UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Annika Laumets-Tättar

Modelling and Simulating Indoor Pedestrian Movement Behaviour and Displacement

Master's Thesis (30 ECTS)

Supervisor: Amnir Hadachi, PhD

Tartu 2018

Modelling and Simulating Indoor Pedestrian Movement Behaviour and Displacement

Abstract: Pedestrian evacuation routes are an important part of building's architecture. Knowing before building a structure if the evacuation routes are efficient enough should benefit both the people and the owners of the building. One's benefit is their life, the other's benefit not paying hospital and other fees for the injured or deceased. Three different approaches of pedestrian simulations are mostly used - particle-based approach (social force model), CA-based approach, and autonomous agents. The two first approaches are macroscopic models and the latter is a microscopic model. This thesis gives an overview of an implemented microscopic model, which uses autonomous agents, group modelling and social comparison theory to evaluate building egress design. With microscopic models each agent can have different attributes and ties with other people in the vicinity (e.g. friends and family). People use social comparison theory (SCT) in their daily lives to compare themselves to others. In an evacuation situation similar people form groups by social comparison theory and influence each other. The model is implemented using Python, SUMO and TraCI. In this thesis a case study is done on Ülemiste center in Tallinn, Estonia using the implemented model. Different number of exits and pedestrians are tested to see how they are associated. The results show that the farther apart the exits are, the smaller the evacuation times are. The same results appears with only one exit and all exits. Bottlenecks near the exits closer together are the reason for higher evacuation times. Agents also prefer exits farther apart when analysing the distribution between exits. Speed does not affect the evacuation time as much as expected with higher number of pedestrians because the density of the crowd disallows the agents to reach their high speed. The final outcome is that with speeds 1.4 m/s and 2.5 m/s 90% of people out of 200 agents exit the building around 400 seconds. This kind of result happened with all exit configurations. The other 10% might be older or confused people who are not familiar with the building's floor plan. Majority of people are, therefore, safe in 6 minutes and 40 seconds, which seems quite realistic for 200 people. The model implemented can be used to assess and evaluate a building's egress design before the structure is actually built. It can help design better evacuation routes for buildings because a user-specified floor plan can be used by the model.

Keywords:
Indoor pedestrian simulation, autonomous agents, social comparison theory, group modelling

CERCS: P175 Informatics, systems theory

Siseruumides inimeste liikumise käitumise ja muutumise modelleerimine ning simuleerimine

Lühikokkuvõte: Inimeste evakuatsiooniteed on hoonete ülesehituse üks tähtsamaid osasid. Nii inimestele kui ka hoone omanikutele tooks kasu, kui enne hoone ehitust oleks teada, kas evakuatsiooniteed on piisavalt efektiivsed. Inimeste kasu oleks nende elu ja hoone omanike kasu oleks kahjunõute puudumine vigastatud inimestelt või surnute lähedastelt. Inimeste liikumise simuleerimiseks kasutatakse üldiselt kolme erinevat meetodit - osakestepõhine lähenemine (ühiskondliku jõu mudel), mobiilsideautomaadi põhine lähenemine ja autonoomsed agendid. Kaks esimest meetodit on makroskoopilised mudelid ning viimane meetod on mikroskoopiline mudel. See lõputöö annab ülevaate arendatud mikroskoopilisest mudelist, mis kasutab autonoomseid agente, grupi mudelleerimist ning sotsiaalse võrdluse teooriat, et hinnata hoone väljapääsude disaini. Mikroskoopiliste mudelitega võib igal agendil olla erinevad omadused ning seosed läheduses asuvate inimestega (näiteks sõbrad ja pere). Inimesed kasutavad sotsiaalse võrdluse teooriat oma igapäevaelus, et end teistega võrrelda. Evakuatsiooni olukorras moodustavad sarnased inimesed, kasutades sotsiaalse võrdluse teooriat, gruppe ja mõjutavad üksteist. Mudel programmeeriti kasutades Pythonit, SUMO-t ja TraCI-t. Lõutöös tehtud simulatsioonid kasutavad näitena Ülemiste keskust, mis asub Tallinnas, Eestis. Erinevate arvu väljapääsude ja inimeste vaheliste seoste jaoks viiakse läbi mitmeid simulatsioone. Tulemused näitavad, et mida kaugemal on väljapääsud üksteisest, seda väiksem on evakuatsiooni jaoks kuluv aeg. Samad tulemused saadi ka ainult ühe väljapääsuga ning kõikide väljapääsudega. Lähestikku asuvate väljapääsude juurde tekkivad kitsaskohad on põhjuseks ka pikematele evakuatsiooniaegadele. Kui uurida väljaspääsudest väljuvate inimeste jaotust, siis eelistavad agendid samuti üksteisest kaugemal asuvaid väljapääse. Inimeste kiirus ei mõjuta evakuatsiooniaegu nii palju, nagu alguses arvatud suurema arvu inimestega, kuna rahvahulga tihedus ei luba inimestel saavutada kiiremat kiirust. Viimane tulemus on see, et kiirustega 1.4 m/s ja 2.5 m/s suudab 200 inimesest 90% hoonest väljuda keskmiselt 400 sekundi jooksul. Selline tulemus tuli kõikide erinevate väljapääsudega. Viimased 10% võisid olla vanurid või segaduses inimesed, kes poole hoonega tuttavad. Järelikult on suurem osa turvaliselt hoonest väljunud 6 minuti ning 40 sekundiga, mis tundub 200 inimese puhul üpris reaalne tulemus. Arendatud mudelit saab kasutada hoone väljapääsude disaini hindamiseks enne hoone ehitamist. Selle mudeli kasutamine võib aidata kavandada paremaid evakuatsiooniteid, kuna kasutaja saab mudelile sisse anda erinevate hoonete plaane.

Võtmesõnad:
Siseruumides jalakäijate simuleerimine, autonoomsed agendid, sotsiaalse võrdlemise teooria, grupi modelleerimine

CERCS: P175 Informaatika, süsteemiteooria

Contents

Acknowledgement

I would like to thank my supervisor Dr. Amnir Hadachi for suggesting me such an interesting topic and always helping me when I didn't fully understand the models or the formulas behind it.
I would also like to thank my husband Andre Tättar, who gave me interesting ideas what kind of simulations to test out besides the ones I could think of myself. It really was fascinating to see what results different kinds of situations generate.

Abbreviation and Acronyms

1. CA - Cellular Automata
2. SCT - Social Comparison Theory
3. SUMO - Simulation of Urban MObility (application)
4. TraCI - Traffic Control Interface (online interaction with the SUMO simulation)

1 Introduction

1.1 General view

Pedestrian simulation is known for its unpredictability and dynamic characteristics. Because of this, the field of pedestrian simulation is still lively and not fully understood. Many articles look at some specific configurations and/or area of movement. For example, the article by Vizzari et al. [VMC13] looks at different-shaped corridors to see how it affects pedestrian's movements and what parts of the areas are more used than others. Traditionally, evacuation simulation is used for fire evacuations. Nowadays there are many more threats where pedestrians need to be evacuated - bomb threats, terrorist attacks, shooters and so on. These threats create a need for efficient evacuation of buildings. Many research articles simulate evacuations, like [SBK04][PHDL07][LH15], but they do not use an actual floorplan of a building. It could be useful for egress design of buildings to evaluate beforehand the effect it will have on evacuation times.

1.2 Objectives and limitation

The objective is to implement a pedestrian model with autonomous agents using SUMO and TraCI. With the model, the number of agents and exits should be specifiable to assess what effect they have on the duration of evacuations. The speed of pedestrians also plays a role because in a dangerous situation people generally move more quickly than when just walking around. In the future, the model can be used before building a structure to assess the quality of the egress design and to see if it is efficient. Because the model is microscopic, the results are not instant, but they should be more accurate than macroscopic models because each pedestrian has different features - the speed, how much space they cover, preferred exits, ties with other pedestrians (e.g. family, friends). The objective is to implement some of these features in this thesis.

1.3 Contribution

The implemented microscopic model can be used to simulate pedestrians moving inside a building with different speeds. The model uses group modelling and social comparison theory to determine the routes for pedestrians. A case study with Ülemiste centre in Tallinn, Estonia is conducted to find out the effect of the different number of exits on evacuation time. Other floor plans can be used with the model to estimate the evacuation time. It can help to assess if there are enough exits for a specified number of pedestrians before building the structure. It can save people's lives. Furthermore, there are only guidelines for evacuation egress for buildings, but since every building is different no requirements that fit all are written in the law. With the model, the building egress can be tested to see whether it is suitable for the purpose of the building.

