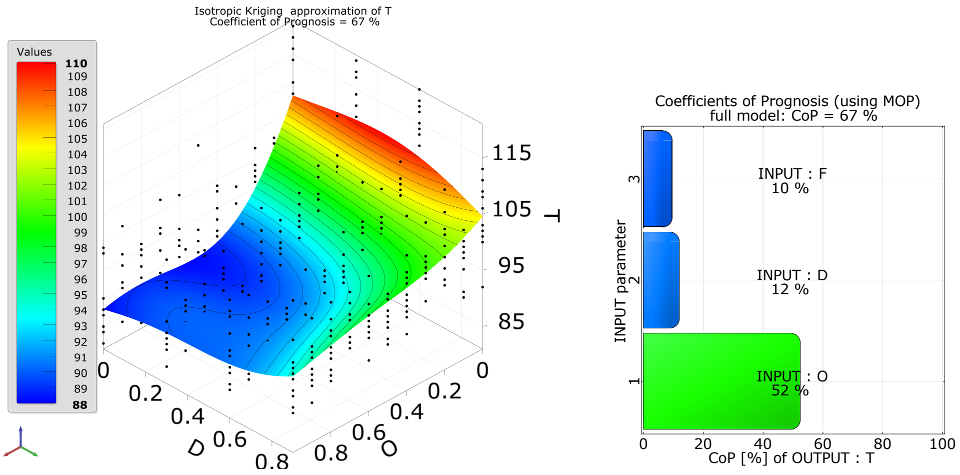


Figure 4.6: Left: 3D graph of T_{TET} from OptiSLang. Right: COP of input parameters, exported from OptiSLang. Data from simulation S_4, $k_S = 1.5$. Agents calculate transition probability to adjacent cells more evenly and are more sensitive to input parameters. *Note: MOP selected Krigin model.*

4.5 Friction parameter μ

As was depicted in previous Figures 4.6 and 4.5, μ and k_O have significant contribution to variance in T_{TET}. Friction parameter μ affects the number of blocking occasions and that can increase the total evaction time. With higher number of agents n, the blocking occasions occur more frequently, because more agents create more conflicts. This results in increased variance in T_{TET}, as is demonstrated in Figure 4.1. The blue box-plots are simulations with high $\mu = 0.9$ and show increasing variance of T_{TET}. Contrary to this, the red box-plots, simulations with low $\mu = 0.1$, show rather constant variance of T_{TET} with increasing n of agents.

For high $k_S = 4.5$ the COP(X_i) of other parameters (k_O, k_D) is minimal due to prevalent *static* sensitivity. However, μ has significant contribution to variance in T_{TET} as can be seen in Figure 4.7. When $k_S = 1.5$, COP(μ) is the lowest (10%) and increases with k_S: for $k_S = 1.5$, COP(μ) is 10%, then with $k_S = 3.0$, COP(μ) is 50%, and later, with $k_S = 4.5$, COP(μ) is highest at 59%. SA of input parameters on T_{TET} for several constant values of k_S shows, that for different k_S, COP(X_i) for individual input parameters are in different order. For lower values of k_S, k_D and k_O have higher COP than μ. With high k_S, μ has highest COP(μ) and other parameters have rather low COP(X_i).

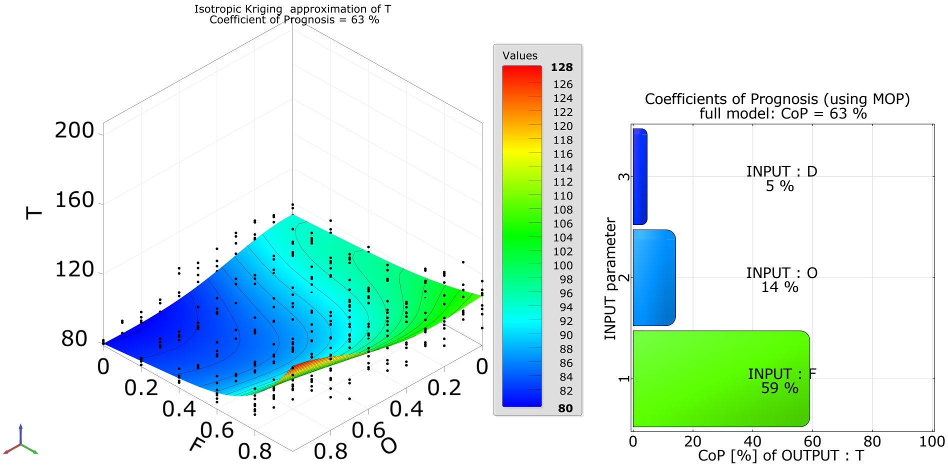


Figure 4.7: Left: 3D graph of T_{TET} from OptiSLang. Right: COP of input parameters, exported from OptiSLang. Data from simulation $S_4, k_S = 4.5$. Friction μ has major influence while COP of k_O, k_D are minimized in response to high k_S. *Note: MOP selected Krigin model.*

