FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of master's thesis

Title:	Evacuation model with leading and following agents focused on evacuation of (pre)schools
Student:	Bc. 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 2023/2024

Instructions

The thesis should combine the idea of using a hierarchical system of multi-agent coordination for evacuation simulations (developed in Janovská, 2021) with expert knowledge on pre-school children's behaviour during evacuation (captured by Najmanová, 2020). The work should result in an evacuation model/software prototype, where the dynamics of the following agents (children) are consistent with the expert knowledge, and the planning actions of leading agents (teachers) are inspired by real-world instructions and limitations.

1. Survey evacuation models enabling assisted evacuation with a focus on evacuation of (pre)-school activities. Focus on cellular models.
2. Survey studies dealing with applications of multiagent planning algorithms in evacuation simulations.
3. In cooperation with the fire engineering expert (H. Najmanová) identify basic principles of children's behaviour and teachers' activities during evacuation. Suggest a transformation of those principles to rules of agent interaction in the cellular model.
4. Design and implement the cellular model of evacuation with the above-mentioned features enabling the application of planning algorithms for the actions of leading agents.
5. Perform several simulation experiments and compare various leading agent strategies with respect to total evacuation time.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 27 December 2022 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Evacuation model with leading and following agents focused on evacuation of (pre)schools

Bc. Matej Šutý

Department of Applied Mathematics
Supervisor: Ing. Pavel Hrabák, Ph.D.

April 25, 2023

Acknowledgements

THANKS (remove entirely in case you do not with to thank anyone)

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 April 25, 2023 . . . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Šutý, Matej). *Evacuation model with leading and following agents focused on evacuation of (pre)schools.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Abstrakt

V několika větách shrňte obsah a přínos této práce v českém jazyce.

Klíčová slova Replace with comma-separated list of keywords in Czech.

Abstract

Summarize the contents and contribution of your work in a few sentences in English language.

Keywords Replace with comma-separated list of keywords in English.

Contents

List of Figures

List of Tables

Introduction

Cooperation

Chapter 1

State-of-the-art

Chapter 2

Analysis and design

2.1 Model

This thesis proposes a novel hierarchical system for the coordination of multi-agent evacuation simulations in a preschool environment. The hierarchical structure is a crucial aspect of the model, as it facilitates the efficient management of the various agents within the simulation highlighted in the work of Janovská [2]. The proposed system consists of two distinct agent types: children and a leader. The children have the capability to form pairs and follow the orders of a leader, while the leader is responsible for navigating towards the goals and influencing children in its close proximity. Additionally, both agent types can have varying speeds of movement to reflect real-world behavior. A centralized planning approach is employed to control the formation of pairs, the location of the leader in the queue, and the assignment of goals among others. The model presents a unique and innovative approach to the simulation of multi-agent evacuations in a preschool setting, which has significant implications for the field of emergency management and preparedness.

Figure 2.1: Map `small.txt` initialized.

2.1.1 Floor field model

The baseline for the evacuation model with leading and following agents is a widely used extended cellular automata floor field model [3]. A cellular automaton is a mathematical or computational system composed of discrete elements that can be in a finite number of states - Moore neighbourhood for a single agent and extended Moore neighbourhood for a pair, further details in Section 2.2.9. These elements are updated according to a set of rules. While deterministic in behavior, the variance in results due to stochastic selection can mimic the randomness observed in human behavior, making it a desired feature of the model. The floor field CA model, which is the inspiration for the model used in this thesis, utilizes several floor fields, including static, dynamic, and leader proximity, to model pedestrian movement in a two-dimensional lattice. Agents, representing pedestrians with individual parameters, move on the lattice and interact with the floor fields. The model update rules include updating the dynamic field, calculating the transition probability for a move to a neighboring cell, choosing a random target cell based on the probability,

resolving conflicts when two or more agents attempt to enter the same cell. The evacuation model in this thesis represents people in evacuation as agents who move on a rectangular two-dimensional grid that acts as a room, corridor or other area.

2.1.2 Hierarchical structure

The hierarchical system follows the research of controlling swarm with leading and following agents [2]. Such a system suits the environment of pre-school, where children (following agents) are used to following a teacher, parent or other adult person with authority (leading agent). The leading agent follows the optimal path (set by static floor field) to the current goal and sets the static floor field for the goal of following agents. At the same time, the leading agent can change it's position in the crowd and influence the following agents by it's proximity. More details are in Section 2.3.3. Leading agent takes into consideration following agents that are left behind and adjusts its speed to allow them to catch up. Also, following agents are semi-autonomous which means that they can evacuate the room when given the opportunity of being close to the exit.

2.1.3 Planning

Through communication with fire engineering expert H. Najmanová, the fundamental principles of children's behavior and teachers' activities during evacuation have been translated into a set of simple rules and strategies for the leading agent. These rules, when implemented, have been shown to impact the course of evacuation in several ways. Firstly, the speed at which agents move is affected, which consequently results in different total evacuation time (TET). Secondly, the structure and coherence of the group of agents is influenced. Lastly, the microscopic behavior of agents, including the formation of pairs, is observed to change. This thesis provides further research for many ideas from the dissertation thesis of Najmanová [1]. The developed model provides a framework for assigning goals and different rules for leading and following agents. These rules are executed sequentially and are checked for whether they are achieved or not. Static assignment of goals and rules was sufficient for the needs of the research in this thesis. However, the framework can be easily adjusted to process goals and rules in real-time, for example using analyzer of the state of the simulation which generates goals. This enables the modeling of the complex interactions that occur during evacuation processes, and provides a basis for analyzing the effectiveness of different evacuation strategies.

2.2 Methods

2.2.1 Static floor field

Each cell in the grid in the static floor field *SFF* holds the value of shortest distance to the goal. The value is pre-computed using breadth-first search *BFS* algorithm which allows diagonal movement. Alternatively, the distance can be computed by other algorithms, e.g. novel approximate algorithm described in [4]. The (virtual) leading agent has full information about the topology of the map and he moves optimally towards the closest goal. The follower agents, solitary or in pairs, follow the leading agent and they do not attempt to find their own way to the exit [1]. The virtual leader agent updates *SFF* for the follower agents - either with the current goal location or the virtual leader agents position.

2.2.2 Occupancy floor field

Each cell in the grid can hold at most one physical agent (leading or following). The virtual leader agent, does not occupy a cell even though its position is the same as the cells position. The occupancy floor field *OFF* is a boolean matrix of the same dimensions as the room. The position of cells occupied by a physical agent hold *true* value, otherwise the value is *false*. The occupancy floor field updates in each step of the model.

2.2.3 Attraction and sensitivity parameters

The selection of the next cell by an agent is influenced by both the agent's own state - sensitivity parameters, timestep, and partner agent, as well as the state of the agent's surroundings, such as the SFF of the cell, distance to the leader, occupancy of the cell, and obstacles in the corner. The agent computes the attraction of each cell in its surroundings using a mixed strategy based on the method proposed by Šutý [5] shown in Equation 2.1. The attraction of each cell is then normalized across all cells to compute the probability of selecting a particular cell from the set. The probability of an agent moving from cell x to cell y, denoted by $P(y \leftarrow x \mid N)$, is calculated based on two members, P_O and P_S.

$$P(y \leftarrow x \mid N) = k_O P_O(y) + (1 - k_O) P_S(y) \tag{2.1}$$

Specifically, $P_O(y)$ takes into account the occupancy of cell y and returns a normalized value in the range $[0, 1]$. On the other hand, $P_S(y)$ considers the static potential of cell y and guides the agent towards the exit, along with the diagonal motion indicator $D(y)$. Both P_O and P_S are normalized across neighboring cells from N.

$$P_O(y) = \frac{\exp(-k_S S(y))(1 - O(y))(1 - k_D D(y))}{\sum_{z \in N} \exp(-k_S S(z))(1 - O(z)(1 - k_D D(z))} \tag{2.2}$$

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

The equation for computing cell attraction serves as the baseline for solitary agents moving to Moore neighborhood. When paired partner agents are involved, each agent computes the attraction of both cells in each maneuver. For instance, agent a_1 in cell c_1, paired with partner agent a_2 in cell c_2, computes the attraction of maneuver $m_1 \in M$ in the following manner: $A_1 = A(a_1 \rightarrow c_{1*})$ and $A_2 = A(a_2 \rightarrow c_{2*})$, where c_{1*} and c_{2*} represent the cells after the maneuver. The attraction of maneuver m_1 is then determined as $A(m_1) = \min(A_1, A_2)$, and the probability of selecting maneuver m_1 by agents a_1 and a_2 is given by $P(m_1|a_1, a_2) = \frac{A(m_1)}{\sum_{m_i \in M} A(m_i)}$.

2.2.4 Conflicts

The finite number of cells in discrete rectangular grid results in conflicts when two or more agents attempt to enter the same cell. The winner of this conflict, allowed to enter the cell, is picked randomly with uniform probability for each participant. This deviates from the conflict solution in research of aggresivity of Šutý[5], where winner of the conflict was the agent with highest aggresion or a conflict happened where no winner was selected. The described method was omitted in this research because it would bring more variability and less readability in the scope of this thesis' research. The framework is ready to include the method if needed. Intuitively the author would recommend to assign highest aggressivity value to the leading agent and lower aggressivity value or distribution of values to the following agents. The conflicts result in a cancellation of movement for agents, that will stay in their position in the next step and thus they prevent the movement of

bounded agents. The method, which cancels the movement and breaks cycles is described in Chapter 3.

2.2.5 Orientation

The grid in this model is a rectangular lattice of square cells where the agents can move from one cell to another in 8 directions of Moore neighbourhood and in special cases of a maneuver of a pair to the extended Moore neighbourhood as defined in Section 2.2.9. The agents have four orientations: North represents the top of the grid and South the bottom, East is the right-hand side of the grid and West is the left-hand side. The orientations are perpendicular one to each other. Even though the movement is possible in 8 directions the agent has only 4 orientations to facilitate the structure of paired agents. Two agents in a pair have the same orientation and are located in adjacent non-diagonal cells. Movement to a non-diagonal cell changes orientation while movement to diagonal cell preserves the agents' orientation. However, there are a few exceptions for paired agents as described in detail in Section 2.2.9.

2.2.6 Directed agents

Agents can move to cells in Moore neighbourhood in 8 directions or stay in the same cell. Directed agents take into consideration the direction of the movement. The agents move in a discretized rectangular grid. The agent has 4 possible orientations - North, East, South, West - which are global and not relative to the agent itself. This means that agent which has orientation East is directed to the right-hand side of the grid. Moore neighbourhood allows movement in 4 non-diagonal and 4 diagonal directions. In case of the non-diagonal directions, the agent orientation after the move is the same as the direction e.g. agent with orientation North moving to adjacent cell on the right will change orientation to East. In case of diagonal directions there are 2 types of movements: the first is movement to diagonal cells where the steering angle is +-45 degrees and the second is movement to diagonal cells where the steering angle is +-135 degrees. In the case of the first movement the agents keeps his orientation unchanged, e.g. agent facing West moves to diagonal upper left cell will have West orientation after the movement. In the case of the second movement the agent changes the orientation to the opposite orientation e.g. North to South and vice versa, East to West and vice versa.

2.2.7 Partner agents

Partner agents form a pair of two directed agents that are tightly bound so that they are in adjacent cells to each other. Two partner agents in the pair cannot be in cells diagonal one to each other. Their movement is synchronous, which means that if any of the agent is not able to move to the desired cell (conflict, bound agent did not move), the partner agent will abort its movement. The algorithm which checks whether both agents will successfully move is described below. In this model, the two agents in the pair are in a hierarchy of one agent being the leader while the second agent is not. The leader is responsible for calculating the probabilities of movement to cells for itself and it's partner according to maneuvers. The leader is the agent which has it's partner on the right. The partner agents are directed agents and both have the same orientation. Note that some maneuvers change the leadership in the pair.

2.2.8 Pair formation

The children in pre-school age are being taught to form a pair and hold hands when walking through corridor or crossing a road as these situations pose risks such as getting

lost or encountering traffic. The model in this thesis simulates the coordinated evacuation with supervisor where a risk element is present. The pupils are located in a classroom in a cluster where some pupils are close to each other and others are more isolated. The model does not consider any friendship preferences between children and assumes that pupils close to each other are more likely to form a pairs. It also assumes that pupils form pairs in group of even number of pupils so that there are no solitary children. If a pupil can't form pair immediately it will do so when other solitary pupil is nearby.

Algorithm 1 Finding pairs

1: $G = (V, E)$
2: **while** $\exists v \in V : d(v) > 1$ **do**
3: Let $v^* \in V : d(v^*) = \max_{u \in V} d(u)$.
4: Let $w \in V : d(w) = \max_{(v^*,w) \in E} d(w)$.
5: Remove (v^*, w).
6: **end while**
7: **for** $(v, w) \in E$ **do**
8: `formPair(v,w)`
9: **end for**

The Algorithm 1 finds a way to form pairs of pupils which are not yet in pair. Locations of pupils (directed agents) not in pair are transformed to a graph. The vertices of the graph correspond to the pupils' locations, and an edge is formed between two vertices in adjacent cells. The vertices in the graph may have different degrees, and cycles may be present. The algorithm iteratively selects the vertex with the highest degree and removes the edge connecting it to the vertex with the highest degree until all vertices have at most one edge. The vertices connected by an edge represent a pair.