1.4 Road Map

In Section 2, the current state-of-the-art pedestrian simulation models are presented. Firstly, the socio-physiological aspect is introduced and a microscopic way of modelling using groups is presented. The applications used to implement the model in this thesis are also introduced. In Section 3, the methodology can be found. The articles inspiring the model are summarised. The model and the implementation are explained. Section 4 shows the different results of the simulation and discusses them. Conclusion and future perspectives can be seen in Section 5.

2 State-of-the-art

2.1 Introduction

There are mostly three categories of pedestrian models - particle-based approach, Cellular Automata (CA) approach and autonomous agents [VMC13]. The social force model is the most adopted particle-based approach. The model usually employs the tendency for pedestrians to stay away from each other while moving towards his/her goal. CA-based approaches are based on a discrete representation of the simulated environment which is divided into cells. Each cell can be either free, occupied by an obstacle or occupied by a pedestrian. The CA-based approach uses a rule, which determines the next state of each cell and considers its current state and the states of nearby cells. Autonomous agents are usually used to visualise believable crowd dynamics instead of quantitative results generated by particle-based and CA-based models. It seems that agent-based approach is the way to go. However, Zhou et al. [ZCC+10] found that one of the issues of agent-based models is the shortcoming of social group process and its impact on human behaviour. They concluded that in an emergency evacuation situation a person may influence others and may be influenced by other persons in the same group. This phenomenon was especially noticeable with people with ties (e.g. family, friends). When dealing with strangers groups can also be formed because the persons share the same goal. This notion of similarity is called Social Comparison Theory (SCT), and it was described by Festinger [Fes54].

2.2 Social Comparison Theory

Social comparison theory (SCT) is a much-used theory for pedestrian simulations. SCT is based on the idea that humans compare themselves to others that are similar. Moreover, not only Festinger investigated about social comparison, but Goethals and Darley [RGMD87] also describe the connection to "Related Attributes Hypothesis" which means that people will compare themselves to others that are similar to them on attributes that are related to their opinion or performance. Goethels and Klein [RGK00] also provide an example which directly admits surface comparisons about a tennis player who will compare his/her ability to others who are the same sex, about the same age, and who are at the same skill level and use comparable equipment. Horstein et al. [AHFH68] point out that people use SCT to decide whether to return a lost wallet, which means that people use SCT in their everyday lives to compare one to others on the surface. Fridman and Kaminka proposed a model based on Social Comparison Theory [FK10]. The article proposes a concrete algorithmic framework for Social Comparison Theory and evaluates its implementation. The results show that that SCT generates behaviour which is more in-tune with human crowd behaviour than other similar models.

2.3 Group modelling

Autonomous agents seem to work well, but the approach has had better performance and more human-like behaviour with group formation and cohesion. It seems logical that if friends or family are present in the same environment at the same time, that persons will try to reach their destination, but at the same time try to preserve a limited distance from other group members. The paper by Collins et al. [CEFR14] discusses the importance of groups using an agent-based model. The safety and effectiveness of pedestrian egress is an important factor in that paper. They propose a model where pedestrian agents have two goals instead of one. The first goal is to exit the venue and the second goal is to maintain a level of cohesion within their group. Two different approaches to updating the agent headings are used - the weighted averaging approach of social forces by Helbing et al. [HM98] and a discrete stochastic selection approach between the two goals. The conclusion is that the updating of simulated pedestrian headings becomes less effective using the traditional method of social forces when pedestrians belong to a group where cohesion with other members is more important than heading to the exit because the exits can get blocked by pedestrians trying to regroup. Therefore, the new model, which randomly chooses a goal to head towards, was to overcome the previously mentioned issue.

Other researchers have also incorporated group cohesion into agent-based models. Vizzari et al. [VMC13] developed an agent-based model, which uses both the traditional method of staying away from other pedestrians while moving and an adaptive mechanism, which represents the influence of the group presence. The preservation of cohesion of groups (e.g. family, friends) in high-density situations is the main goal. When using scenarios of not comprising groups, the model produces results similar to available evidence from literature (both from the perspective of pedestrian flows and space utilisation). The model also preserves the cohesion of groups in challenging situations when groups are present. Agent-based models are not the only approach where group modelling has gained popularity. Particle-based models and CA-based models using group modelling can also be found in literature, but they are out of the scope of this thesis [MPG$^+$10][XD10][SHZHT09].

2.4 Simulation of Urban Mobility

The most important part while choosing a simulator was that it would be open source and that microscopic models could be used. The choice fell on Simulation of Urban MObility (SUMO) because it ticked all of the two boxes mentioned earlier [KEBB12][sum]. SUMO is licensed under the Eclipse Public License V2 and, therefore, it is open source. It also has high portability because only standard C++ libraries and portable libraries are used. On the SUMO webpage, there are packages for downloading for Windows and Linux operating systems. SUMO is a multi-modal traffic simulation - vehicles, buses, trains, bicycles, and pedestrians can be used in the simulation. The simulation is microscopic, which makes it easy to implement agent-based models. Moreover, SUMO can handle large networks with many vehicles. Therefore, it is easy to use on a personal machine. The SUMO package also contains many other applications. Some of the functions are used to import or prepare road networks, some compute routes through the network using different techniques, and some generate demand and information about emission.

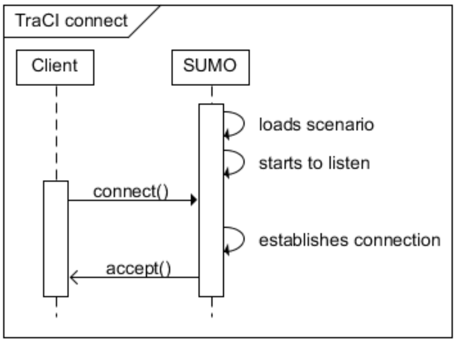


Figure 1. TraCI establishing a connection to SUMO [WPR+08][WHWL08]

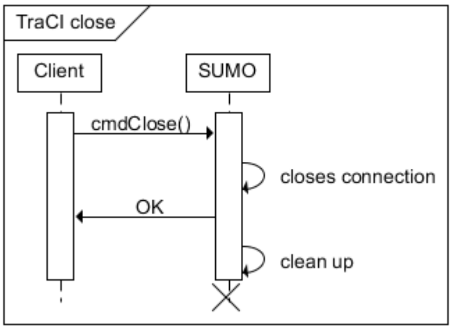


Figure 2. TraCI closing a connection to SUMO [WPR+08][WHWL08]

Traffic Control Interface (TraCI) is needed to interact with the simulation while it is running [WPR+08][WHWL08]. TraCI uses TCP-based client/server architecture to provide access to SUMO. In Figure 1 the architecture for establishing a connection can be seen and closing the connection is shown in Figure 2. SUMO acts as a server when communicating with TraCI. Multiple TraCI clients can connect to one SUMO server. With TraCI, it is possible to retrieve values of simulated objects and access their genetic parameters, change their state, and subscribe to objects during the simulation runtime. Performing simulation steps and reloading the simulation is also possible with TraCI.

2.5 Conclusion

In this section, the different types of current pedestrian models were introduced. Social Comparison Theory and group formation were explained. Together with traditional methods, they seem to give more life-like results than just traditional methods alone. Applications that will be used to implement the agent-based model - SUMO and TraCI - were introduced and their primary purpose explained. The background in this section helps to understand the following sections.

3 Methodology

3.1 Introduction

As seen in section 2 many models could be used and implemented for SUMO. The inspiration for this thesis comes from two specific articles. The first article is by F. Qiu and X. Hu [QH10a] and the second article is by Zhang et al. [ZZL09]. Next, short summaries of the articles are provided, and the exact methods used in this paper are described.