4.6 Other discovery: heterogeneity in parameters

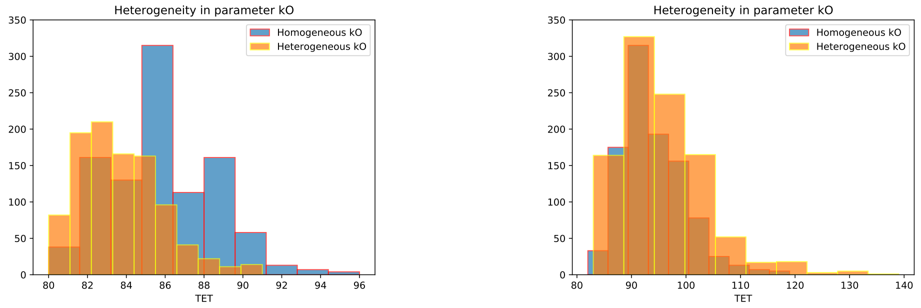
The evacuation model in this thesis can assign different parameters to individual agents from a uniform distribution, or assign different parameters to groups of agents. Values are generated by PRNG G_2 and are subject to stochastic selection. To see how the model responds to homogeneous and heterogeneous parameters, simulations $S_{6.1}$ and $S_{6.2}$ were performed. In both simulations grid 15×15 was used, with *exit* placed at $(0, 8)$ and populated with 70 agents. Simulations were repeated 1000 times,

First simulation $S_{6.1}$ with set of input parameters $k_S = 2.0,\ k_D = 0.5$ and low $\mu = 0.1$. This simulation was, at first, run with *homogeneous* $k_O = 0.5$ for all agents. The blue histogram in Figure 4.8a shows distribution of T_{TET} from 1000 iterations of this simulation. Most iterations, more than 300, resulted in $T_{\mathrm{TET}} \approx 85$ and all iterations had $T_{\mathrm{TET}} \in [80, 96]$.

This simulation was then rerun, also repeated 1000 times, with *heterogeneous* distribution of k_O. The 70 agents were split to two groups of 35. First groups was assigned $k_O = 0.1$ and second group $k_0 = 0.9$, so the average k_O of all agents was 0.5, identical to the previous run. The orange histogram in Figure 4.8a shows that this simulation resulted in shorter evacuations — all iteration had $T_{\mathrm{TET}} \in [80, 91]$.

Another simulation $S_{6.2}$ was performed, this time with increased friction $\mu = 0.9$, other parameters remained unchanged $k_S = 2.0, k_D = 0.5$. Identical to $S_{6.1}$, simulation was repeated 1000 times — once for *homogeneous* and then for *heterogeneous* k_O of agents. Figure 4.8b shows that distribution of T_{TET}

for *homogeneous* k_O, blue histogram, is similar to *heterogenenous* k_O for two groups of agents, orange histogram.



(a) Simulation $S_{6.1}$, low friction $\mu = 0.1$.

(b) Simulation $S_{6.2}$, high friction $\mu = 0.9$

Figure 4.8: Simulations with heterogeneous and homogeneous distribution of parameter k_O, repeated 1000 times. Blue histograms show T_{TET} for homogeneous $k_O = 0.5$. Simulations with two groups of agents with with different $k_O = 0.1$ and $k_O = 0.9$ are captured in orange histograms.

Conclusions

This thesis aimed to perform research of evacuation simulation models and to analyze the process of conflict solution. The *floor-field* model, which was an inspiration for model used in this thesis, uses two cellular fields: *dynamic* and *static* field. The model has a set of input parameters, that influence the behavior of agents in the model and the microscopic processes within (conflict solution, choosing destination cell).

To perform said research on conflict solution and the influence of input parameters, an agent based cellular evacuation model was implemented using Mesa ABM framework for Python. Using agents, the model is able to capture the collective motion of pedestrians during an evacuation. Individual agents are autonomous and interact with each other. These interactions, mainly the *conflicts*, are analyzed. The model consists of various components, that can be further extended or customized.

During the implementation of the model, after observing the evacuation process, a new strategy for choosing destination cell was proposed and implemented. It affects the movement of agents and thus significantly impacts the conflicts that emerge from these movements. Also, this strategy puts more attention to the input parameter of sensitivity to occupation k_O, which produces more evenly distributed probabilities of moving to adjacent cells and thus k_O can be interpreted more straightforwardly.