2.2.9 Maneuvers

Each agent in a pair has the potential to move to 8 cells in the Moore neighbourhood, allowing for a vast number of possible maneuvers. However, most of these maneuvers do not maintain the structure of the pair and are therefore prohibited. Specifically, only 18 viable maneuvers are allowed for each orientation of the paired agents.

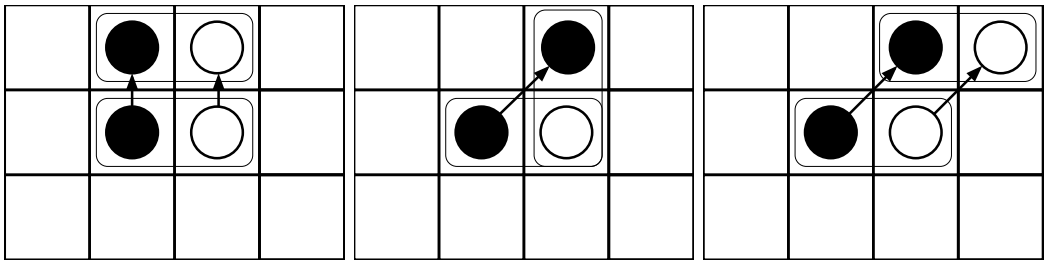

Figure 2.2: Simple maneuvers of paired agents.

Some maneuvers can alter the leadership of the pair or the orientation of the agents. Additionally, in some cases, the agents may have different movement speeds. During the synchronous atomic movement of a maneuver, both agents may move to diagonal cells, or one agent may move to a diagonal cell while the other remains in its current cell. Alternatively, one agent may even move outside its Moore neighbourhood in order to preserve the structure of the pair. In the Figure 2.3 can be seen the notable maneuvers.

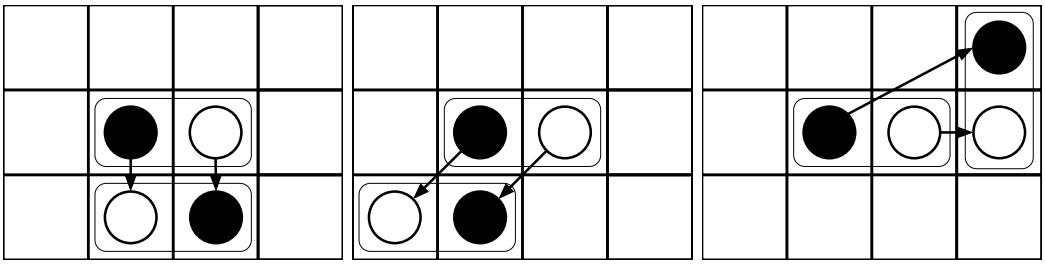

Figure 2.3: Complex maneuvers of paired agents.

2.2.10 Leader

The solitary agent responsible for navigation is the leader agent, which has complete information about the map topology and goals. The leader agent is not directed and can move to cells in the Moore neighborhood based on the SFF of the current goal. The virtual leader is a leader agent, which does not occupy a cell but his position is is used as a goal to set SFF for follower agents. The SFF for the leader agent differs from the SFF of follower agents. The SFF for follower agents is calculated in every step based on the virtual leader's position. The virtual agent navigates the followers when the leader moves to the end of the crowd. Agent's proximity to the leader (not virtual leader) proportionally increases the static potential value by increasing the sensitivity to static potential k_S:

$$d = \text{distance}(\text{leader}, \text{follower})$$
$$k_S = k_S * (1 + \frac{1}{d})$$

With a higher static potential value, the agent is less likely to deviate from the optimal trajectory set by the SFF of the virtual leader. This method is described in more detail in Section 2.3.3. The leader has simple rules for navigating towards the closest goal and checking if all follower agents reached the goal. One rule is the leader's ability to command the follower agents to continue to the goal while the leader moves to the most distant agent. Additionally, the (virtual) leader agent waits near the goal area and attracts the follower agents until they all reach it.

2.3 Strategies and rules

2.3.1 Leading agent strategy

In the hierarchical system a leading agents is responsible for navigating the following agents. The leading agent can achieve it in different ways as defined by the strategy. There are three leading agent strategies:

(A) Navigate towards the exit and evacuate.

Leading agent follows the optimal path to the exit. Leading agent positioned at the front of the crowd, the following agents are navigated by SFF calculated for them by virtual leader. Any agent, including the leading agent, that reaches the exit evacuates the room immediately.

(B) Navigate towards a location with specific position of the leading agent in the crowd.

Virtual agent follows the optimal path to the location. The leading agents position can be at the front or at the back of the crowd. The role of a virtual leader is essential in this strategy because the virtual agent is always at the front of the crowd and sets SFF for following agents. The virtual leader agent dynamically evaluates the

distance to the most distant following agent and adjusts its own speed (and the speed of the leading agent if positioned at the front) so that the distant agent is not left too far behind. In the case of a leading agent positioned at the back of the crowd, leading agent adjusts its speed dynamically so that it is always at the back of the crowd. For detailed explanation visit Section 2.3.4. The close proximity of the leading agent at the back of the crowd increases the speed of the following agent at the back of the crowd so that they could join the crowd and make it more compact. In the end, the virtual agents tries to make the crowd compact by adjusting the speed at the front and the leading agent at the back influences the following agents at the back to make the crowd more compact.

(C) Leading agent standing guard at a location and following agents navigate towards next goal location.

Leading agent is standing at a strategic location, for example next to the exit or at an apex of a corner and the following agents are navigated by SFF set by virtual leader to their goal location. Following agents in the crowd pass the leading agent at a close proximity and adjust their speed within the crowd due to its influence. The leader joins the crowd at the back.

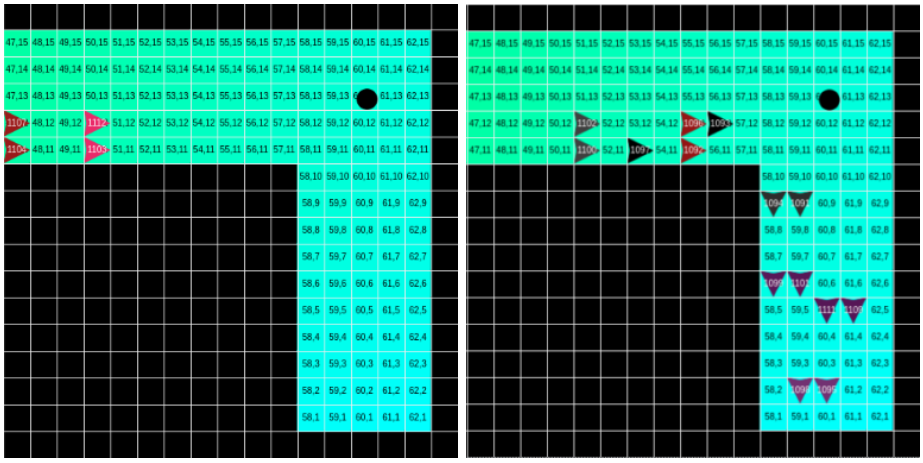

Figure 2.4: Leader strategy standing guard at a location and waiting (left), following agents passing the leader and navigating towards goal (right).

The three strategies can be used to organise children as described in [1]:

"Based on the leaving strategy employed, the following situation were observed during the experimental evacuation drills:

1. Leaving as a group at once: All children were gathered in front of a closed exit from a classroom forming a standing queue, first the whole group was completed the door was opened and children started to leave the classroom.

2. Leaving as a group gradually: All children were leaving together, however they were not gathered and checked in front of an exit from a classroom but when they arrived at the door as a moving queue, the door was already opened and supervised by a staff member and they could leave the room smoothly.

3. Individual leaving: Even though supervised children could leave a classroom individually instructed to wait at a specified place inside or outside the building." [1]

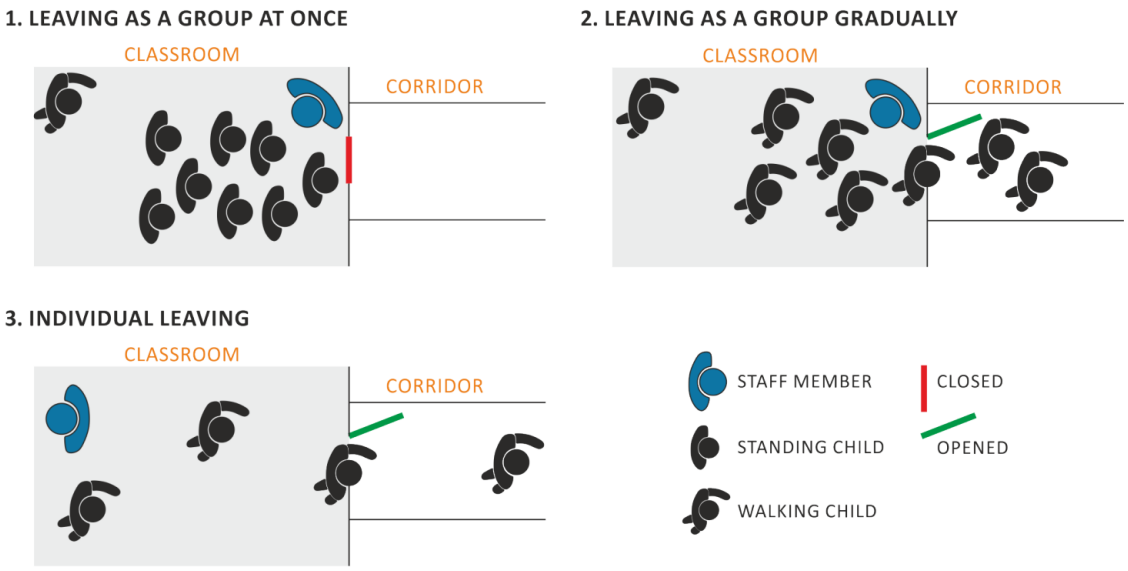

Figure 2.5: Image taken from [1].

2.3.2 Breaking pairs

The model implements the movement of paired agents through atomic and consistent maneuvers, preserving the structure of a pair. However, when paired agents approach a congested exit cell, their movement maneuvers become challenging due to high cell occupancy in the area of the exit. To overcome this difficulty, the pairs are split when they are in close proximity to the exit, but not when they are near their current goal. It is noteworthy that the topology of the map assumes a suitable path and adequately wide corridors to accommodate paired agents, thus eliminating the need for methods to handle dynamic breaking and creation of pairs based on the current evacuation status. Nonetheless, the framework can be customized to cater to specific requirements, as dynamic creation or breaking of pairs is already incorporated into the model and can be easily modified.

2.3.3 Penalization and discipline

The behavior of children is known to be strongly influenced by the presence of authority figures, such as parents, teachers, or other responsible adults. In the proposed model, the leader assumes the role of authority and guides the children as they move through the environment. Specifically, children who are in close proximity to the leader exhibit higher levels of *discipline* and move in a more orderly manner, preserving the structure of the queue of pairs and following the optimal path as determined by the static force field (SFF). To promote this behavior, the static force value assigned to each agent is adjusted according to their proximity to the leader. In particular, as the distance d between an agent and the leader decreases, the static force value S assigned to that agent is multiplied by a factor of $(1+\frac{1}{d})$. By moving back and forth within the queue, the leader can selectively increase the discipline of agents in close range and have greater control over the overall behavior of the group.