3.2 Dynamic group modelling

The first article that inspired this thesis is by F. Qiu and X. Hu [QH10a], where they present a new model to simulate the movement of pedestrians using both utility theory and social comparison theory. Most of the models mentioned in Section 2 used one or the other. The argument to describe a new model is that "most existing work simulates groups based on socio-psychological theories such as social comparison theory and five-factor personality theory". However, these models only describe the dynamics of pedestrians' socio-psychological states. The new model uses two steps – the first step is choosing a new group to follow or to stay with the old group and the second step is choosing the individual in the group to follow. For the first step utility theory is used. It assumes that "person, even in a dangerous situation, can still make rational decisions" and it will choose the action with higher utility. For the second step, Festinger's social comparison theory is used, which was described in Section 2.2. There are also multiple variables in the model that can affect the outcome – the distance to the group, sociality of the pedestrian agent, duration the agent has stayed with a group, the time the pedestrian agent can change the group (threshold), similarity of the agent which is bounded by $Smin$ and $Smax$. Equations 1 - 5 show the formulas that are used to find which group an agent should follow or if the agent should stay with the old group. $DesiredDist$ and c are constants. $Duration$ is the number of time steps an agent has stayed with a group. After following a new group, $Duration$ will be set to zero. $Threshold$ is the maximum number of time steps during which an agent can join another group. Equation 1 shows the distance between agent i and group GP_j using the closest agent j who belongs to group GP_j. Equation 2 shows that the smaller the distance between agent i and group GP_j, the bigger variable t will be. t is, however, used in Equation 3 which finds the maximum utility U_f of joining other groups. Therefore, the closer the group is, the more likely the agent will join this group.

$$Dist_{i,GP_j} = Dist(i,j) \tag{1}$$

$$t = DesiredDist/Dist_{i,GP_j} \tag{2}$$

$$U_{i,f} = max(t * e^{Sociality_i*(1-Duration/Threshold)}) \tag{3}$$

U_s is the utility to stay with the current group, and the formula is in Equation 4. $Sociality_i$ shows how social the agent i is. The higher the value is, the more likely the agent will join another group.

$$U_{i,s} = c * e^{Sociality_i*Duration/Threshold} \tag{4}$$

Equation 5 shows the conditions whether to follow another group or stay with the current one.

$$GroupToJoin = \begin{cases} \text{Group with } U_{i,f}, & \text{if } U_{i,f} > U_{i,s} \\ GP_s & \text{otherwise} \end{cases} \tag{5}$$

In Equations 6 - 10 a and b are constant numbers. i and j are agents. If agent i decides to follow a new group, $Similarity$ is calculated using the formula from Equation 6. It is the agent's similarity value, which finds the similarity between agents i and j. Moreover, the $Similarity$ value is bounded by predefined minimum value $Smin$ and maximum value $Smax$.

$$Similarity_i = Sociality_i * D_1 * (a * D_2 + b * e^{1-Dist(i,j)/Dist}) \tag{6}$$

Equation 6 uses Equations 7 and 8. In Equation 7 v_i and v_j are the velocities of agents i and j. In Equation 8 "v_l and v_m are the vectors pointing from from i's current position to the position of the selected member and i's destination, respectively" [QH10a]. The moving direction is important, because an agent will prefer to follow an agent who moves in the same direction. Moreover, the more social the agent is and the smaller the distance between agents i and j, the more likely agent i will join the group where agent j belongs to.

$$D_1 = \begin{cases} 1 & \text{If } v_l * v_m \geq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{7}$$

$$D_2 = \begin{cases} 1 & \text{If } v_i * v_j \geq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

After finding all of the $Similarity$ values between agent i and agents of the selected group, the agent with the maximum $Similarity$ value from the group is found.

$$MostSimilarId = max(Similarity_1, \ldots, Similarity_n) \tag{9}$$

If this agent exists, agent i will follow the selected agent and, therefore, the selected group. Otherwise, the agent will stay with the old group. Equation 10 shows the rules for these cases.

$$AgentToFollow = \begin{cases} Agent \text{ with } MostSimilarId \text{ If } MostSimilarId \text{ exists} \\ \emptyset \text{ Otherwise} \end{cases} \tag{10}$$

Also, the pedestrian agent has “a set of attributes which characterise its internal states” and three behaviour models – random movement, obstacle avoidance and maintaining group. Random movement behaviour is used to generate a random destination for the pedestrian and move the agent there. Obstacle avoidance behaviour ensures that the pedestrian does not collide with obstacles, other agents and groups. The agent follows a member in a selected group with maintaining group behaviour. The experiments made show that the developed model can simulate dynamic grouping. Moreover, a second experiment was made trying different threshold values to see how it affects the group changing rate. The outcome was that the bigger the threshold, the more times agents change groups. In conclusion, a two-layer model was developed, where, firstly, the pedestrian agent chooses which group to follow – follow a new group or stay with the old group – and then the agent who is most similar to himself is chosen to follow.

In this thesis, the formulas for following and staying with the group will be used. SUMO already has a built-in obstacle avoidance, which means that it will not be implemented. Random movement behaviour will be implemented by generating a random destination for the agent. If the user gives destination edges, a random edge from these is selected. Maintaining group will also be implemented by changing the pedestrian’s route to the destination so that the pedestrian can follow the group.

3.3 Pedestrian egress safety

The second article that that was used as a basis is by Zhang et al. [ZZL09], where egress safety was analysed by using different situations. Since building codes “only provide basic guidelines and are not exhaustive” and each building is different, simulations should be conducted to test the safety and operation of egress. They used a multi-agent system instead of system theories of organisation because the latter is quite abstract. With multi-agent systems individual agents can have different and complex behavioural patterns. Furthermore, agent behaviour can be assessed by the user with multi-agent system simulations. The casualty rate was assessed in different scenarios using force modelling. The experiments included “normal and emergency situations to show that the crushing force between individuals significantly affects the results”. In addition, there were three different cases of emergency evacuation – evacuation with no barriers, evacuation with barriers and various egress distances. In a normal crowd evacuation, there were no casualties and the total evacuation time for 2m wide exit was 93.9 seconds, for 4m wide exit it was 59.2 seconds, and for the 8m wide exit it was 43.3 seconds. The fatality probability for 2m wide exits was 75%, less than 10% for 4m wide exits and 0%

for 8m wide exits in the emergency evacuation with no barriers. The evacuation time for 2m exits was 53.5 seconds, for 4m exits 32 seconds and 21.8 seconds for 8m exits. With barriers, the evacuation times were only slightly increased, but the fatality probability was significantly increased. The last experiment tested various gaps between exits. The 2m, 4m and 8m exits were distanced 10, 30, and 50 meters apart. The test results showed that for 2m wide and 4m wide exits the exit distances did not have an obvious effect. However, for 8m wide exits the evacuation time for the 30m distance between the exits was a bit smaller than for the other distances. In conclusion, it seems that the 30m exit distance is the most reasonable solution.

In this thesis, the distance between exits will be used to see how it affects the evacuation time. Assessing different widths for exits could be used in future work.

3.4 Model implementation

The implemented model is divided into four files. The first file has the main method, the second file holds the Agent class, the third file has the Group class, and the last file contains the constants. In the next subsections, the files are further discussed.

3.4.1 Agent class

The $Agent$ class is a child class of TraCI's class $PersonDomain$. When initialising an $Agent$ instance, multiple parameters are created for the instance. As discussed in Section 3.2 every agent has randomised $Sociality$ and $Similarity$ and also the $Duration$ of staying with the current group. Every agent has a unique ID, and their origin and destination edges are saved as well. A group with the same ID as the agent is also created when initialising an $Agent$ object. For parameters ID, $Sociality$, $Duration$, $GroupID$, $GroupObject$, $StartEdgeID$, $StartEdgeObject$, $DestinationEdgeID$, $DestinationEdgeObject$ the get-method is implemented. The $Duration$ parameter can be increased or reset to zero with the implemented methods. The origin and destination edges can be set using the implemented methods. After initialising an $Agent$ object, the origin and destination edges are randomly generated, or if they are given as parameters, those parameters are used.

Many of the methods are the same as in the class $PersonDomain$, but they require the string ID of agents. To use the methods with instances, reimplementing the methods was required. The methods that were reimplemented are the following:

- $add()$, which adds the agent to the simulation,
- $getPosition()$, which returns the position of the agent in x and y coordinates,
- $getRoadID()$, which returns the ID of the edge the agent was on at the last time step,

- $getSpeed()$, which returns the speed of the agent within the last step in m/s,
- $getLanePosition()$, which returns the position of the agent along the lane at the last time step measured in metres,
- $getAngle()$, which returns the angle that the agent is facing within the last step in degrees,
- $getColour()$, which returns the RGBA colour of the agent,
- $getEdges()$, which returns a list of all edges in the n-th next stage. The TraCI documentation also says that for different actions, different items are returned. For example, for waiting stages a single edge is returned; for walking stages the complete route is returned; for driving stages the list is [origin, destination] [tra],
- $getNextEdge()$, which returns the next edge on the agent's route if the agent is walking; otherwise an empty string is returned,
- $getSubscriptionResults()$, which returns the subscription results for the last timestep for a given agent,
- $setColour()$, which uses the colour of the group of the agent to set the colour of the agent in the simulation,
- $subscribe()$, which subscribes to the given agent for a given time interval,
- $removeStage()$, which removes the n-th next stage,
- $moveToXY()$, which places the agent at given x, y coordinates.