Additional outcome of the observations was a new strategy for solving conflicts, that intended to prevent jamming occasions without lowering the friction parameter μ. These occur when agents with identical low aggressivity get into conflict — the probability of one of the agents winning the conflict is low. After successful implementation and initial simulations and tests, this strategy was abandoned as it did not bring expected results — the jamming occasion are still present.

The model uses a stochastic selection for several processes in the simulation. It is a desirable feature as it replicates the randomness in human behavior. Due to variance in observed quantities (total evacuation time) with the same input parameters, it was necessary to analyze the contribution of individual input parameters to the variance in T_{TET}. This was achieved by employing sensitivity analysis (SA) in OptiSLang software and by other auxiliary methods.

The results of SA show the contribution of individual input parameters to the variance in total evacuation time. One concrete contribution of this thesis is connected to sensitivity to *static* potential k_S. SA of k_S revealed, that low values of $k_S < 1$ produce erratic movement of agents, that degrades the course of the simulation. On the other end, with increasing k_S, for example $k_S > 4.5$, the influence of other sensitivity parameters $k_O,\ k_D$ is marginalized.

Friction parameter μ showed consistent influence on the variance in T_{TET}. This parameter directly affects the number of blocking occasions, which increase the total evacuation time.

The sensitivity to occupancy k_O allows agent to choose an occupied cell as his destination cell. The formation of certain structures (queues and congestion cones) during the evacuation was linked to k_O. These structures, when present, can be connected to variance in T_{TET}. Qualitative analysis of k_O also revealed, that parameters k_O and sensitivity to diagonal motion k_D affect each other, as k_D allows agents to overtake queues, which are formed with low k_O. Apart from this, SA showed, that the contribution of k_D to variance in T_{TET} is marginal.

Other notable discoveries consist in analysis of number of agents n, and of heterogeneity in parameters. Numerous simulations with variable n showed that T_{TET} is linearly dependent on n.

Other simulations examined the course of the evacuation with different distribution of k_O of agents. In simulations where all agents are uniformly assigned one value of $k_O = a$, total evacuation times are different than in simulations, where two different values of k_O are assigned to equally sized groups and the average k_O of agents in both groups is a. This discovery will be further researched.

Bibliography

[1] Lee, M.; Lee, J.; et al. An extended floor field model considering the spread of fire and detour behavior. *Physica A: Statistical Mechanics and its Applications*, 2021: p. 126069, ISSN 0378-4371, doi: https://doi.org/10.1016/j.physa.2021.126069. Available from: `https://www.sciencedirect.com/science/article/pii/S0378437121003423`

[2] Hrabák, P.; Bukáček, M. Conflict Solution According to "Aggressiveness" of Agents in Floor-Field-Based Model. In *Parallel Processing and Applied Mathematics*, edited by R. Wyrzykowski; E. Deelman; J. Dongarra; K. Karczewski; J. Kitowski; K. Wiatr, Cham: Springer International Publishing, 2016, ISBN 978-3-319-32152-3, pp. 507–516.

[3] Kleinmeier, B.; Köster, G.; et al. Agent-based simulation of collective cooperation: from experiment to model. *Journal of The Royal Society Interface*, volume 17, 10 2020: p. 20200396, doi:10.1098/rsif.2020.0396.

[4] Schadschneider, A.; Klingsch, W.; et al. *Evacuation Dynamics: Empirical Results, Modeling and Applications.* New York, NY: Springer New York, 2009, ISBN 978-0-387-30440-3, pp. 3142–3176, doi:10.1007/978-0-387-30440-3_187. Available from: `https://doi.org/10.1007/978-0-387-30440-3_187`

[5] Mitsopoulou, M.; Dourvas, N.; et al. Spatial Games and Memory Effects on Crowd Evacuation Behavior with Cellular Automata. *Journal of Computational Science*, volume 32, 09 2018, doi:10.1016/j.jocs.2018.09.003.

[6] Yuan, Y.; Goñi-Ros, B.; et al. Macroscopic pedestrian flow simulation using Smoothed Particle Hydrodynamics (SPH). *Transportation Research Part C: Emerging Technologies*, volume 111, 2020: pp. 334–351, ISSN 0968-090X, doi:https://doi.org/10.1016/j.trc.2019.12.017. Available from: `https://www.sciencedirect.com/science/article/pii/S0968090X19310125`

[7] K. Hirai, K. T. A simulation of the behavior of a crowd in panic. In *Proceedings of the 1975 International Conference on Cybernetics and Society*, 1975, pp. 409–411.

[8] Zhao, X.; Lovreglio, R.; et al. Using Artificial Intelligence for Safe and Effective Wildfire Evacuations. *Fire Technology*, 2020: pp. 1–3.