The paired partner agents form a queue and move one pair after other. Due to the nature of *maneuvers* described in Section 2.2.9 the paired agents sometimes rotate and change orientation because a non-optimal maneuver was selected stochastically. To limit this behavior a penalization for maneuvers with incorrect change of orientation is introduced. The penalization value can be set to fit needs of the simulation. In this model the value was set to 0.5 and every maneuver which results in unwanted change of direction is

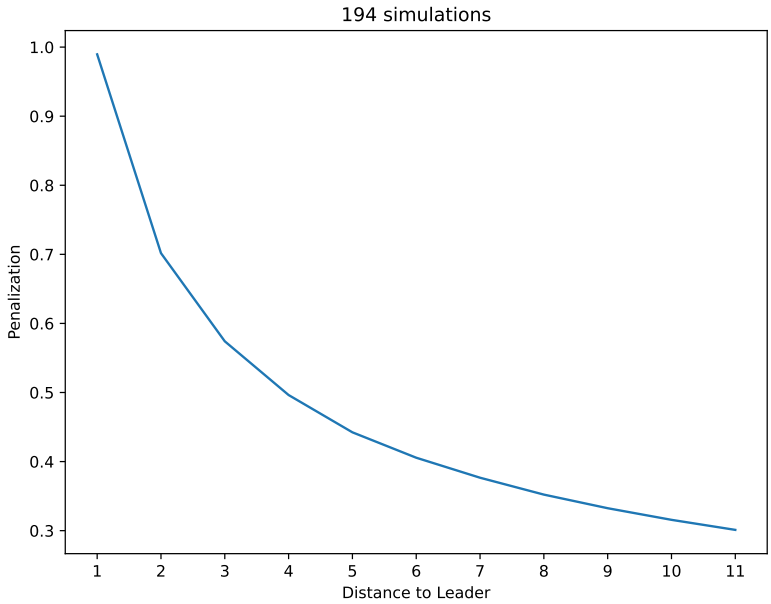

Figure 2.6: Penalization decreases with distance to the leader.

multiplied by the penalization to lower the chance of it being selected. Maneuver resulting in rotation is wanted in case of agents turning in corner due to the change of leaders position. The correctness of the orientation is thus determined by the orientation of the most attractive maneuver. In example, paired agents facing North follow the leader which leads the queue situated on the North. The attractivity of maneuver moving both agents to the North is the highest and thus the correct orientation is North and maneuvers resulting in different orientation are penalized. In other example, paired agents facing North approach a right-turn in corridor where leader passed the turn and is located on the East. The maneuver which results in rotation of the agents to the East is the most attractive and thus the orientation East is selected as correct and maneuvers resulting in different orientation are penalized.

The nature of discrete rectangular grid lowers the resolution of movement around obstacles which needs to be addressed. A obstacle crossing penalization is introduced, which aims to restrict diagonal movement of agents across obstacles. The degree of penalization may be customized based on the requirements of the simulation, with a value of 0.5 being adopted in this instance. Non-obstacle crossing maneuvers do not incur any penalty. The degree of attraction is scaled by the penalization factor, with higher penalization values resulting in less obstacle-crossing maneuvers. Lower penalization of crossing penalization maneuvers in environments with narrow corridors or numerous obstacles can result in smoother movement of paired agent queues.

2.3.4 Adaptive time span and speed

This model allows agents to adjust their movement speed during an evacuation, with the leader represented as an adult with an average speed of $1.2m/s$, as per the findings in [6]. Follower agents, who are modeled as children, are assigned a speed of $0.9m/s$, based on experimental measurements of adult pairs traveling at a speed of $0.97m/s$. Each model step consists of two timesteps, representing the unit duration of movement. For instance, paired agents have a movement duration of three timestep units with a speed of $0.9m/s$, resulting in a speed of $0.3m/timestep$. The leader agent has a speed of $0.4m/timestep$.

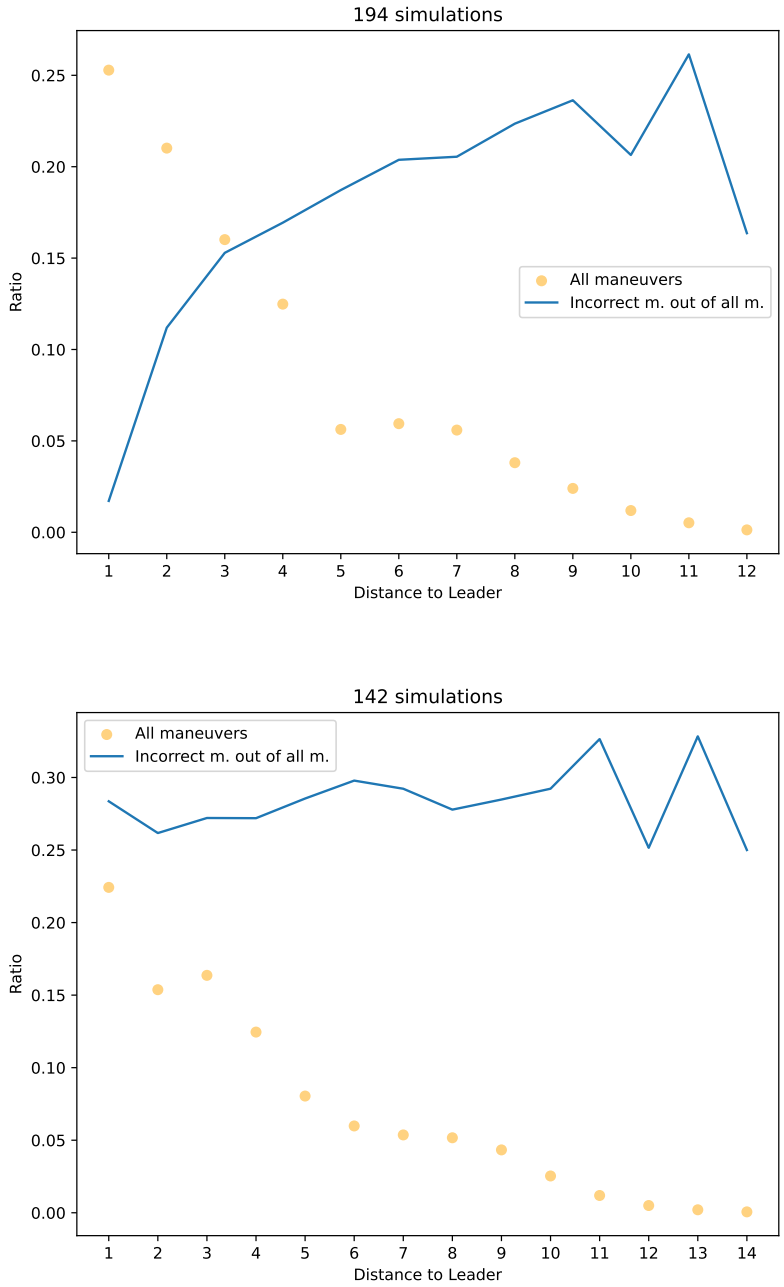

Figure 2.7: Simulations with penalization value 1 in the top figure show few incorrect maneuvers compared to bottom figure where penalization is 0.

This update frequency of two timesteps per model step allows for synchronization of movement every three steps between the leader agent and paired agents. Specifically, each agent is equipped with an internal counter, denoted by τ, which is evaluated against the model's timestep clock, denoted by T. The timestep clock advances by two timesteps for each model step. Whenever τ is less than or equal to T, the agent initiates cell selection and increments τ by the duration of the maneuver. Paired agents can dynamically adapt their speed in specific conditions. Close proximity to the leader and empty cell in front of them increases their speed. The leader agent can move back and forth in the queue and locally increase the speed of paired agents to repair the queue structure or close the gaps between agent pairs. The change in speed reverts when previous conditions are not

met. The leading agent, modeled as an adult person, has higher speed than following agents. For n following agents in the simulation and leading agent at the front the model calculates distance from the virtual leader to the most distant agent d and lowers his speed when $\frac{n}{2} \leq d$. In case of leading agent at the back, the distance d is calculated from the leading agent to the agent at the back. When $d < 5$, the leading agent keeps its current speed, otherwise its speed is increased to catch up with the crowd. The leading agent being very close to the most distant agent from the virtual agent at the front of the crowd increases the speed of the slow agent to catch up with the rest of the crowd.

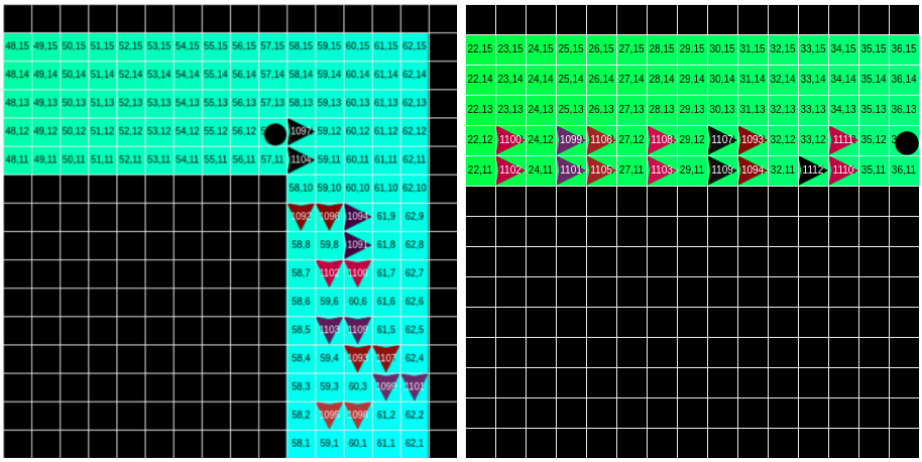

Figure 2.8: Leader position at the back (left) and at the front of the crowd (right).

Chapter 3

Implementation

The implementation of the model framework was carried out in Python, version 3.10, using several accompanying libraries. The complete source code is accessible at `https://gitlab.fit.cvut.cz/sutymate/mt-master-thesis`, along with comprehensive documentation and installation guidelines. The maps, an integral component of the model, are also included in the repository.

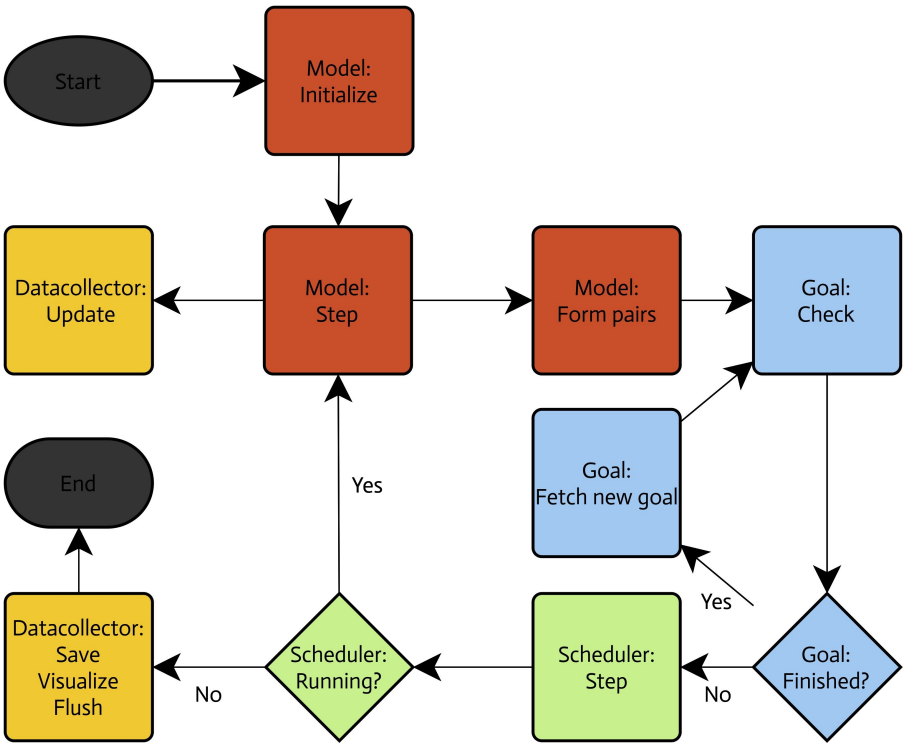

Figure 3.1: Schema of one simulation run.

In short, the architecture of the model is made out of several objects resting in the *RoomModel* class. *RoomModel* orchestrates initialization of the model, running it, collecting and saving data, creating graphs. *SequentialActivation* is a scheduler of agent movement in the cells. As can be seen in the Figure 3.2, the sequence starts by selecting agents that have $\tau < T$ as described in Section 2.3.4. These agents calculate *attractivity* of the cells in their neighbourhood and stochastically select their desired cell. Next, the cells which are selected by agents for entrance execute winner selection, which means that they select the agent from all candidates that will enter the cell. Next, all participanting agents verify whether they will enter their desired cell. If so, they will update their τ with the duration of the movement. If they can't enter their desired cell, they will *bubbledown*

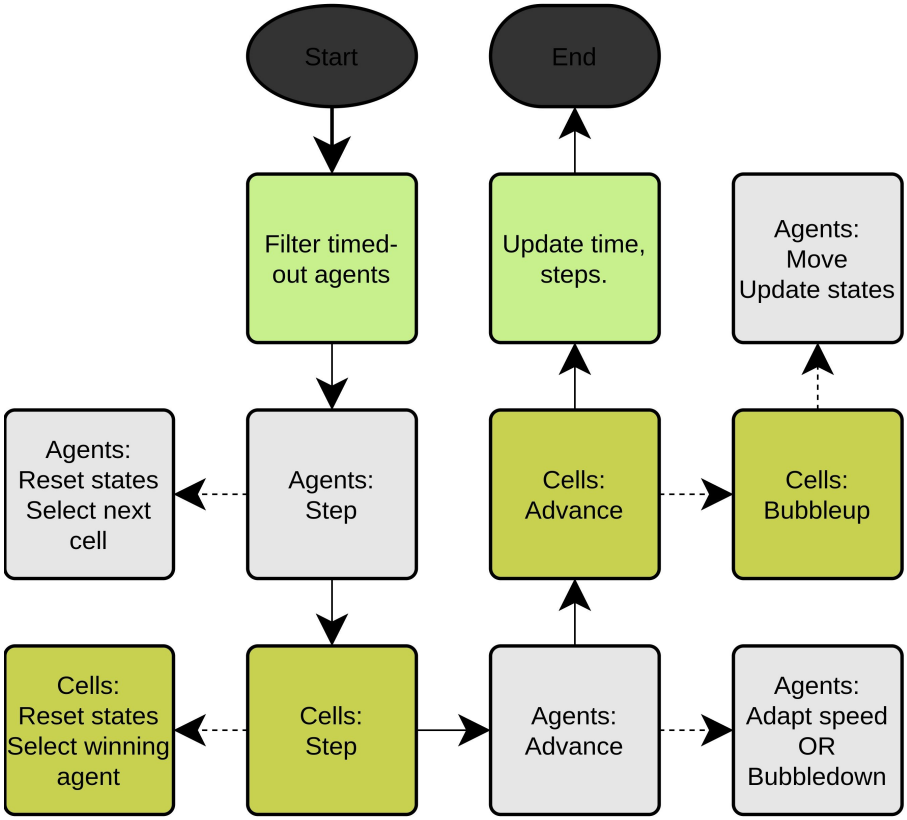

Figure 3.2: Task flow in one step of *SequentialActivation* object.

the information to bound agents behind them and cancel their movement as they won't move from their position, see Algorithm 2.