Since the inspiration comes from the article in Section 3.2, $follow()$ and $stay()$ methods are implemented as well. $stay()$ method checks if the given group ID is the same as the agent's and if it is, the $Duration$ is increased. With the $follow()$ method, the $Group$ parameter for the agent is changed, and the agent is coloured according to the colour of the new group. Furthermore, the agent is added to the group's user list. After changing the group, the agent changes its route to follow the most similar agent in the selected group. For this $calculateRoute()$ method was written that finds the route between two edges. The method uses TraCI's method $findIntermodalRoute()$, which reads origin and destination edges and calculates the currently fastest route for the agent.

The algorithm for $calculateRoute$ can be seen in Algorithm 1.

Algorithm 1: $calculateRoute()$ algorithm

```
Data: A list with edge ID's called $edges$
Result: Route between given edges in the $edges$ input
if less than two edges then
    print("There must be atleast 2 edges")
    return
start = first edge in $edges$ list
destination = last edge in $edges$ list
if more than two edges in $edges$ then
    middle = edges between the start and destination edges in the $edges$ list
    route = None
    foreach edge in middle do
        route += findIntermodalRoute(edge, next edge)
    add route to agent's stages
else
    route = findIntermodalRoute(start, destination)
    add route to agent's stages
```

For the sight of the pedestrian, a fan-shaped field of vision is used. This field of vision is supposed to copy the actual perception field of a human. Similar to the work of [QH10b] and [AIK10] the perception model specifies an elliptical area which each pedestrian can perceive, as shown in Figure 3. In the Figure, $Direction$ is the current moving direction of the pedestrian. $Dist1$ and $Dist2$ are the maximum front and side distance for visibility respectively. $Angle$ is half of the maximum visibility range the pedestrian can detect.

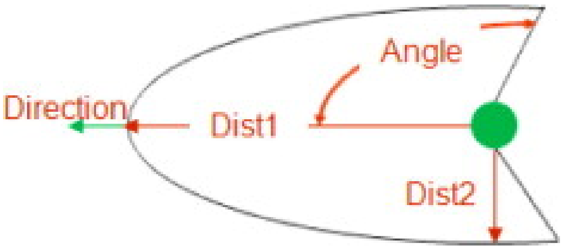


Figure 3. Agent's perception model [QH10b]

Each pedestrian has their perception field which they use to detect other pedestrians and obstacles. In this thesis, the $Angle$ is 180°. The equation $\cos^2(\theta) + \sin^2(\theta) = 1$ is used to find the ellipse used for the perception field. Only one radius is used - $Dist1$. The algorithm for the $perceive()$ method using the described model can be seen in Algorithm 2.

Algorithm 2: $perceive()$ algorithm

```
Data: Network file, the position of the agent, the angle of the agent
Result: ID's of perceived agents
dist1 = 20 * radius
dist2 = 6 * radius
x, y = position
/* Get the lanes and edges in agent's perception field */
lanes = getNeighboringLanes(net, x, y, angle, dist1, dist2)
edges = $\varnothing$
foreach lane in lanes do
    edge = getEdgeID(lane)
    edges += edge
/* Collect IDs of agents on the perceived edges */
perceived IDs = $\varnothing$
foreach edge in edges do
    IDs = getLastStepPersonIDs(edge)
    perceived IDs += IDs
remove agent's own ID from perceived IDs
```

3.4.2 Group class

The $Group$ class has the initialisation method and five other methods. When initialising a new group instance, an ID, the new user and the colour of the group are given. The ID of the group is the ID of the agent because, in the beginning, every agent has their own group. The agent is also added to the users list. The colour of the group is to visualise the different groups in the simulation. To not be able to change the parameters by just using the name of the parameter, all of the parameters are private. The 5 other methods in the Group class are $getID()$, $getUsers()$, $getColour()$, $removeUsers()$, and $addUser()$. $getID()$ function is to get the ID of a Group instance, $getUsers()$ function is to get the Agent objects belonging to the Group instance, $getColour()$ returns an RGBA colour. $removeUsers()$ removes users that are not in the simulation anymore or have joined another group. $addUser()$ function adds an agent to the group if the agent is already not a part of the group.

3.4.3 Main method

Firstly, all of the important constants are in the $constants.py$ file. For example, the file holds $Smin$ and $Smax$ parameters that are the limits when randomly selecting the $Similarity$ value for agents. Other constants are $Threshold$, $Desired\_Dist$, A, B and C which are used to find the most similar agent j in another group when following a new group or deciding whether to stay with the current group. $Radius$ is used for the perception model and the default pedestrian speed is controlled by $Speed$.

The main method has command-line parsing. The user has to specify the configuration file that is used for the simulation. The other arguments are optional. For example, the user can specify the number of agents in the simulation, one or multiple destination edges for all agents, whether to save the simulation states to file, and whether to run different tests on the simulation. After reading the network, the beginning time, the ending time, and the step length values from the configuration file, the agents can be generated. Every agent gets a randomly generated colour when initialised, and a new group is also created using the agent's ID. The origin edge and a random position on the edge are randomly selected from all edges in the network for the agent. If the user has specified a destination edge(s) for all agents, one destination edge is randomly selected for the $Agent$ instance. Algorithm 3 shows the pseudocode.

Algorithm 3: Creating agents and adding them to the simulation

```
Data: Network file, number of agents, destination edge(s)
Result: Agents added to the simulation
foreach n in number of agents do
    agent = Agent()
    agent.colour = randomly generated RGBA
    agent. group = create a new Group instance
    agent.start = randomly selected from network file
    if destination edges given as parameter then
        agent.destination = randomly selected one edge from the parameter
    else
        agent.destination = randomly selected from network file
    add agent to the simulation
```

After adding the agents to the simulation, it is possible to use TraCI's commands to find routes between edges. If the user specified destination edge(s), the closest destination edge for the agent's starting position is found by using TraCI's $findIntermodalRoute()$ method. Finally, the route for the agent is produced using method $calculateRoute()$ shown in Algorithm 1. Algorithm 4 shows the pseudocode for the implemented main method. It uses Equations 1 - 10 from Section 3.2.

Algorithm 4: Algorithm for dynamic group modelling

```
Data: Network file, agents
foreach agent in agents do
    if agent's $Duration$ > $Threshold$ then
        continue to the next agent
    perceived agents = $perceive()$
    perceived groups = collect group IDs from perceived agents
    find the closest $agent\ j$ for all of the perceived groups
    foreach $agent\ j$ do
        find distance between $agent$ and $agent\ j$ using Equation 1
        calculate Equation 2
    calculate staying utility $U_s$ with Equation 3
    if number of $agent\ j$'s > 0 then
        find the most similar $agent\ j$ with Equation 4
        if following utility $U_f$ > staying utility $U_s$ then
            $agent$ follows $agent\ j$'s group and changes colour
            calculate new route for $agent$
            reset $agent$'s $Duration$ to zero
        else
            stay with the old group
            increase $agent$'s $Duration$
    else
        stay with the old group
        increase $agent$'s $Duration$
```

Moreover, if the test argument is set to True, the duration of the simulation until there are no active agents or the duration is 7200 seconds is written to a file. When the simulation finishes, TraCI closes its connection to SUMO.