[9] Branke, J. *Artificial Societies*. Boston, MA: Springer US, 2010, ISBN 978-0-387-30164-8, pp. 44–48, doi:10.1007/978-0-387-30164-8_36. Available from: https://doi.org/10.1007/978-0-387-30164-8_36

[10] Kretz, T.; Schreckenberg, M. The F.A.S.T.-Model. *Cellular Automata*, 2006: p. 712–715, ISSN 1611-3349, doi:10.1007/11861201_85. Available from: http://dx.doi.org/10.1007/11861201_85

[11] Murphy, S. O.; Brown, K. N.; et al. The EvacSim Pedestrian Evacuation Agent Model: Development and Validation. In *Proceedings of the 2013 Summer Computer Simulation Conference*, SCSC '13, Vista, CA: Society for Modeling; Simulation International, 2013, ISBN 9781627482769.

[12] Thornton, C.; O'Konski, R.; et al. Pathfinder: An Agent-Based Egress Simulator. In *Pedestrian and Evacuation Dynamics*, edited by R. D. Peacock; E. D. Kuligowski; J. D. Averill, Boston, MA: Springer US, 2011, ISBN 978-1-4419-9725-8, pp. 889–892.

[13] Ronchi, E.; Colonna, P.; et al. The evaluation of different evacuation models for assessing road tunnel safety analysis. *Tunnelling and Underground Space Technology*, volume 30, 2012: pp. 74–84, ISSN 0886-7798, doi:https://doi.org/10.1016/j.tust.2012.02.008. Available from: https://www.sciencedirect.com/science/article/pii/S0886779812000405

[14] Colombi, A.; Scianna, M. Modelling human perception processes in pedestrian dynamics: a hybrid approach. *Royal Society Open Science*, volume 4, no. 3, 2017: p. 160561, doi:10.1098/rsos.160561, https://royalsocietypublishing.org/doi/pdf/10.1098/rsos.160561. Available from: https://royalsocietypublishing.org/doi/abs/10.1098/rsos.160561

[15] Chraibi, M.; Zhang, J. JuPedSim: an open framework for simulating and analyzing the dynamics of pedestrians. In *SUMO2016 - Traffic, Mobility, and Logistics, Proceedings*, *Berichte aus dem DLR-Institut für Verkehrssystemtechnik*, volume 30, SUMO Conference 2016, Berlin (Germany), 23 May 2016 - 25 May 2016, Braunschweig: Deutsches Zentrum für Luft- und Raumfahrt e. V., Institut für Verkehrssystemtechnik, May 2016, pp. 127–134. Available from: https://juser.fz-juelich.de/record/809790

[16] Weidmann, U. Transporttechnik der fußgänger: transporttechnische eigenschaften des fußgängerverkehrs, literaturauswertung. *IVT Schriftenreihe*, volume 90, 1993.

[17] Selvek., R.; Surynek., P. Engineering Smart Behavior in Evacuation Planning using Local Cooperative Path Finding Algorithms and Agent-based Simulations. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KEOD,*, INSTICC, SciTePress, 2019, ISBN 978-989-758-382-7, ISSN 2184-3228, pp. 137–143, doi:10.5220/0008071501370143.

[18] Berto, F.; Tagliabue, J. Cellular Automata. In *The Stanford Encyclopedia of Philosophy*, edited by E. N. Zalta, Metaphysics Research Lab, Stanford University, spring 2021 edition, 2021.

[19] Wainwright, R. *Conway's Game of Life: Early Personal Recollections*. London: Springer London, 2010, ISBN 978-1-84996-217-9, pp. 11–16, doi:10.1007/978-1-84996-217-9_2. Available from: `https://doi.org/10.1007/978-1-84996-217-9_2`

[20] Wolfram, S. *A New Kind of Science*. Wolfram Media, 2002, ISBN 1579550088. Available from: `https://www.wolframscience.com`

[21] Nishinari, K.; Kirchner, A.; et al. Extended Floor Field CA Model for Evacuation Dynamics. *IEICE Transactions on Information and Systems*, volume E87-D, 07 2003.

[22] Ben-Jacob, E. From snowflake formation to growth of bacterial colonies. *Contemporary Physics*, volume 34, no. 5, 1993: pp. 247–273, doi:10.1080/00107519308222085, `https://doi.org/10.1080/00107519308222085`. Available from: `https://doi.org/10.1080/00107519308222085`

[23] Ross, J. L.; Ozbek, M. M.; et al. Aleatoric and epistemic uncertainty in groundwater flow and transport simulation. *Water Resources Research*, volume 45, no. 12, 2009, doi:https://doi.org/10.1029/2007WR006799, `https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2007WR006799`. Available from: `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007WR006799`

[24] Rahn, S.; Gödel, M.; et al. Dynamics of a Simulated Demonstration March: An Efficient Sensitivity Analysis. *Sustainability*, volume 13, no. 6, 2021, ISSN 2071-1050, doi:10.3390/su13063455. Available from: `https://www.mdpi.com/2071-1050/13/6/3455`