Algorithm 2 Agent bubbledown method

```
1:  function BUBBLEDOWN(AGENT)
2:      if agent.next_cell.winner == agent then
3:          agent.next_cell.winner ← null
4:      end if
5:      agent.next_cell ← null
6:      agent.confirm_move ← false
7:      if agent.partner.next_cell ≠ null then
8:          bubbledown(agent.partner)
9:      end if
10:     if agent.head ≠ null then
11:         agent.head.tail ← null
12:         agent.head ← null
13:     end if
14:     if agent.tail ≠ null then
15:         agent.tail.bubbledown()
16:     end if
17: end function
```

Next, the cells with winning agents *bubbleup* the bounded agents to the front of the bounded chain up to the agent which: a) has empty cell to enter, b) the bounded chain forms a cycle, which is detected and broken, see Algorithm 3. In either case, the agents participating in *bubbleup* will successfully move to their desired step. Lastly, the scheduler

updates internal states such as number of steps and timestep value.

Algorithm 3 Cell bubbleup method

```
1: function BUBBLEUP(CELL)
2:     head ← cell.winner
3:     origin ← cell.winner
4:     while head.head ≠ null do
5:         head ← head.head
6:         if head = origin then
7:             origin.head.tail ← null
8:             origin.head ← null
9:         end if
10:    end while
11:    return head
12: end function
```

3.1 Mesa

Mesa is a popular framework for ABM research [7]. The API offers many options for visualization, setting of parameters, data collection, etc. This framework has been successfully used in the previous research of SFF CA model of pedestrian evacuation in 2020 [5]. Moreover, Mesa was used in research of software for urban planning practice [8], multi-drone truck routing problem [9] and in the simulation of balancing consumer and business value of recommender systems [10].

3.1.1 Visualization

Mesa framework has built-in visualization module for web browser, which is suitable for fast development and visual analysis. The user can adjust the speed of simulation or continue step by step to watch in detail. Simulations can be rerun. Figure 3.3 shows the Mesa visualization in browser for a map *small.txt*. The black circle represents a physical leading agent, the two arrow agents of same color are paired agents with orientation to the west. The color gradient shows *SFF* with high values in green and low values in blue. Black grid cell represent obstacles. The parameter interface is described in Section 3.1.2.

3.1.2 Run-time parameters

The user can better understand the simulation when it runs under various settings. Mesa offers a user-friendly interface for setting global parameters of the simulation. As can be seen in the Figure 3.3, user can set various parameters:

- Sensitivity to static field k_S, where higher value increases the attractivity of the optimal cells as can be deducted from the Equation 2.1.
- Sensitivity to occupied cell k_O, where higher value prevents the agent from selecting occupied cell for next move.
- Sensitivity to diagonal movement k_D, where higher value prevents the agent from selecting diagonal cell for next move.
- Movement duration for leader in timesteps, where higher value means a longer duration of leading agent's movement results in lower speed.

- Movement duration for followers in timesteps, where higher value means a longer duration of following agents' movement results in lower speed.
- Penalization for incorrect orientation, where higher value penalizes maneuvers resulting in incorrect orientation and lowers their attractivity.
- Switch for leader location at the front forces the agent to keep position at the front of the crowd. When the simulation follows goals defined in the map file, this setting will be overwritten by the goal rules.

After setting new parameters, the simulation needs to be restarted.

3.1.3 Definition of map and rules

The maps used by the model are defined in an ASCII text file and can be created or edited for user's needs. The model requires the following structure:

```
maps/
 - data/<map_name>.data
 - topology/<map_name>.txt
```

An example of the input file format used in the simulation is shown in Listing 3.1.3. The map itself is defined at the beginning of the file using # symbol to represent obstacle, empty space for empty space and various types of agents as follows: L for leading agent's positions, D for directed agents' positions, other symbols are deprecated. After the map topology definition follows a custom hash value (in example 42), which validates the map topology to the precomputed SFF values in `/maps/data/<map_name>.data`. After that follow sequentially the goals together with rules. As described in Section 2.3, there are three goal types:

- Navigate towards the exit and evacuate is indicated by E followed by x, y coordinates and keyword *All*.
- Navigate towards a location with specific position of the leading agent in the crowd is indicated by L followed by integer x, y coordinates. Then an integer t tells the leader how long to stay in the position after the goal was reached. Position of the leader is marked by keyword *Front* and *Back*. Keyword *All* ends the sequence.
- Leading agent standing guard at a location and following agents navigate towards next goal location is indicated by L followed by integer x, y coordinates of leading agent's location. Then an integer t tells the leader how long to stay in the position after the goal was reached. Position of the leader is marked by keyword *Front* and *Back*. Keyword *All* ends the sequence.

```
################
#      #     LD #
#              D #
#      #       #
################
 42
G 7 3 0 Back All
L 2 2 0 Front All
E 1 2 All
```

3.1.4 Simulation experiments module

Simulation experiments are defined in the abstract class *Experiment*. *Experiment* provides interface to simply initialize simulation data in required format, load data from previous runs, update them with current run and save for later use. Also, each experiment defines how to visualize data.

3.1.5 Datacollector

RoomDatacollector class provides a simple interface for running multiple *Experiment* instances and saving, showing or just aggregating the data. At the same time, *RoomDataCollector* is a derived class from the Mesa *DataCollector* class, which can be used to visualize data in real time in the browser visualization.

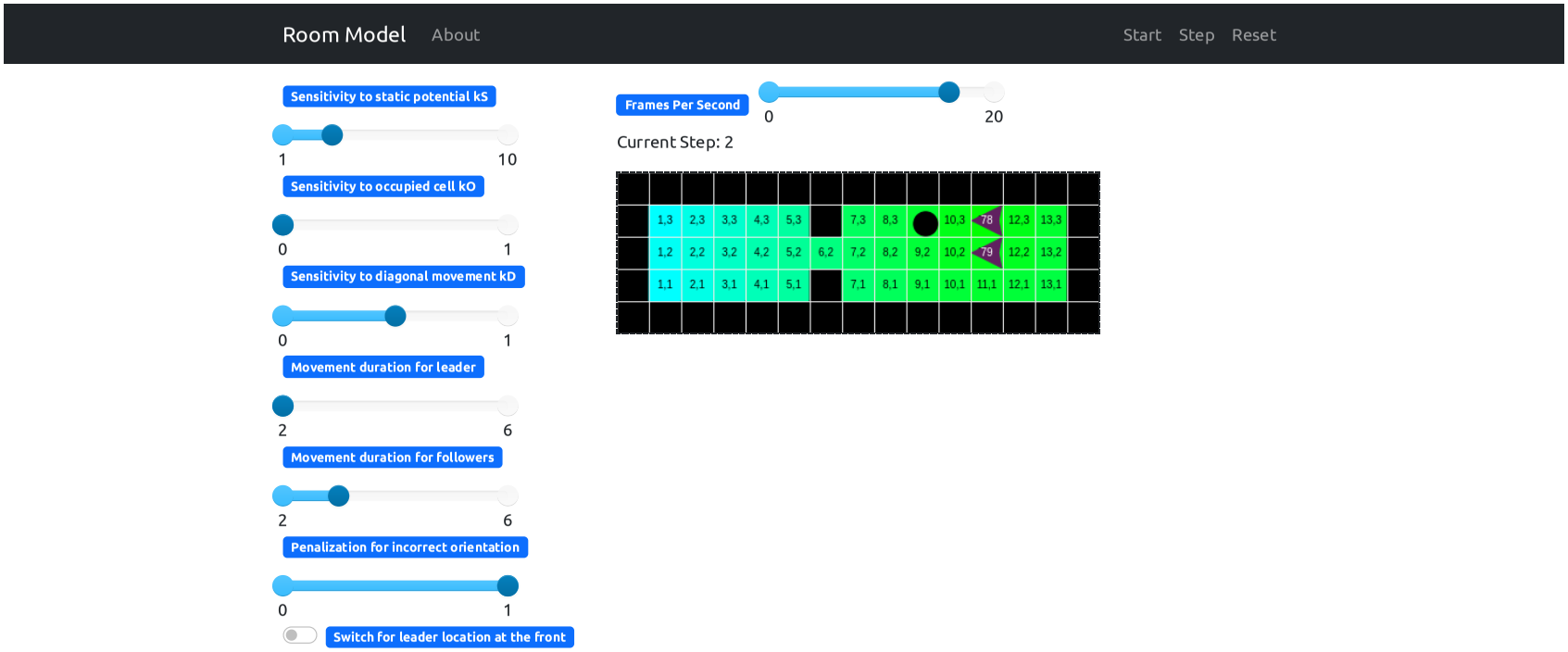

Figure 3.3: Visualization of evacuation simulation in browser.

Chapter 4

Simulation experiments

4.1 Maps

In the simulation experiments, four distinct map files were utilized. The first three maps, `mapX1.txt`, `mapX2.txt`, and `mapX3.txt`, each featured a classroom setting with varying corridor lengths and shapes. `mapX1.txt` had a long corridor, while `mapX2.txt` had a shorter corridor. `mapX3.txt` included a turn in addition to the short corridor, making it slightly more complex than the other two maps. The first two maps have same exit location at the end of corridor and *A, B* locations are shared with `mapX3.txt`. The fourth map file, `gaps.txt`, depicted a narrow corridor with trigger gates. For clarity, the maps are depicted below in Figures 4.1, 4.2,and 4.3, while `gaps.txt` is shown in Figure 4.4. Overall, the selection of these four maps allowed for a diverse range of scenarios to be simulated and tested.

Simulation experiment runs with different goals and strategies are described in the Table 4.1.

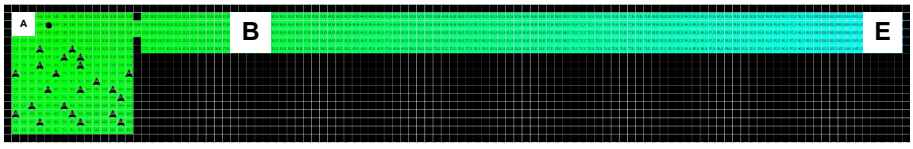

Figure 4.1: Initial positions and goals of leading and following agents in the classroom `mapX1.txt`.

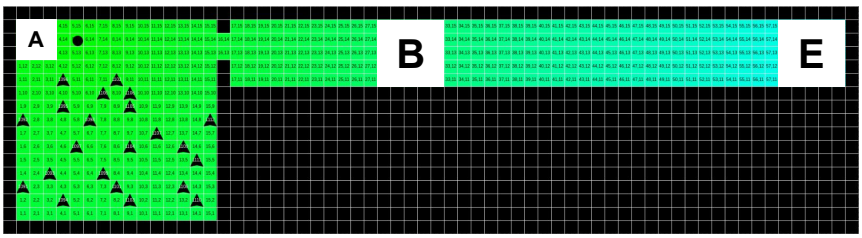

Figure 4.2: Initial positions and goals of leading and following agents in the classroom `mapX2.txt`.