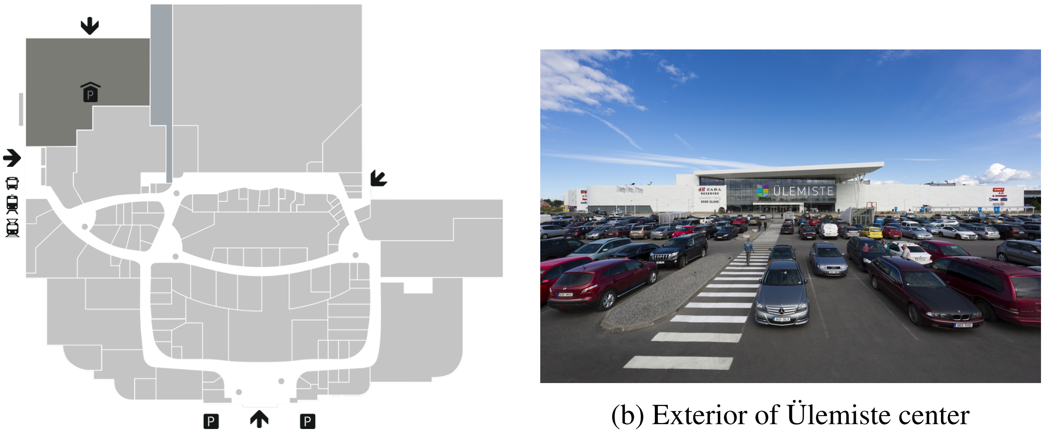
3.5 Conclusion

The model inspired by articles summarised in Sections 3.2 and 3.3 was implemented using the algorithms and descriptions in this section. Python programming language, SUMO, and TraCI were used to implement the model. In the next section, different tests and benchmarking are described and conducted. The results will be discussed.

4 Results and discussion

4.1 Introduction

The newly implemented model needs benchmarking to see how it reacts with the different number of exits and pedestrians. Benchmarking shows how good the model performs and if it seems real to actual evacuations. Furthermore, the distribution of agents between the exits is calculated to see if it affects the evacuation time. For the simulation, a real building is used. The building is Ülemiste centre in Tallinn, Estonia. The exterior with the main exit and the floor plan are shown in Figure 4.



(a) Floorplan of Ülemiste center

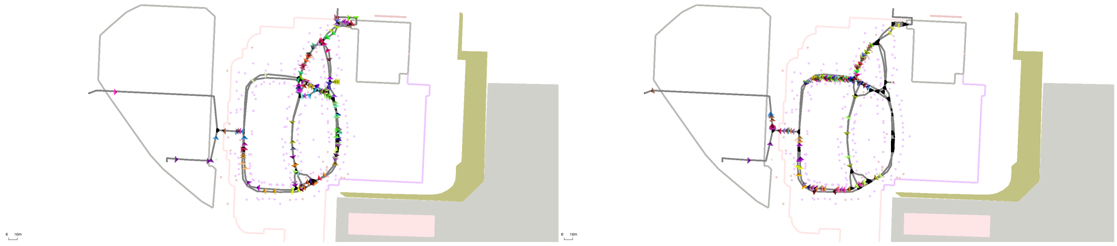
(b) Exterior of Ülemiste center

Figure 4. Floorplan and exterior of Ülemiste center [yle]

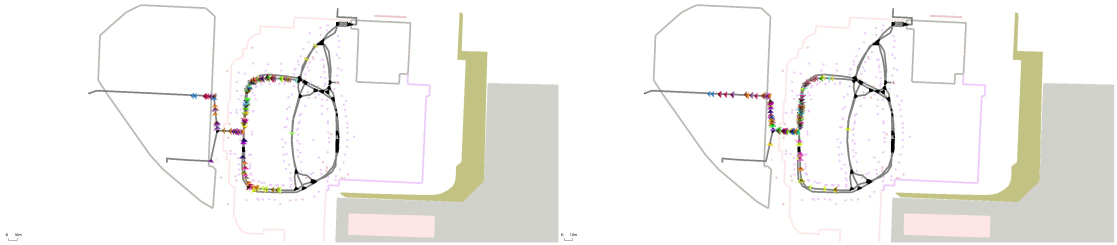
The simulations use a network that is downloaded from OpenStreetMap [OSM] and then converted into a suitable file to use with SUMO. One of the exits - the top right one from Figure 4a - is missing from the network file used for the simulations because there were no connecting edges on the outside of the building. Because the main exit - the bottom one on Figure 4a - is larger than other exits, it is considered to be two exits. In the following sections, firstly, the different testing strategies are explained, and then the benchmarking is done. The results are discussed and analysed to understand them.

4.2 Default configuration

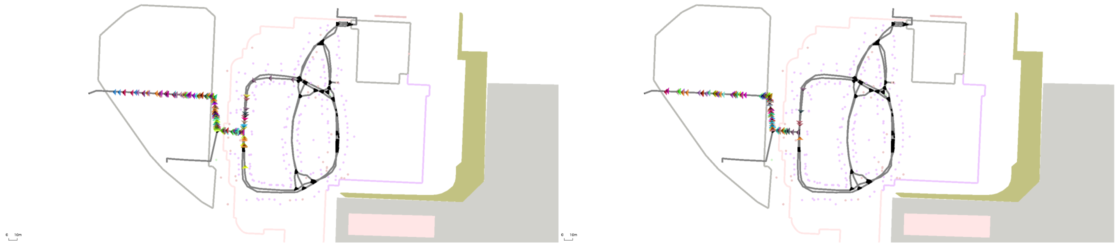
A test was conducted with 200 agents to see what the average duration of pedestrians leaving the building through one exit is for 20 runs. Screenshots from the running simulation can be seen in Figure 5. The agents change colour, which means that they change groups, and they move towards the exit.



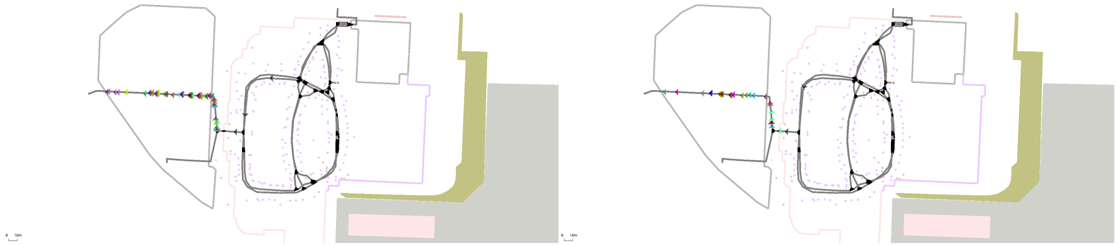
(a) Step 1 (b) Step 50



(c) Step 100 (d) Step 150



(e) Step 200 (f) Step 250



(g) Step 300 (h) Step 350

Figure 5. Screenshots of a running simulation with 200 agents

Two different speeds were used to see whether it affects the evacuation time. One speed was 1.4 m/s, which from literature is the average walking speed for people [BABAHK06][MBTCR$^+$07][LN99]. When in danger, people can also achieve higher speed, and because of this, the other walking speed was 2.5 m/s. The results for the default configuration can be seen in Figure 6. As expected, most of the runs with the smaller speed get a longer evacuation time. However, some of the runs with the higher walking speed of 2.5 m/s take longer. Because of this, not only one run is taken into action, but multiple runs. The mean duration with speed of 1.4 m/s over 20 runs is 704.90 seconds, and the standard deviation is 158.96 seconds. The mean duration with speed of 2.5 m/s over 20 runs is 687.85 seconds, and the standard deviation is 171.76 seconds. It seems that with higher walking speed, the evacuation time over runs fluctuates more than with 1.4 m/s walking speed.

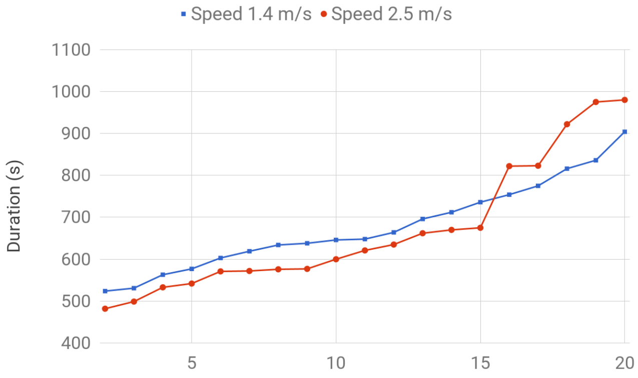


Figure 6. Duration time for one exit, 200 agents, and 20 runs

4.3 Different number of exits

The next tests will all have a different number of agents - 1, 50, 100, . . ., 350, 400. For each number of agents, ten runs were conducted and an average of those was calculated. The exits are specifically chosen edges in the network map. For these simulations, three exits were chosen. They can be seen in Figure 7. Exit 1 is in yellow, Exit 2 is in orange, and Exit 3 is in blue. The evacuation duration for the different number of exits will be shown and also the distribution of agents between exits.