[25] Feliciani, C.; Nishinari, K. Measurement of congestion and intrinsic risk in pedestrian crowds. *Transportation Research Part C: Emerging Technologies*, volume 91, 2018: pp. 124–155, ISSN 0968-090X,

doi:https://doi.org/10.1016/j.trc.2018.03.027. Available from: `https://www.sciencedirect.com/science/article/pii/S0968090X18304133`

[26] Gödel, M.; Fischer, R.; et al. Sensitivity Analysis for Microscopic Crowd Simulation. *Algorithms*, volume 13, no. 7, 2020, ISSN 1999-4893, doi: 10.3390/a13070162. Available from: `https://www.mdpi.com/1999-4893/13/7/162`

[27] Kretz, T.; Schreckenberg, M. *F.A.S.T. - Floor field- and Agent-based Simulation Tool.* 10 2009, ISBN 978-1420095098, pp. 125–135, doi: 10.13140/2.1.3146.1444.

[28] Rogsch, C. The Influence of Moore and von-Neumann Neighbourhood on the Dynamics of Pedestrian Movement. In *Traffic and Granular Flow '15*, edited by V. L. Knoop; W. Daamen, Cham: Springer International Publishing, 2016, ISBN 978-3-319-33482-0, pp. 129–136.

[29] Bukáček, M.; Hrabák, P.; et al. Cellular Model of Pedestrian Dynamics with Adaptive Time Span. In *Parallel Processing and Applied Mathematics*, edited by R. Wyrzykowski; J. Dongarra; K. Karczewski; J. Waśniewski, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, ISBN 978-3-642-55195-6, pp. 669–678.

[30] Cordeiro de Amorim, R.; Mirkin, B. Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering. *Pattern Recognition*, volume 45, no. 3, 2012: pp. 1061–1075, ISSN 0031-3203, doi: https://doi.org/10.1016/j.patcog.2011.08.012. Available from: `https://www.sciencedirect.com/science/article/pii/S0031320311003517`

[31] Piatetsky, G. Python leads the 11 top Data Science, Machine-Learning platforms: Trends and Analysis. online, 2019. Available from: `https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html`

[32] Masad, D.; Kazil, J. Mesa: An Agent-Based Modeling Framework. In *14th PYTHON in Science Conference*, 01 2015, pp. 51–58, doi:10.25080/Majora-7b98e3ed-009.

[33] Tashakor, G.; Suppi, R. Agent-based model for tumour-analysis using Python+Mesa. *CoRR*, volume abs/1909.01885, 2019, 1909.01885. Available from: `http://arxiv.org/abs/1909.01885`

[34] Namany, S.; Govindan, R.; et al. Sustainable food security decision-making: An agent-based modelling approach. *Journal of Cleaner Production*, volume 255, 2020: p. 120296, ISSN 0959-6526, doi: https://doi.org/10.1016/j.jclepro.2020.120296. Available from: `https://www.sciencedirect.com/science/article/pii/S0959652620303437`

[35] Bloice, M. D.; Holzinger, A. *A Tutorial on Machine Learning and Data Science Tools with Python*. Cham: Springer International Publishing, 2016, ISBN 978-3-319-50478-0, pp. 435–480, doi:10.1007/978-3-319-50478-0_22. Available from: `https://doi.org/10.1007/978-3-319-50478-0_22`

[36] McKinney, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, Inc., second edition, 2017, ISBN 1491957662.

[37] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, volume 9, no. 3, 2007: pp. 90–95, doi:10.1109/MCSE.2007.55.

[38] Koeune, F. *Pseudo-random number generator*. Boston, MA: Springer US, 2005, ISBN 978-0-387-23483-0, pp. 485–487, doi:10.1007/0-387-23483-7_330. Available from: `https://doi.org/10.1007/0-387-23483-7_330`

[39] Hrabák, P.; Bukáček, M. Influence of agents heterogeneity in cellular model of evacuation. *Journal of Computational Science*, volume 21, 2017: pp. 486–493, ISSN 1877-7503, doi:https://doi.org/10.1016/j.jocs.2016.08.002. Available from: `https://www.sciencedirect.com/science/article/pii/S1877750316301259`

[40] Most, T.; Will, J. Sensitivity analysis using the Metamodel of Optimal Prognosis. In *Proceedings of 8th Weimar Optimization and Stochastic Days 2011*, 11 2011, p. 17p.