Table 4.1: Goals of each map

Map	# of Agents	Goal 1	Goal 2	Goal 3	Goal 4
map01.txt	22 + 1	Exit (Front)	Location B (Front)	Exit (Front)	
map02.txt	22 + 1	Location A (Front)	Location B (Back)	Exit (Back)	
map03.txt	22 + 1	Location A (Front)			
map11.txt	22 + 1	Exit (Front)	Location B (Front)	Exit (Front)	
map12.txt	22 + 1	Location A (Front)	Location B (Back)	Exit (Back)	
map13.txt	22 + 1	Location A (Front)			
map21.txt	22 + 1	Exit (Front)	Location B (Front)	Guard C (Front)	Exit (Back)
map22.txt	22 + 1	Location A (Front)	Location B (Front)		
map23.txt	22 + 1	Location A (Front)			
gaps.txt	8 + 1	Exit (Front)			
gaps_back.txt	8 + 1	Exit (Back)			

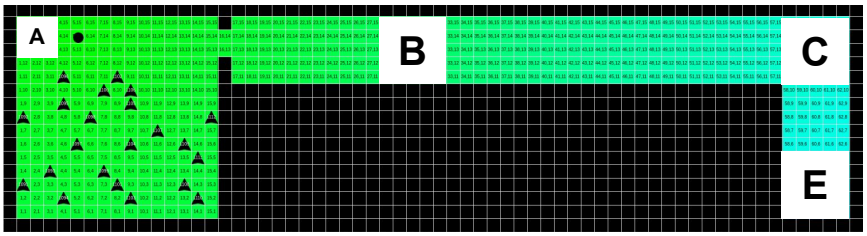

Figure 4.3: Initial positions and goals of leading and following agents in the classroom `mapX3.txt`.

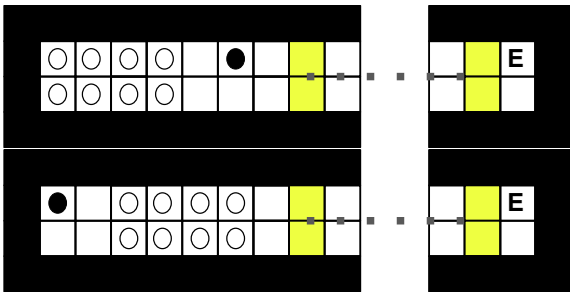

Figure 4.4: Illustration of `gaps.txt` with trigger points in yellow. Dots show the cells that were hidden due to the length of the map.

4.1.1 Room size for simulation experiments

According to legislation outlined in the civil engineering law of the Czech Republic, a minimum of $1.65m^2$ of space per person is required during the educational process in pre-school classrooms, with a recommended corridor width of 2 meters and a minimum width of 1.2 meters. Doors must have a required width of 90 centimeters [11]. In 2019, the Czech Statistical Office released a report on pre-schools and other educational institutions, indicating that the average number of children in a pre-school classroom is 22.6 [12]. Subsequent research conducted by Najmanová in 2020 further supports this finding, revealing that the most frequent number of children in a class ranged between 20 and 28 children (80.1%) with a median value of 24 children [1]. For the purposes of simulation experiments, classroom dimensions are set to 15×15 cells with a door width of 2 cells and a corridor width of 5 cells. Assuming a cell size of 40 centimeters, commonly used in evacuation simulation CA models, the resulting classroom dimensions are 6×6 meters with a door width of 80 centimeters and a corridor width of 2 meters. With each child having approximately $1.63m^2$ of space in the classroom, the minor discrepancy between the required space per child and the space utilized in the simulation, resulting from differences in door width and resolution of the SFF CA model, is negligible [11].

There is also other map type, with names of `gaps`, which puts focus on the structure of a queue of pairs following leader and the selection of incorrect orientatin maneuvers. These maps are narrow and long, with 2 cells in width and 60 in length. There are 8 directed agents and one leader, so that these agents could form pairs and follow the leader. The maps are have trigger gates which initiate experiment of measuring distance between pairs at cell 11 of width and cell 40 of width, which ends the experiment.

Simulation experiment	k_S	k_O	k_D	Penalization	Leader	Pairs
Penalization low	3	0	0	0	Yes	Yes
Penalization middle	3	0	0	0.5	Yes	Yes
Baseline	3	0	0	1	Yes	Yes
k_O medium	3	0.5	0	1	Yes	Yes
k_O high	3	1	0	1	Yes	Yes
k_S low	1	0	0	1	Yes	Yes
k_S high	5	0	0	1	Yes	Yes
Leader, no pairs	3	0	0	1	Yes	No
No leader, no pairs	3	0	0	1	No	No

Table 4.2: Parameter values for simulation experiments

Chapter 5

Results

The simulation experiments conducted using the proposed hierarchical model investigated various aspects of the simulated evacuation. One part of the simulation experiments aim to investigate the structure and behavior of paired agents following a leading agents. The results describe the gaps between pairs during the course of the evacuation, the consistency of their distance to the leader, the symmetry of the results based on the leader position in the pair and lastly the effect of the penalization parameter on the ratio of maneuvers resulting in incorrect orientation.

The model needs to be versatile to simulate evacuations in various environments and situations, which is investigated in the simulation experiments of the total evacuation time in different maps, scenarios and parameters. Additional analysis of specific flow of agents through key areas in the environment explain the dynamics of the simulation and the way scenarios effect it. The hierarchical model is based on the floor field CA model, which operates with several model parameters. The influence of the major parameters k_S and k_O, as noted in the sensitivity analysis [5], is analysed.

The analysis of the simulation experiments provided valuable insights into the pedestrian movement in the hierarchical model. The results shed light on how individual parameters impact the course of the evacuation and the structure of agent pairs. Additionally, the use of strategies and rules was closely examined to correspond to the behavior of child agents on qualitative level.

5.1 Gaps between pair

The present study involves a simulation experiment conducted on the map `gaps.txt`, which features a narrow long corridor with a leading agent (positioned at the front and at the back of the crowd) and eight following agents, as depicted in Figure 4.4. The Figure depicts two trigger zones in yellow that border the investigated area. The following agents in the area are already in pairs and the initial noise in pair formation is reduced as well as the noise at the exit. In each step of the model, the pair position is captured and later processed as a distance between pairs. It is important to note that the trigger zones shown in Figure 4.4 are only for illustration purposes and do not affect the simulation experiment.

The primary objective of the experiment is to investigate the structure of the queue formed by the pairs of following agents. A key aspect of this structure is the inter-pair distance, which is illustrated in Figure 5.1 through a series of histograms. Specifically, each histogram captures the distribution of distances between a given pair and the pair immediately in front of it. The analysis of these histograms reveals that the distance between the first and second pair is comparable to the distances between the second and third pair and the third and fourth pair.

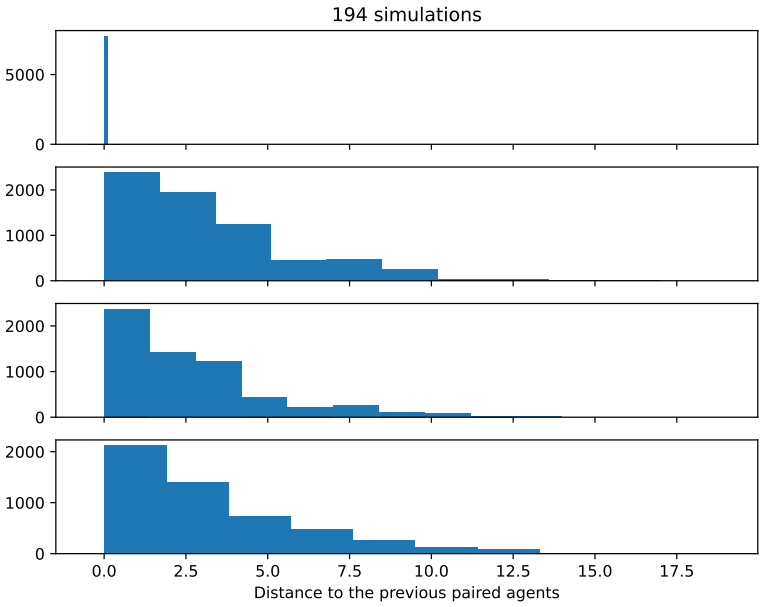

Figure 5.1: Distance between pairs in map `gaps.txt` is consistent.

The simulation parameters *Baseline* used in almost 200 runs can be seen in the Table 4.2.

5.2 Pair formation

One goal of the model was to capture the movement of pairs of children. Once two following agents form a pair, it is assumed they will stay together until they are close to the evacuation cell. Figure 5.2 depicts the distance of individual agents to the leading agent throughout the whole simulation. The distance of individual agent was calculated by collecting distance at every step in the simulation, averaged across all runs and then smoothed using rolling average with windows size 2.

Three scenarios on the same map with long corridor `map01.txt, map02.txt, map03.txt` were analyzed. As can be seen in the Figure 5.2, once the following agents are paired and pass the bottleneck (noisy part of the graphs at early stage of evacuation), the agents move together, keeping same distance to the leader as their partner agent - the paired agents plot similar line.

The first scenario features the goal of navigating to the exit. The second scenario's goals are to group in the classroom, move to the location B and only then navigate to the exit. In both scenarios, the leading agent position was at the front of the queue. These two scenarios show small difference in the dynamics as can be also seen in the specific flow graphs in Figure 5.8.

Also a group split is worth noting, as can be seen in the graph as few agents had low distance to the leading agent, while other agents were left behind and their distance increased until the congestion situation happened sometime around 200 model steps.

Opposite to this is the third scenario, where the leading agent was positioned at the back of the group and managed to keep the more compact as can be seen on the y axis, where highest distance is 25 compared to 35 in previous two scenarios. As expected, more distant agents evacuated the room earlier as the leader was positioned at the back, which is the opposite of the previous two scenarios.

The figures of three scenarios at map `map1X.txt` with shorter corridor are shown in Figure B.1 in the Appendix as they are comparable to the previous findings.

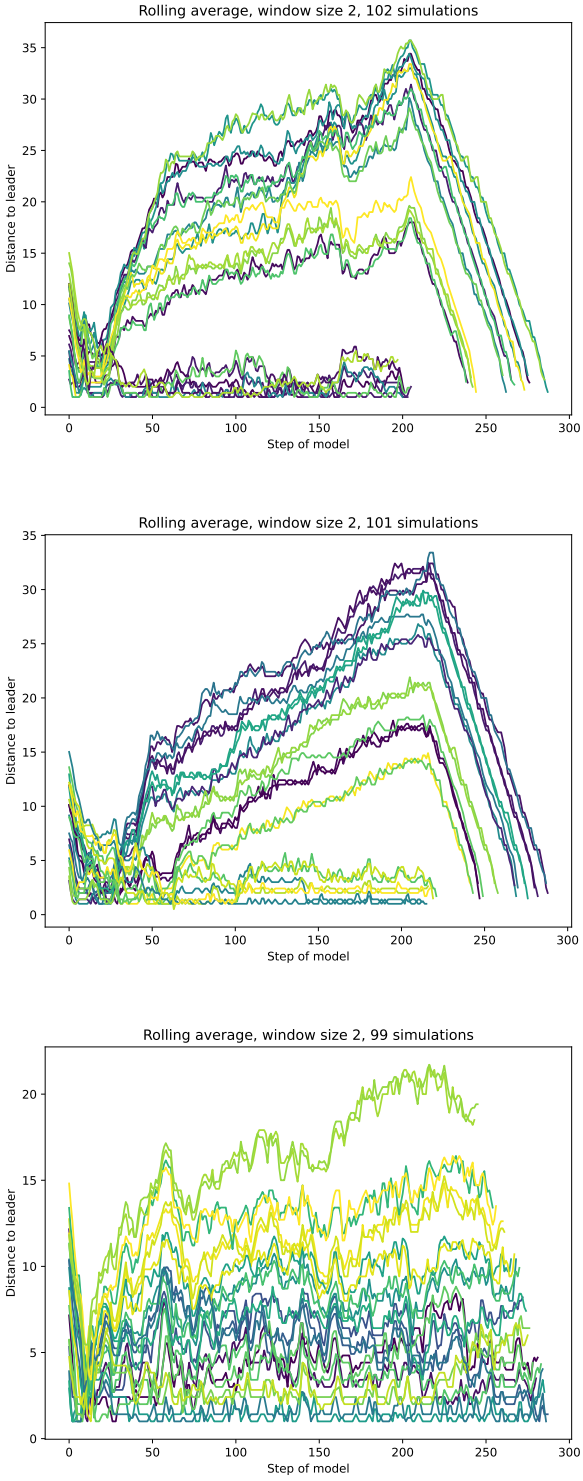

Figure 5.2: Distance of agents to the leader of `map01.txt` in the top figure, `map02.txt` in the middle figure, `map03.txt` in the bottom figure.

For completeness, three scenarios on the map with short corridor and right-turn `map21.txt, map22.txt, map23.txt` are depicted in Figure 5.3. The first scenario is comparable to the situation in `map01.txt` as the leading agent is navigating to the exit.

The distance of paired agents increases in time up to the congestion situation. The second scenario follows the goals and rules described in the Table 5.1. The last rule makes the agent positioned at the back of the group and he navigates the group towards the exit. There is less difference in the distances of agents and the value does not evolve in time, which is attributed to the agent position at the back. The last scenario included a rule of leading agent waiting at a location C. After evacuating the classroom and reaching location B, the agent departs from the group and waits for them at the corner apex. This is shown in the bottom graph as the sharp spike in distances of agents.