4.3.1 Evaluating evacuation duration with one exit

On the one exit test, the pedestrians had only one exit through which they were able to escape the building. This exit was Exit 1. The speeds of 1.4 m/s and 2.5 m/s were both used. Figure 8 shows that with a smaller number of agents the speeds do not matter concerning the duration of the evacuation. With 300 agents the duration already seems to be much higher and the same is with 400 agents. This might be because of the crowd density inside the building and with it also confusion in which group to select. Because of only one exit, the distribution of agents between exits cannot be done.

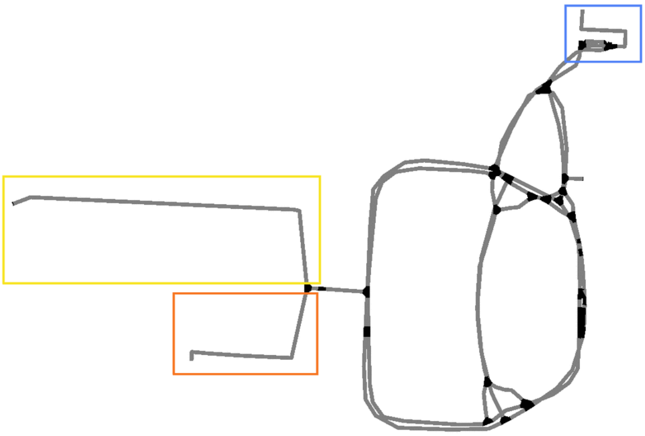


Figure 7. Network map with chosen exits - Exit 1 in yellow, Exit 2 in orange and Exit 3 in blue

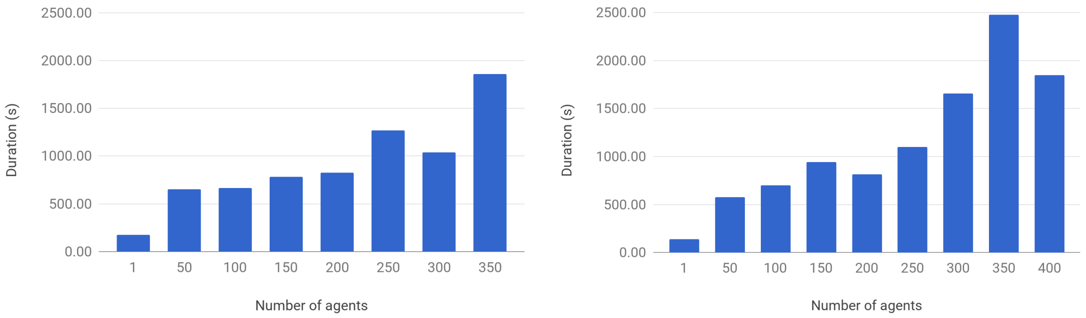
Figure 8. Duration time for one exit and different number of agents

4.3.2 Evaluating evacuation duration with two exits

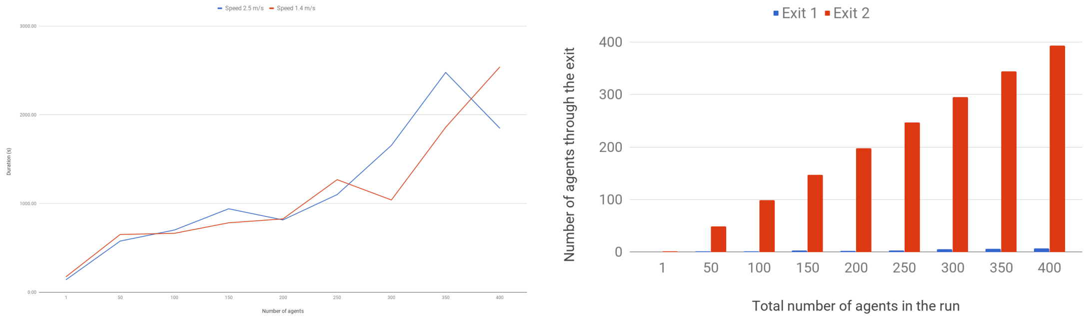
Two exit configuration was tested to see if it affects the duration time and how agents distribute between exits. Three choices of two exits were possible - Exit 1 and Exit 2, Exit 1 and Exit 3, and Exit 2 and Exit 3.

Exit 1 and Exit 2 The mean evacuation time with Exit 1 and Exit 2 for speeds 1.4 m/s and 2.5 m/s respectively are shown in Figure 9a and Figure 9b. A better duration comparison between the speeds can be seen in Figure 9c. As expected, the evacuation time increases with more agents. When agents have higher speed and the amount of agents is high, the evacuation times are also higher because the crowd density is high. Therefore, the agents cannot achieve their high speed. Furthermore, because Exit 1 and

Exit 2 are close together, there might be a bottleneck on the incoming edge that leads to both Exit 1 and Exit 2. Exit 1 and Exit 2 are quite close together as seen in Figure 7. Therefore, it makes it interesting to see how people choose their destination exit. With speed 1.4 m/s and 2.5 m/s, the distribution between exits stayed the same. The distribution with a different number of agents can be seen on Figure 9d for Exit 1 and Exit 2. A majority of the agents choose Exit 2 when there is a choice between Exit 1 and 2, even though they are close together.



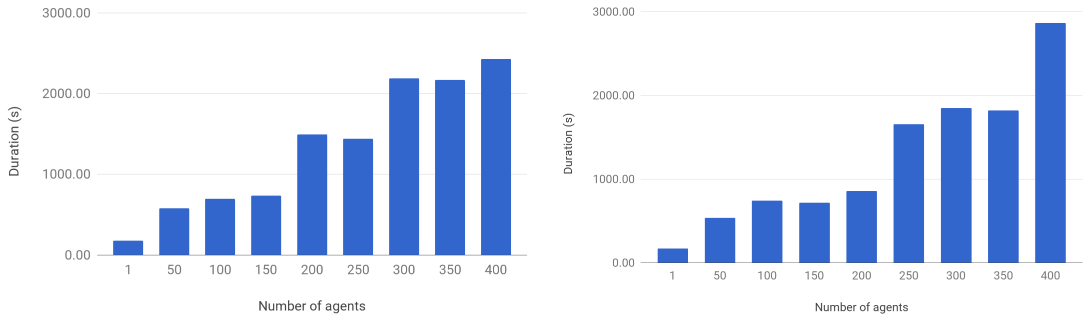
(a) Evacuation duration with speed 1.4 m/s (b) Evacuation duration with speed 2.5 m/s



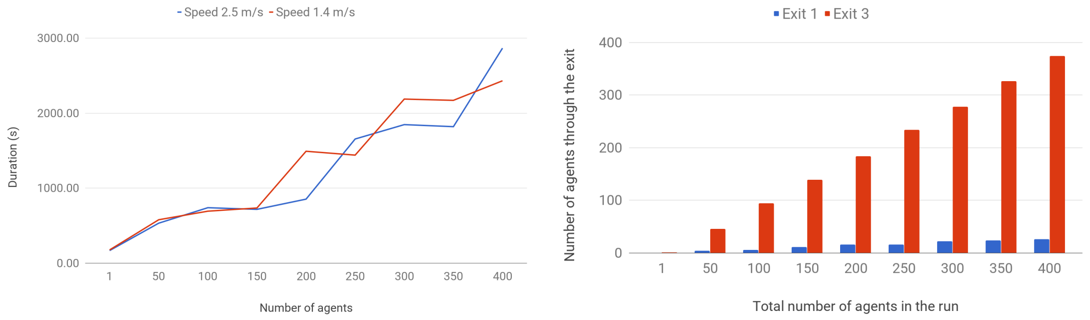
(c) Mean duration (d) Agent distribution

Figure 9. Simulation results with Exit 1 and Exit 2

Exit 1 and Exit 3 Figure 10a and Figure 10b show the barplots with Exit 1 and Exit 3 for speeds 1.4 m/s and 2.5 m/s separately. The comparison of mean evacuation time with Exit 1 and Exit 3 for speeds 1.4 m/s and 2.5 m/s is shown in Figure 10c. The more agents, the longer the evacuation time as with all the previous tests. Here, however, the evacuation times are shorter for the higher speed simulation, except with 400 agents. The exits are further apart from each other, and there is no bottleneck forming near the exits, which means the agents can reach their high speed when evacuating. The distribution between Exit 1 and Exit 3 is shown in Figure 10d. Because the distributions with speeds of 1.4 m/s and 2.5 m/s stayed the same, only one distribution is shown. As opposed to Exit 1 and Exit 2 distribution, more people choose Exit 1. The other exit, in this case Exit 3, is still more popular than Exit 1.