Appendix A

Acronyms

SA Sensitivity analysis

CA Cellular automaton

ABM Agent-based model

MOP Metamodel of Optimal Prognosis

COP Coefficient of Prognosis

PRNG Pseudo-random number generator

TET Total evacuation time

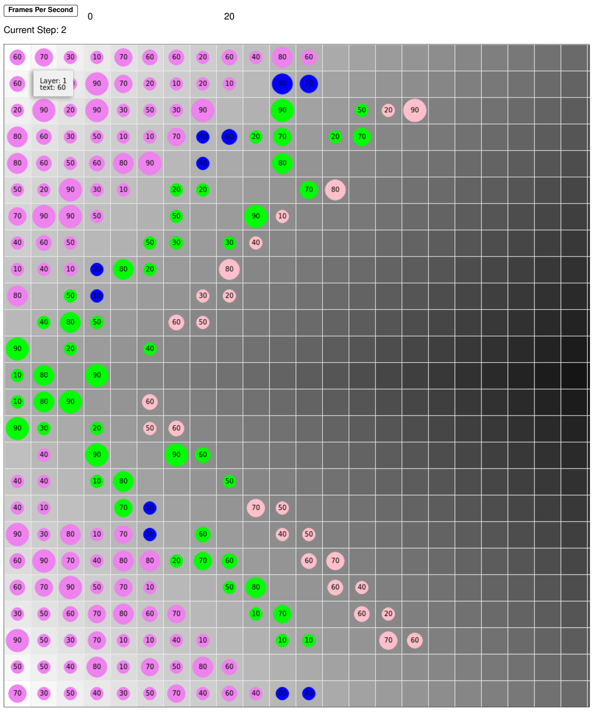
MLS Method of least squares

Appendix B

Further graphical output

B.1 Diagonal movement, waves

Mesa framework allows visualization of evacuation with individual agents, who have color based on their situation. The grid consists of rectangular cells, that have darker color when closer to the exit — according to the L_1 distance. Figure B.1 shows the formation of waves at the start of the evacuation. Agents are not allowed to select an occupied cell, $k_0 = 1$, and diagonal movement is allowed, $k_D = 0$. The first wave of agents, closest to the exit on the right in dark color area, have pink color, as they already advanced by two diagonal steps. Their inner time period was increased by $2 \cdot \frac{3}{2} = 3$ in step 2 thus they were not allowed to move in step 3.

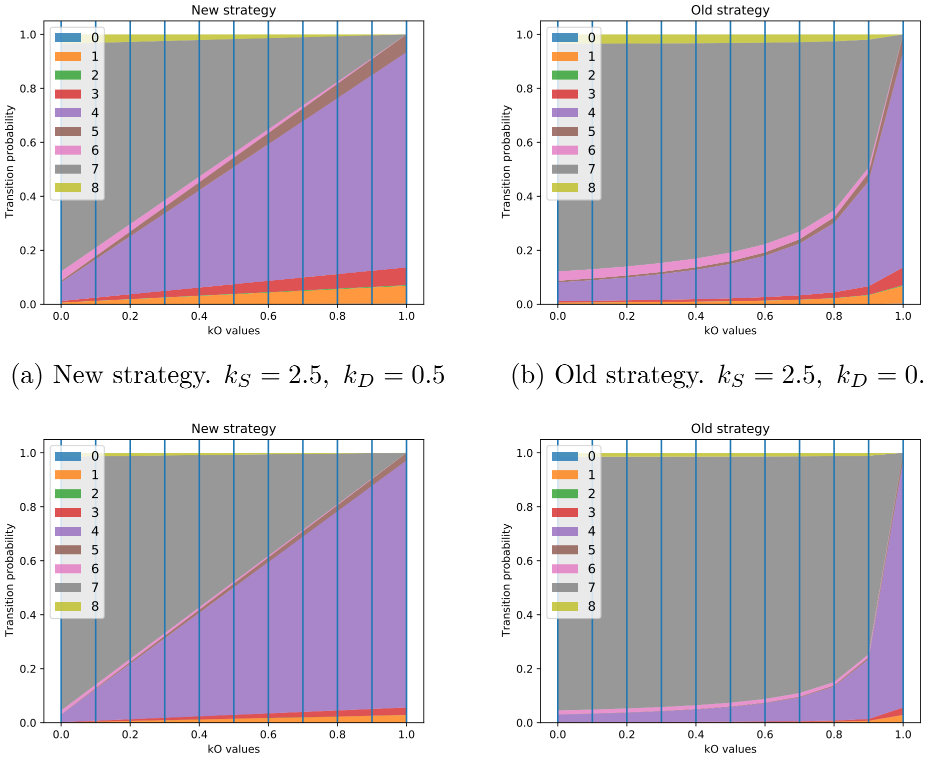


Current Step: 2

Figure B.1: Waves are forming at the start due to low $k_D = 0$ and high $k_O = 1$. Pink agents are penalized for repeated diagonal movement.