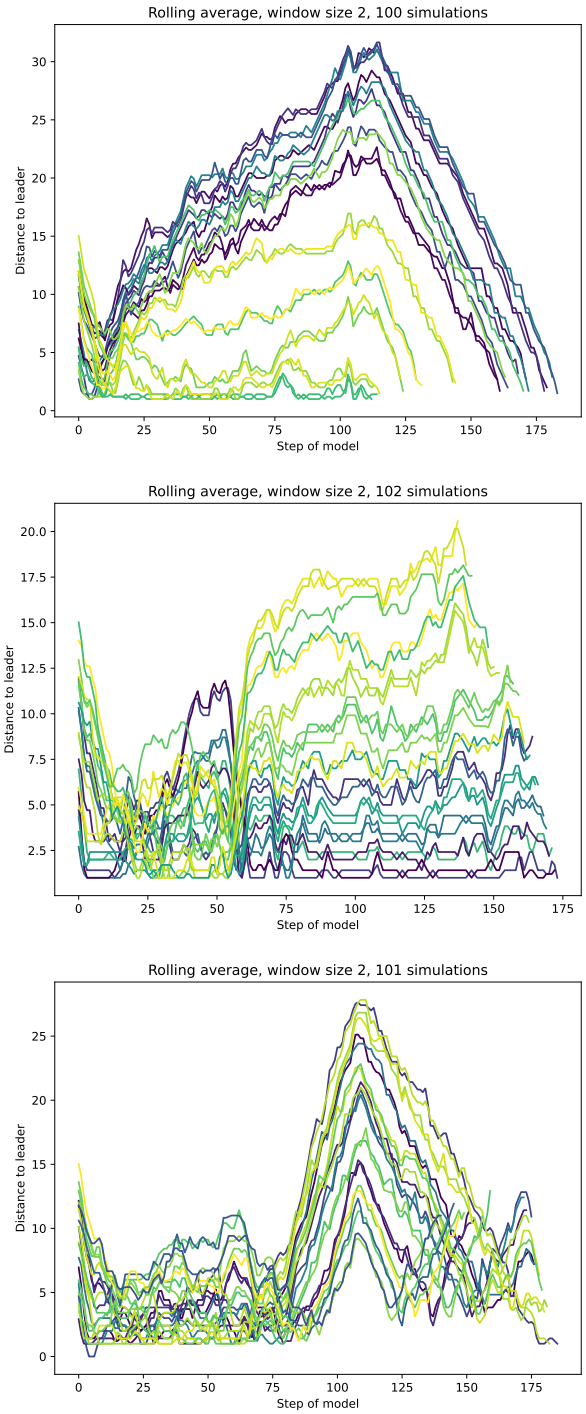

Figure 5.3: Distance of agents to the leader of `map21.txt` in the top figure, `map22.txt` in the middle figure, `map23.txt` in the bottom figure.

Another finding can be spotted in the Figure 5.4, where paired agents have same similar distances to the leader and its variance. The close the pair is to the agent, the less variance is in the distance, which supports influence of *discipline* on following agents, described in Section 2.3.3.

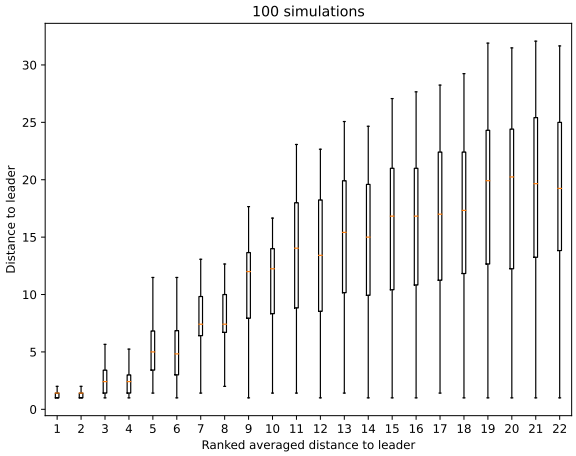

Figure 5.4: Distribution of distances of agents to the leading agent in `map21.txt`.

5.3 Symmetry of the experiments

The structure of a pair of following agents consists of a leader, positioned on the right in the direction of the orientation. This can raise question whether such requirement might result in *asymetric* movement or other form of *biased* movement. The following simulation experiments were run on mirrored versions of the maps described in Table 4.2. The map topology was mirrored on the x axis, as well as initial positions of the agents and the goals.

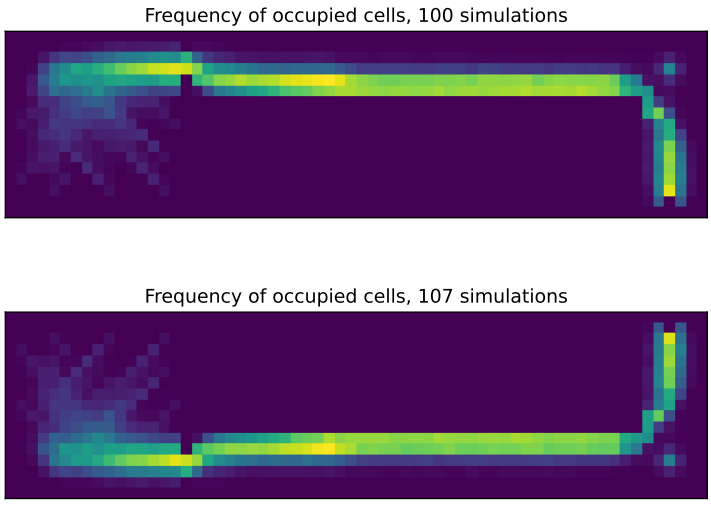

Figure 5.5: Heatmap of cell visits shows symmetry in `map23.txt` on top and the mirrored version below it.

The two Figures 5.5 depict two heatmaps of visits of the cells, which are almost identical apart from the mirrored topology. The most complex map and scenario `map23.txt`

was selected to allow all kinds of movements to be present.

Also, the boxplot graphs of the agents' distance to the leader in Figure 5.6 shows comparable variance and average values of distance of agents to the leading agent in the original and mirrored simulation experiments. This provides further evidence to support the claim, that the structure of agents' pair does not result in asymmetric movement or other form of biased movement.

In more than 100 simulations, further quantitative analysis revealed that the average TET was 178.29 ± 5.63 steps in map `map23.txt`, and 178.94 ± 4.84 in map `map23_mirror.txt`. Moreover, the TET in other maps was found to be comparable in both the original and mirrored versions.

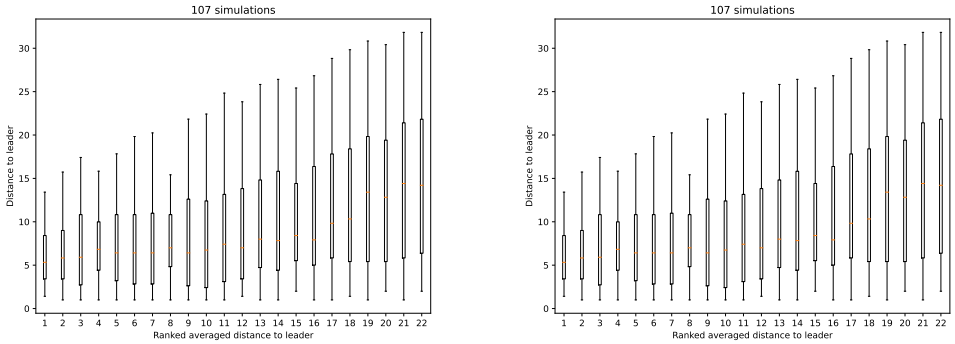

Figure 5.6: Boxplot of distances to the leading agent in original map `map23.txt` and mirrored map `map23.txt`.

5.4 Total evacuation time

TET is averaged across more than 100 simulations in three different maps `map0X.txt`, `map1X.txt`, `map2X.txt`. The first digit in map name indicates the topology and second digit in map name indicates scenario type as was explained in Table 4.2. Each map contains three scenarios with different goals such as navigating to a location, waiting for following agents to form a queue and different rules as position of the leading agent at the front or at the back of the queue and standing guard at a location. The TET values can be seen in Table 5.1.

The TET from the *Baseline* simulation experiments with a leading agents and following agents forming pairs is compared to simulation experiments without leading agent and with leading agent but without pair formation. There is another comparison with optimal path TET set by a single leading agent.

The highest TET is on map type `map0X.txt` which features long corridor. Simulation experiments with leading agent and following agents without forming pairs have lowest TET across all maps, which can be attributed to the following agents better ability to pass through the bottleneck of class door (width of 2 cells).

Table 5.1: Comparison of path lengths for different scenarios

	Baseline	No Leader, No Pairs	Leader, No Pairs	Optimal path
`map01.txt`	277	279	257	146
`map02.txt`	281	278	269	148
`map03.txt`	287	279	277	148
`map11.txt`	160	170	152	76
`map12.txt`	166	170	164	84
`map13.txt`	173	170	171	84
`map21.txt`	175	183	166	83
`map22.txt`	186	183	183	92
`map23.txt`	178	183	176	98

Simulation experiments without leading agent have comparable TET to the *Baseline* even though this method is very unlikely to happen in real world. As Najmanová noted in [1], children that are not assisted during evacuation do not necessarily select the most optimal exit, quite the opposite, and choose the familiar paths. Evacuation simulation experiments without leading agent are thus somewhat imaginary, because of the assumption that following agents know optimal evacuation path.

The *Baseline* simulation experiments show that adding goals to the evacuation leads to small increase in TET as can be seen in all maps across all scenarios. One exception is the scenario in map `map23.txt`, where leading agent stands guard at a location at the apex of the corner and evacuates as a last person. This scenario resulted in lower TET.

Across all maps and scenarios, TET in *Optimal path* set by single agent is close to half of TET in other simulations. This can be explained by the bottleneck situation at the single exit cell when $22 + 1$ agents reach the exit and evacuate one by one.

5.5 Penalization parameter

The penalization parameter is used to discourage the selection of maneuvers that lead to incorrect orientation. As the value of this parameter increases, the frequency of such maneuvers decreases, as evidenced in Figures 5.7. The yellow dots represent the proportion

of all maneuvers based on their distance to the leading agent, where a distance of 0.15 at distance 1 indicates that 15% of all maneuvers occur at distance 1. The blue line, on the other hand, indicates the proportion of maneuvers resulting in incorrect orientation, specifically at the given distance. For instance, 0.3 at distance 1 implies that 30% of maneuvers at distance 1 resulted in incorrect orientation, which corresponds to ratio $0.15 \times 0.3 = 0.045$ of all maneuvers.

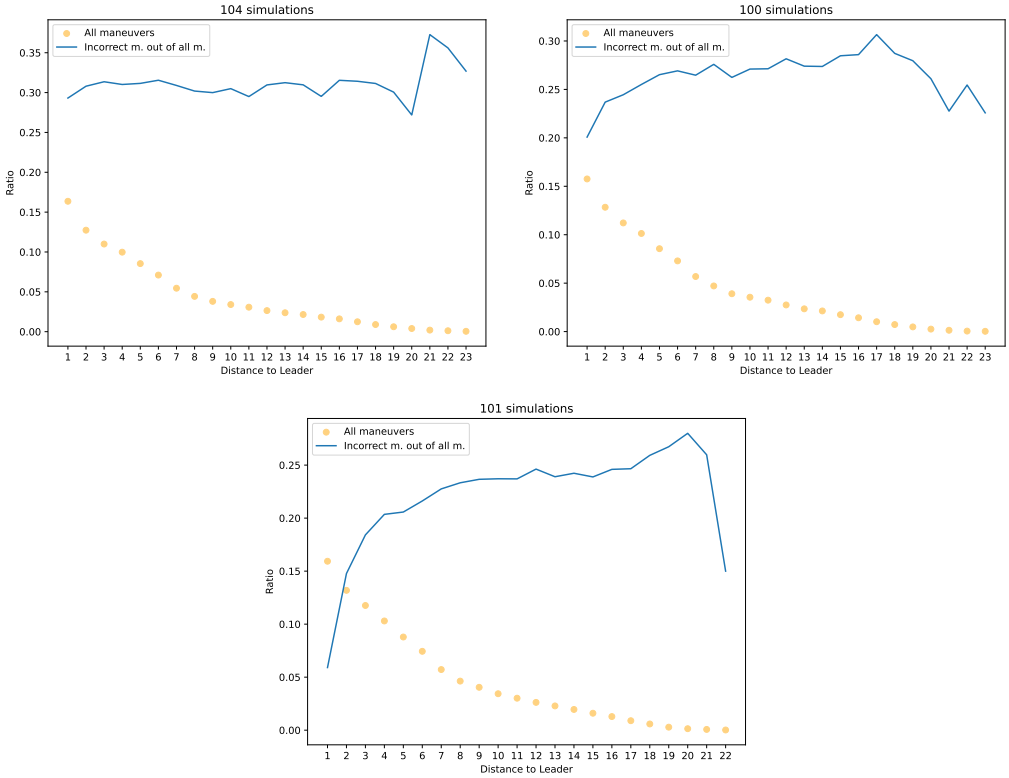

Figure 5.7: Penalization 0 in top figure, 0.5 in middle and 1 at the bottom. Ratio of maneuvers resulting in incorrect orientation decreases with higher penalization.