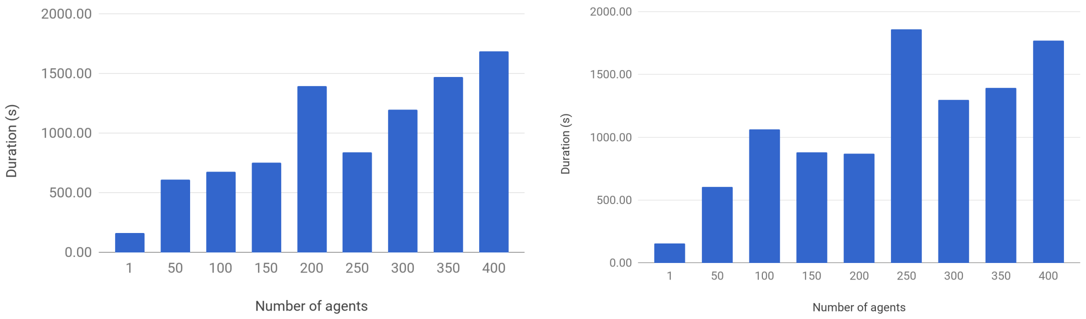
(a) Evacuation duration with speed 1.4 m/s (b) Evacuation duration with speed 2.5 m/s



(c) Mean duration (d) Agent distribution

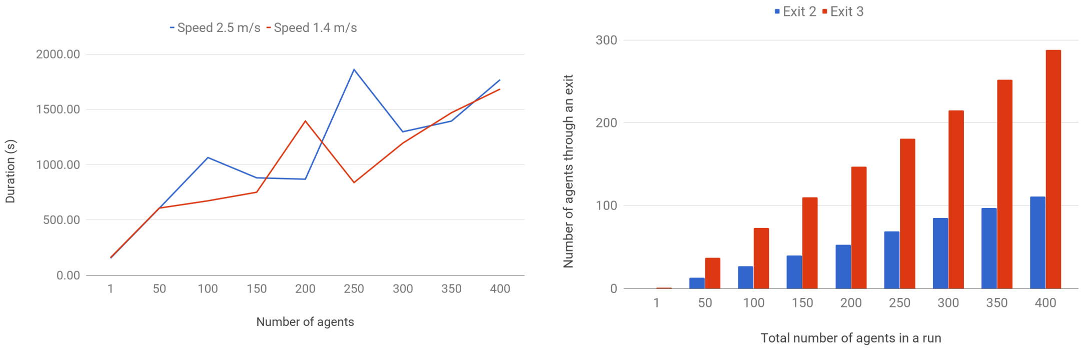
Figure 10. Simulation results with Exit 1 and Exit 3

Exit 2 and Exit 3 The evacuation duration for speed 1.4 m/s and 2.5 m/s with Exit 2 and Exit 3 are shown in Figure 11a and Figure 11b respectively. The comparison between the duration is in Figure 11c. The evacuation durations fluctuated more than with other combinations of two exits. The speed does not seem to affect the duration with Exit 2 and Exit 3 very much. The exits are the furthest apart from each other as opposed to the other two combinations of two exits - Exit 1 and Exit 2, and Exit 1 and Exit 3. As seen in Figure 11d most of the agents choose Exit 3 as their destination, but almost 1/3 of the agents seems to choose Exit 2. This is much higher than the other combinations of exits, where the less chosen exit only let through a minimal amount of pedestrians.



(a) Speed 1.4 m/s

(b) Speed 2.5 m/s



(c) Mean duration

(d) Agent distribution

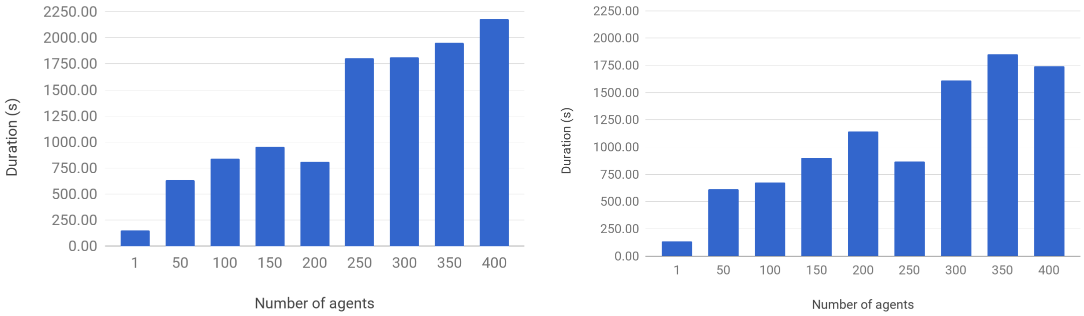
Figure 11. Simulation results with Exit 2 and Exit 3

4.3.3 Evaluating evacuation duration with three exits

The results for three exits are shown on Figure 12. The 1.4 m/s speed and 2.5 m/s results separately can be seen in Figure 12a and Figure 12b. It seems that with three exits, the higher walking speed results in a smaller evacuation time even with a higher number of agents. The same can be seen in Figure 12c. The distribution of agents between different exits is in Figure 12d. It appears that most of the agents prefer Exit 3, some prefer Exit 2 and a small number exit through Exit 1. Exit 1 and Exit 2 might have a bottleneck as seen with the two exit configuration. This might be the reason why Exit 3 is more preferred between agents.

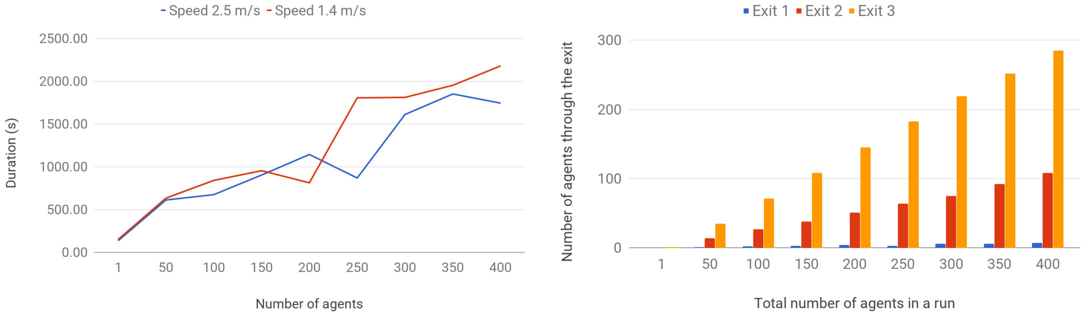
4.3.4 Comparing different number of exits

On Figures 13a and 13b the comparison of different number of exits with speed 1.4 m/s and 2.5 m/s respectively can be seen. With a smaller number of agents for the lower speed, the durations do not vary very much between each other. After 250 agents, three



(a) Speed 1.4 m/s

(b) Speed 2.5 m/s



(c) Mean duration

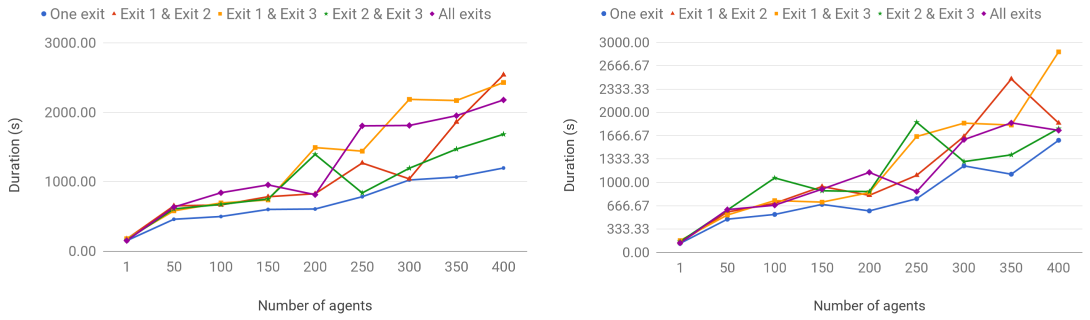
(d) Agent distribution

Figure 12. Simulation results with Exit 1, Exit 2 and Exit 3

different configurations are spaced evenly with the smallest evacuation time the one-exit configuration. After the one-exit configuration, the best time is achieved by Exits 2 and 3, which are the farthest apart, and the three-exit configuration. Configurations with Exits 1 and 2 and Exits 1 and 3 are quite similar. The reason might be a bottleneck between the exits.