B.2 Strategies for choosing destination cell

In new proposed *strategy B for choosing destination cell*, parameter k_O influences the selection of destination cell, which is more even, compared to old **strategy A**. Graphs in Figure B.2 below are 100% stacked graphs of individual transition probabilities of adjacent cells in situation with blocked exit, see Figure 2.3. The horizontal axis is variable $k_0 \in \{0.0, 0.1, \ldots, 1.0\}$. The legend in the upper left corner assigns colors to cell numbers.



(a) New strategy. $k_S = 2.5,\ k_D = 0.5$

(b) Old strategy. $k_S = 2.5,\ k_D = 0.5$

(c) New strategy. $k_S = 3.5,\ k_D = 0.5$

(d) Old strategy. $k_S = 3.5,\ k_D = 0.5$

Figure B.2: Comparison of strategies for choosing destination cell. On the left are probabilities of adjacent cell calculated with new *strategy B*. On the right are calculated using the old *strategy A*. The new strategy distributes the probabilities more evenly.

B.3 Congestion structures

During the implementation of the evacuation model, certain patterns were noticed, namely the formation of *queues* and *cones* at the exit.

In Figures B.3 and B.4 are two visuals. The heatmap on the left shows the results from 100 iterations of simulation with grid 30×30, populated with 300 agents with exit placed at $(0, 15)$ and homogeneous input parameters $k_S = 2.5, k_D = 0.5, \mu = 0.7$. From all iterations, the iteration M with longest evacuation time $T_{\text{TET}} = m$ was selected. In epoch $\{1, 2, 3, \ldots, m\}$ each cell of the heatmap grid was assigned color based on number of iterations where agent was present in that cell. At the start of the evacuation, epoch 1, all iterations had agents in the same cells so they all had white color. Later individual iterations moved agents to different places, and some cells were occupied in more iterations than others. These frequent cells, where many iterations placed their agents, had brighter color. Contrary to this, cells never visited had black color. The colorbar on the right next to the heatmap shows the resolution of agents occupancy in cells. The brighter the color of a cell, the more agents were present in that cell in that epoch.

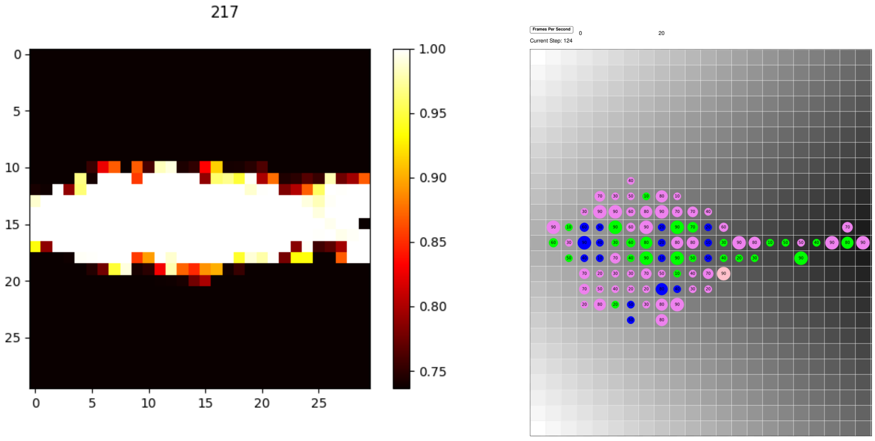


Figure B.3: Queue forms at the exit due to low k_O.

The simulations with low $k_O = 0.1$ produced heatmaps where kind of a *queue* can be seen in Figure B.3. One concrete occasion of this phenomena is on the right — visualization of one evacuation simulation in Mesa framework with identical parameters to simulation mentioned before.

In comparison, with high $k_O = 0.9$, the heatmap in Figure B.4 shows a formation of a cone structure at the exit, where agents are clustered and wait for evacuation. One the right is one instance of evacuation simulation in Mesa framework, parameters are identical to the simulation with heatmap.

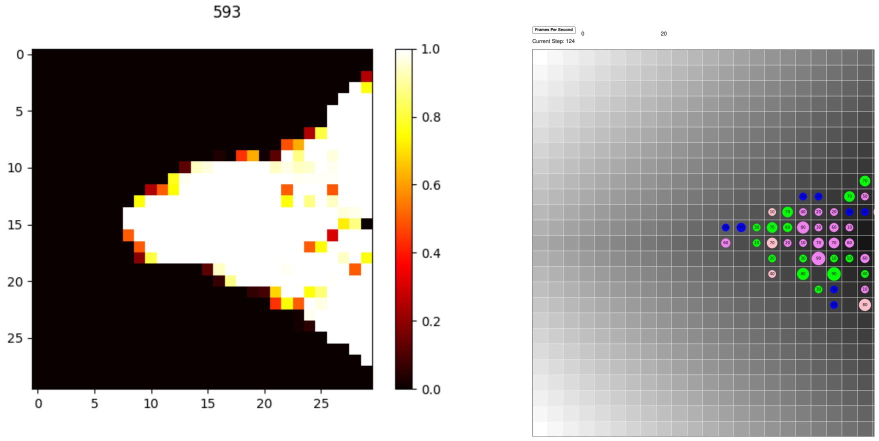


Figure B.4: Cone forms at the exit due to high k_O.