With penalization parameter set to 1, the ratio of incorrect maneuvers is almost not present at small distance to the leader (less than 10% of maneuver at said distance) and increases with distance where it does not cross 30% as can be seen in the Figure 5.7 in the last graph. The graphs do not show values in distances higher than 23, because too few maneuvers in this distance were spotted and the results would be skewed by stochasticity. With penalization parameter set to 0, the ratio of incorrect maneuvers is consistent across all distances to the leader, stabilised around that 30% value as is seen in the first graph. With penalization parameter of 0.5, the graph in the middle shows a decline in incorrect maneuvers for close distance but the rest of the blue line is comparable to the 0 penalization value. The yellow dotted plot of all maneuvers is comparable in all penalization parameter settings, which means that the ratio of maneuvers happening at various distances remained the same.

5.6 Specific flow

Specific flow of agents passing through three different areas explains the dynamics of the pedestrian movement. The first area, called *Gate 0* is the classroom door of 2 cell width. Every agent needs to evacuate the classroom through this door. The specific flow in this area can explain the pedestrian movement through bottleneck.

The second area, called *Gate 1*, covers the whole width of the corridor with 5 cells and is located on the x axis at position 51. This area is situated after goal location B and aims to investigate the dynamics of agents moving in corridor. These two areas are present in all map types (`map0X.txt, map1X.txt, map2X.txt`).

The third area, called *Gate 2* is present only in map with right-turn and covers the whole width of the corridor 5 cells before the exit at y position 6. Analysis of specific flow through this area will explain how passing a turn affects the structure of the pedestrian crowd as well as congestion at the exit. The detailed definition of the area positions can be found in the the class `ExperimentSpecificFlow`.

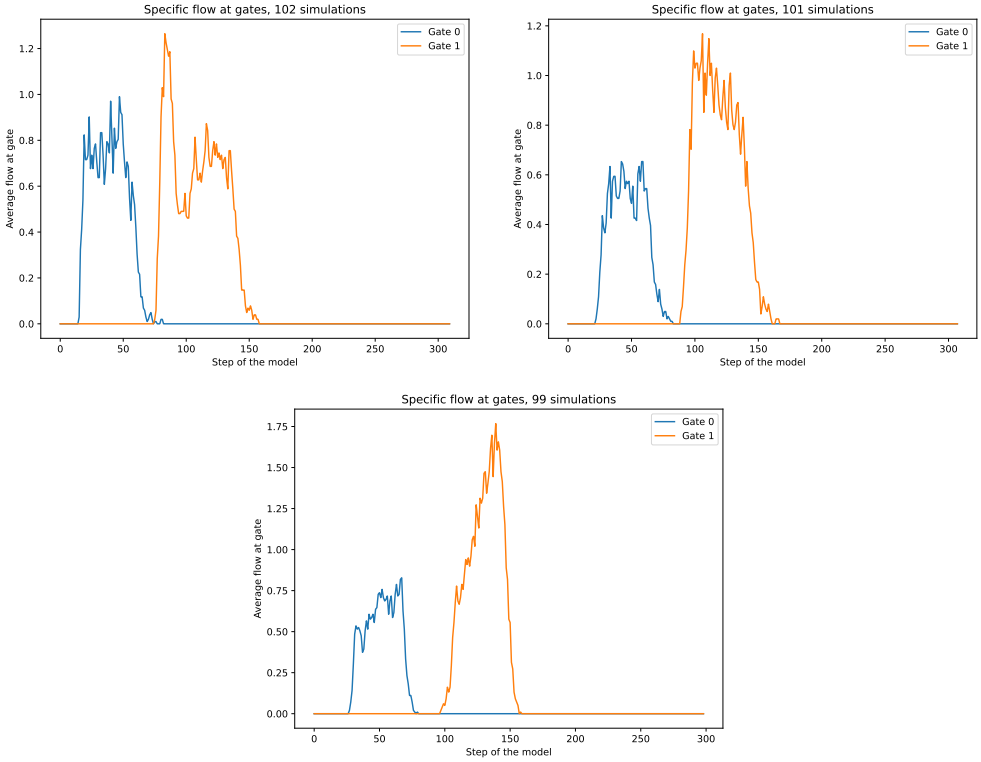

Figure 5.8: From the left, specific flow for `map01.txt, map02.txt, map03.txt`.

The three Figures in 5.8 show specific flow in area of the class door and in the middle of the corridor during three scenarios in the map `map0X.txt`. The first scenario, with leading agent navigating directly to the exit has higher flow in the *Gate 0* area. Because the agents do not form a group at location A, they leave the classroom sooner and the peaks are moved to the left. The second scenario has comparable specific flow in *Gate 1* apart from the spike. The third scenario has highest flow in *Gate 1*, which can be explained by the group being more compact and thus reaching higher local density.

The specific flow of three scenarios in the map `map1X.txt` with shorter corridor is similar to previous map `map0X.txt`. The visualized specific flow is in the Figure B.3 in Appendix.

The three scenarios in the most complex map `map2X.txt` produced specific flows shown in the Figure 5.9.

In all three scenarios, the figures show that dynamics of agent group movement did not change after passing the corner. The yellow line *Gate 1* shows the specific flow in the middle of the corridor and it is visually similar to the red line of *Gate 2* specific flow in the area after corner.

The highest specific flow can be found in the second scenario, where the agent is positioned at the back of the group after passing location B. The *Gate 1* area is located

after location B, so it is affected by the leading agent's position at the back.

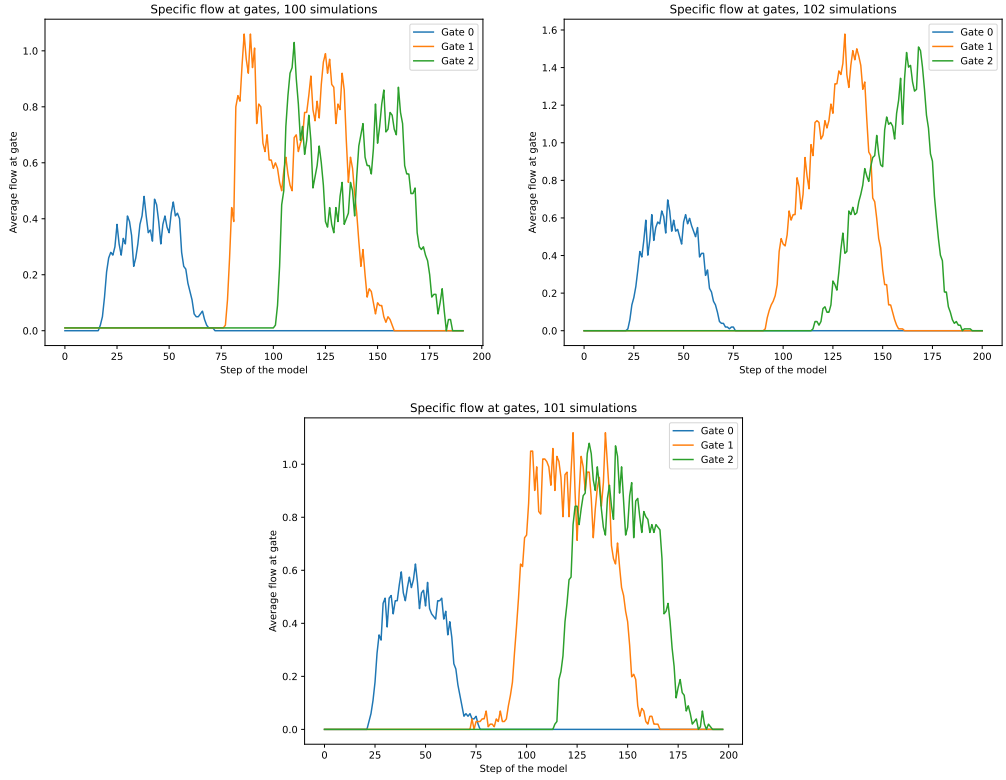

Figure 5.9: Specific flow maps 21, 22, 23.

5.7 Sensitivity to static field parameter

One of the parameters with major influence on the course and dynamics of the evacuation simulation is sensitivity to the static field k_S as was previously analysed in [5] where the study showed, that reasonable value range for this parameter lies somewhere in the range of $[1.5, 4.5]$.

The hierachical model in this study also depends on k_S and it is important to see how it affects the course of the simulation. As noted in the simulation parameters Table 4.2, there were three simulation experiments made with $k_S \in \{1, 3, 5\}$. The influence on the simulation is shown in Figure 5.10 using the heatmap of visits of cells. As k_S increased the heatmap trace became less dispersed.

With higher k_S, the agents choose the optimal cell more often and thus the movement is more deterministic. The graph on the top shows the simulation runs with low k_S, which produced erratic movement of agents across the whole width of the corridor and made the simulation much longer. Average TET was 250 ± 18.3 compared to 178.39 ± 5.68 in simulations with $k_S = 3$ and 171.43 ± 3.58 in simulations with $k_S = 5$.

whole width of the corridor and made the simulation much longer. Average TET was 250 ± 18.3 compared to 178.39 ± 5.68 in simulations with $k_S = 3$ and 171.43 ± 3.58 in simulations with $k_S = 5$.

5.8 Sensitivity to occupied cell parameter

Higher k_O prevents the agent from choosing occupied cell for next move and prefers the side unoccupied cells. As can be seen in the Figure 5.11, where with increasing k_O the

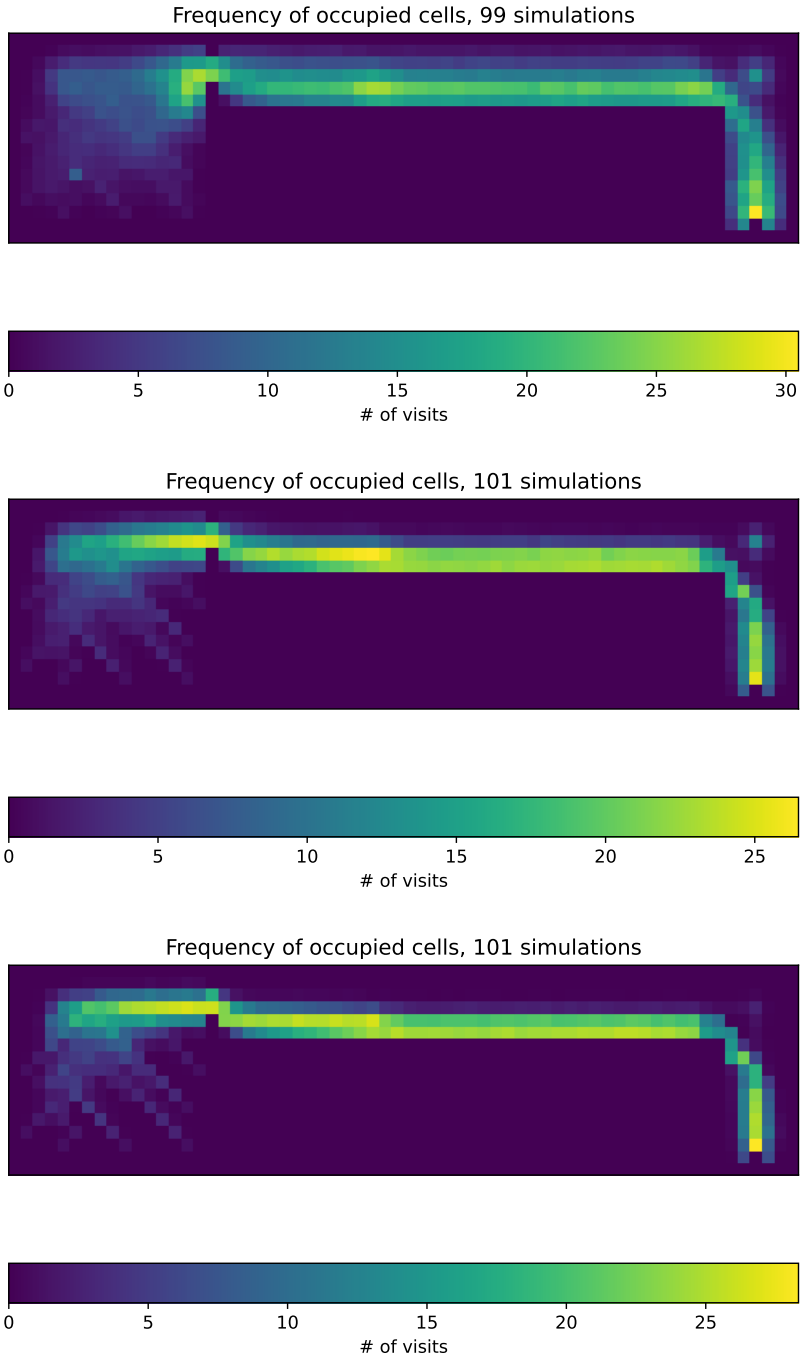

Figure 5.10: Heatmap of visits of cells in `map23.txt` with $k_S = 1$ in graph on the top, $k_S = 3$ in the middle, $k_S = 5$ on the bottom.

trace of agents made by visiting cells is more spread out and the agents do not follow the leader in packed queue but rather move in whole width of the corridor. Also, the maximal number of visits, as shows the colorbar, is higher for low k_O, which means the group of agents was more compact as they spread less and visited less unique cells.

The paired agents that move in queue need to select the cell in front of them to move in tightly packed group. Once the probability of choosing it is zero because of the $k_O = 1$, the paired agents can choose to stay in their cell and create gaps between the pairs. The gaps are present even if $k_O > 0$ as was analysed in Section 5.1 so the high k_O parameter affects the course of the evacuation mostly in bottlenecks, when pairs are grouped together

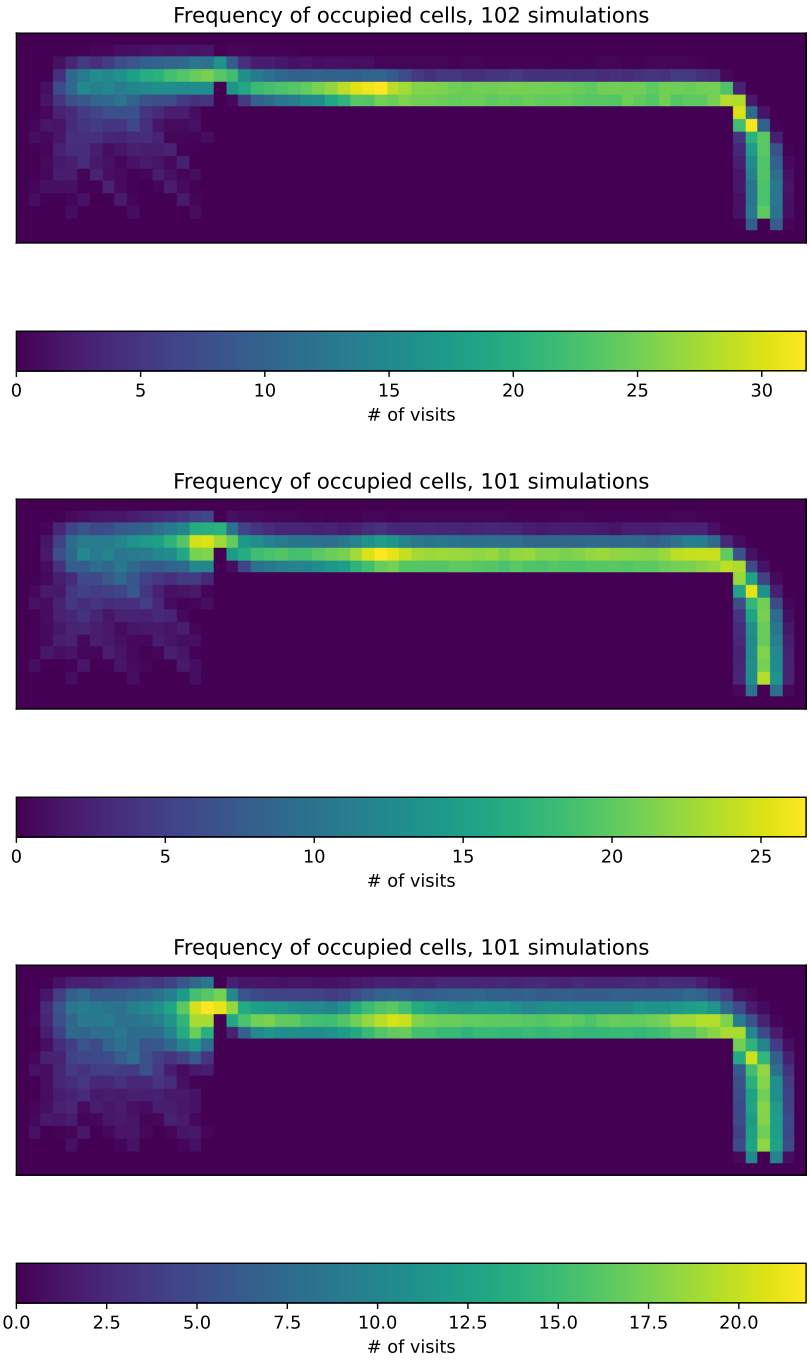

Figure 5.11: k_O with values of 0 on the top, 0.5 in the middle and 1 on the bottom in `map22.txt`.

without gaps.

Conclusion

Bibliography

[1] Najmanová, H. *Evacuation of Pre-school Children Aged from 3 to 6 Years*. Dissertation thesis, Czech Technical University in Prague, 2021.

[2] Janovská, K. *Hierarchické řízení rojů při evakuaci*. B.S. thesis, Czech Technical University in Prague, 2022.

[3] 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.

[4] Huang, R.; Zhao, X.; et al. Static floor field construction and fine discrete cellular automaton model: Algorithms, simulations and insights. *Physica A: Statistical Mechanics and its Applications*, volume 606, 2022: p. 128150, ISSN 0378-4371, doi:https://doi.org/10.1016/j.physa.2022.128150. Available from: `https://www.sciencedirect.com/science/article/pii/S0378437122007099`

[5] Šutý, M. *Conflict solution in cellular evacuation model*. B.S. thesis, Czech Technical University in Prague, 2021. Available from: `https://dspace.cvut.cz/bitstream/handle/10467/95146/F8-BP-2021-Suty-Matej-thesis.pdf`

[6] Gorrini, A.; Bandini, S.; et al. Group Dynamics in Pedestrian Crowds Estimating Proxemic Behavior. *Transportation Research Record: Journal of the Transportation Research Board*, volume 2421, January 2014: pp. 47–55, doi:10.3141/2421-06.

[7] Kazil, J.; Masad, D.; et al. Utilizing Python for Agent-Based Modeling: The Mesa Framework. In *Social, Cultural, and Behavioral Modeling*, edited by R. Thomson; H. Bisgin; C. Dancy; A. Hyder; M. Hussain, Cham: Springer International Publishing, 2020, pp. 308–317.

[8] Yap, W.; Janssen, P.; et al. Free and open source urbanism: Software for urban planning practice. *Computers, Environment and Urban Systems*, volume 96, 2022: p. 101825.

[9] Leon-Blanco, J. M.; Gonzalez-R, P. L.; et al. A multi-agent approach to the truck multi-drone routing problem. *Expert Systems with Applications*, volume 195, 2022: p. 116604.

[10] Ghanem, N.; Leitner, S.; et al. Balancing consumer and business value of recommender systems: A simulation-based analysis. *Electronic Commerce Research and Applications*, volume 55, 2022: p. 101195.

[11] Vyhláška Ministerstva MMR č. 268/2009 Sb. Online, 2009, accessed: April 21, 2023. Available from: https://mmr.cz/getmedia/2bf72909-e837-4dc8-9488-599950e8f9f6/Vyhlaska-MMR-268-2009

[12] Czech Statistical Office. České školy v číslech. https://www.czso.cz/csu/stoletistatistiky/ceske-skoly-v-cislech, 2019, accessed: April 21, 2023.

Appendix A

Acronyms

ABM Agent-based model

CA Cellular automaton

SFF Static floor field

OFF Occupancy floor field

TET Total evacuation time

Appendix B

Further graphical output

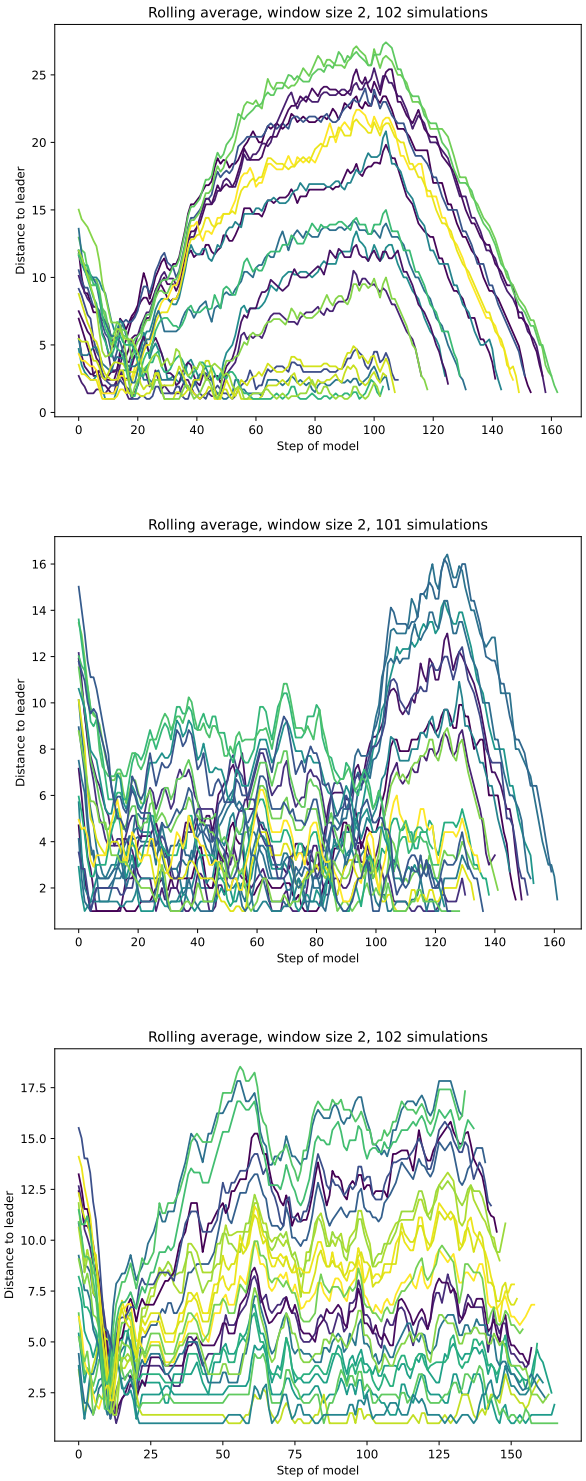

Figure B.1: Distance of agents to the leader of `map11.txt` in the top figure, `map12.txt` in the middle figure, `map13.txt` in the bottom figure.

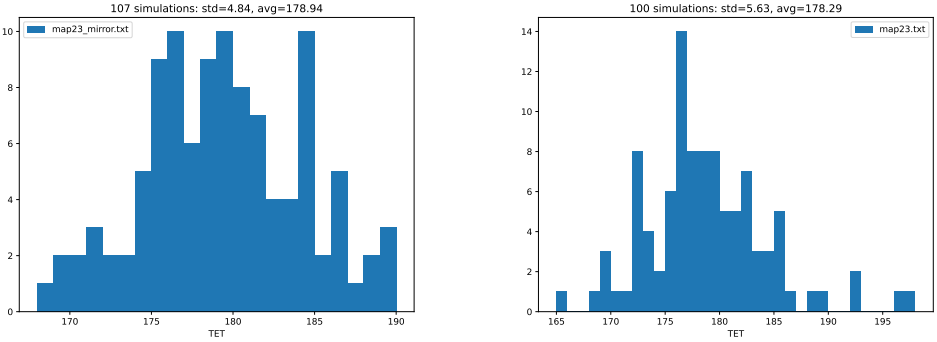

Figure B.2: TET.

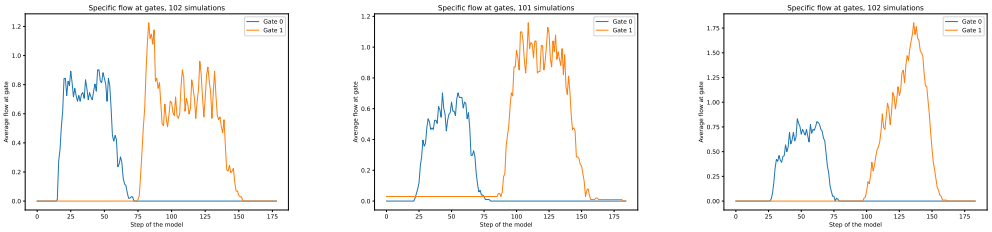

Figure B.3: Specific flow maps 11, 12, 13.

Appendix C

Contents of enclosed CD

- `readme.txt` ..... the file with CD contents description
- `exe` ..... the directory with executables
- `src` ..... the directory of source codes
  - `wbdcm` ..... implementation sources
  - `thesis` ..... the directory of LaTeX source codes of the thesis
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format
  - `thesis.ps` ..... the thesis text in PS format