B.4 Flow analysis

Three joint graphs in Figures B.5 andB.6 show the course of the evacuation. The simulations were performed on a grid 15×15, with exit placed at $(0, 8)$, with 70 agents. Input parameters $k_S = 2.5$, $k_O = 0.5$, $k_D = 0.5$, $\mu = 0.7$, 100 iterations were executed.

First two joint graphs depict the start of the simulation, up to epoch 33. Below are two joint graphs that show the later stages of evacuation, from epoch 62 to 95.

In the title of the graph is the current epoch of evacuation. One of three graphs, the one in the upper left corner shows how many iterations were still running in that exact epoch. First drop of running iterations can be seen in the Figure B.6 on the right.

Following graphs is the heatmap of position of agents. The heatmaps are explained in Section B.3. The brighter the color of a cell, the more iterations placed agents in that cell.

The last graph of the three, is the average flow through the exit. The exit can be enter by one agent at a time so the upper limit of flow is 1. As can be seen in Figure B.5 on the left, at the start of the evacuation no agent are being evacuated. Later, on the left in Figure B.6, is the stationary flow of agents through the exit. This flow oscillates because in some iterations a blocking occasion occurred at exit cell. Later can be noticed, in the same graph on the right, how the average flow of 100 iterations decreases, when some iterations already finished evacuation.

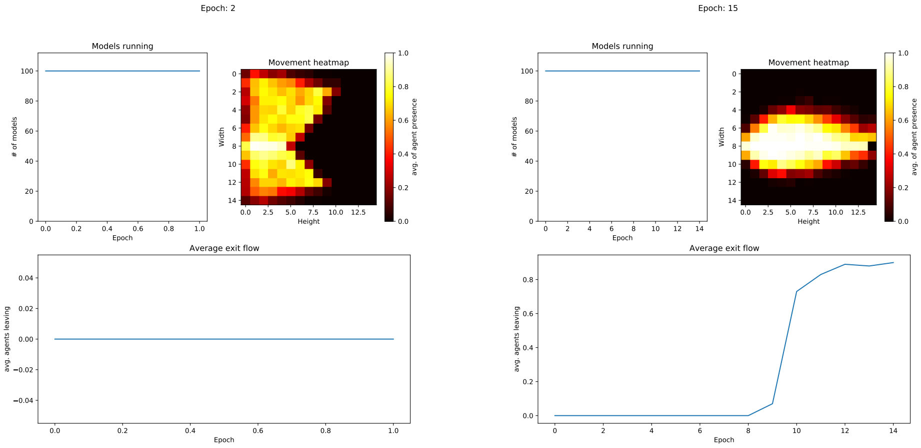


Figure B.5: Start of the simulation.

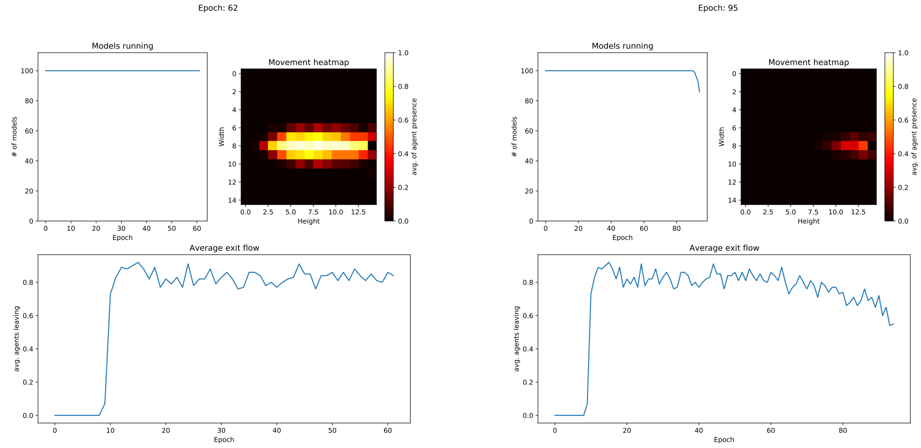


Figure B.6: Later stages of the simulation.

Appendix C

Contents of enclosed CD

```
│ readme.txt ...................... the file with CD contents description
├─ src..................the directory of evacuation simulator source codes
└─ text ........................................ the thesis text directory
   └─ thesis.pdf ........................the thesis text in PDF format
```