When the agents are configured with the higher speed, and the number of agents is small, the number of exits does not seem to make a big difference in the evacuation duration. With the number of agents over 250, the evacuation times seem more similar than with the lower speed. For 400 agents, four of the five configurations finished at a very similar time. The only outlier is the configuration with Exits 1 and 3, which was much higher. The reason might be because of the randomness of the runs - some runs finish very quickly, while others take a long time.

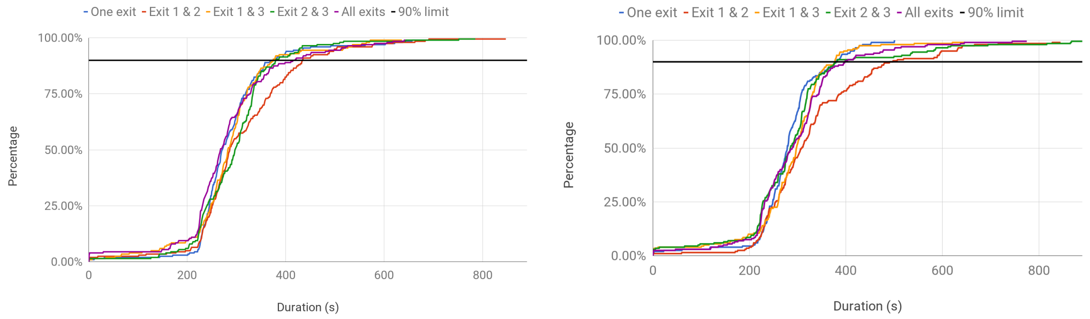
Another interesting thing to see is the correlation between the percentage of agents who have exited the building and the evacuation duration. This correlation is highlighted in Figure 14 for both 1.4 m/s and 2.5 m/s speeds using 200 agents. With both speeds and with the majority of configurations 90% of the agents are evacuated in approximately



(a) Speed 1.4 m/s

(b) Speed 2.5 m/s

Figure 13. Comparison of evacuation times between different number of exits



(a) Speed 1.4 m/s

(b) Speed 2.5 m/s

Figure 14. Percentage of people who have left the building vs. duration

400 seconds, which seems like a reasonable time to get out 180 agents out of 200. The percentage seems to increase more abruptly with the higher speed, which is to be expected.

4.4 Conclusion

In this section different simulation scenarios were described and then carried out. The results of the simulations were presented and analysed. It was found out that when the exits are close together, a bottleneck forms and the evacuation times are higher like with Exits 1 and 2 and Exits 1 and 3. When the exits are far apart, like Exits 2 and 3, the evacuation time is smaller. The evacuation time is also smaller with only one exit and all three exits being used. Moreover, the majority of people (90%) are evacuated in approximately 400 seconds for both 1.4 m/s and 2.5 m/s speeds.

5 Conclusion

5.1 Conclusion

The thesis aims to implement a microscopic model using autonomous agents, group modelling and social comparison theory to evaluate building egress design. With microscopic models, every agent has their own beliefs, ties with other people in the vicinity and it is possible to use ethnic properties for each agent like average weight which can determine the size of the agent. Social comparison theory introduces the notion that everybody compares others to themselves and partly bases their actions on this. For example, a tennis player might compare himself to another tennis player who is the same sex, around the same age and uses comparable equipment. In evacuations, strangers might use similar looking people to follow to exit the building. If agents follow each other, they form groups who influence each other. An article by F. Qiu and X. Hu [QH10a] describing a model based on these notions was the inspiration for this thesis. The model was partially implemented using the formulas for the article. Another article that gave the idea to assess the egress of the building was by Zhang et al. [ZZL09]. After implementing the model using Python, SUMO and TraCI, different simulations were run to test the difference between the different number of exits. One of the outcomes was that with 200 agents the number of exits does not play a role for 90% of the agents because they exit the building in approximately 400 seconds which is quite a reasonable time. Another outcome was that bottlenecks are a source of longer evacuation times. They can happen when the exits are closer together. The evacuation times were shorter with far apart exits, with only one exit and when all of the exits were used.

5.2 Future perspectives

For future work, more simulations should be tested and analysed. One aspect that was not tested is the width of the exits. The article by Zhang et al. [ZZL09] tested different exit widths and found that the width of the exit and the distance between exits affects the evacuation time. In this thesis, all of the exits were the same width. Another thing to note is that not all people walk with the same speed. Children and older people usually walk slower than adults and young people. Ethnic attributes could affect the walking speed as well. The model could benefit from implementing random speed assignment for each agent. The idea is that the user could give in the minimum and maximum speed limits as parameters and for each agent, a speed is generated randomly in that range. Lastly, many buildings have multiple floors, especially public places like shopping centres, hotels, and cinemas. Having multiple floors used in the simulation would make it more realistic and accurate. Furthermore, it would be interesting to develop the model further to use different floors and see how stairs, escalators and elevators influence the evacuation time.

References

[AHFH68] Harvey A. Hornstein, Elisha Fisch, and Michael Holmes. Influence of a model's feeling about behavior and his relevance as a comparison other on observer's helping behavior. 10:222–226, 11 1968.

[AIK10] Miho Asano, Takamasa Iryo, and Masao Kuwahara. Microscopic pedestrian simulation model combined with a tactical model for route choice behaviour. *Transportation Research Part C: Emerging Technologies*, 18(6):842 – 855, 2010. Special issue on Transportation Simulation Advances in Air Transportation Research.

[BABAHK06] Raymond Browning, Emily A Baker, Jessica A Herron, and Rodger Kram. Effects of obesity and sex on the energetic cost and preferred speed of walking. 100:390–8, 03 2006.

[CEFR14] Andrew Collins, Terra Elzie, Erika Frydenlund, and R. Michael Robinson. Do groups matter? an agent-based modeling approach to pedestrian egress. *Transportation Research Procedia*, 2:430 – 435, 2014. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands.

[Fes54] Leon Festinger. A theory of social comparison processes. *Human Relations*, 7(2):117–140, 1954.

[FK10] Natalie Fridman and Gal A. Kaminka. Modeling pedestrian crowd behavior based onacognitive model of social comparison theory. *Computational and Mathematical Organization Theory*, 16(4):348–372, Dec 2010.

[HM98] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. 51, 05 1998.

[KEBB12] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[LH15] Dewei Li and Baoming Han. Behavioral effect on pedestrian evacuation simulation using cellular automata. *Safety Science*, 80:41 – 55, 2015.

[LN99] Robert Levine and Ara Norenzayan. The pace of life in 31 countries. 30, 03 1999.

[MBTCR$^+$07] Betty Mohler, William B Thompson, Sarah Creem-Regehr, Herbert L Pick, and William Warren. Visual flow influences gait transition speed and preferred walking speed. 181:221–8, 09 2007.

[MPG$^+$10] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLOS ONE*, 5(4):1–7, 04 2010.

[OSM] Openstreemap web page. `https://www.openstreetmap.org/`. [Online; last accessed 21-05-2018].

[PHDL07] Xiaoshan Pan, Charles S. Han, Ken Dauber, and Kincho H. Law. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI & SOCIETY*, 22(2):113–132, Nov 2007.

[QH10a] F. Qiu and X. Hu. Modeling dynamic groups for agent-based pedestrian crowd simulations. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 461–464, Aug 2010.

[QH10b] Fasheng Qiu and Xiaolin Hu. Modeling group structures in pedestrian crowd simulation. *Simulation Modelling Practice and Theory*, 18(2):190 – 205, 2010.

[RGK00] George R. Goethals and William Klein. Interpreting and inventing social reality, 01 2000.

[RGMD87] George R. Goethals and John M. Darley. Social comparison theory: Self-evaluation and group life, 01 1987.

[SBK04] Marja-Liisa Siikonen, Kim Bärlund, and Risto Kontturi. Transportation design for building evacuation. 53, 01 2004.

[SHZHT09] Siamak Sarmady, Fazilah Haron, and Abdullah Zawawi Hj. Talib. Modeling groups of pedestrians in least effort crowd movements using cellular automata, 01 2009.

[sum] SUMO wiki. `http://sumo.dlr.de/wiki/Simulation_of_Urban_MObility_-_Wiki`. [Online; last accessed 15-05-2018].

[tra] TraCI documentation. `http://www.sumo.dlr.de/daily/pydoc/traci._simulation.html`. [Online; last accessed 14-05-2018].

[VMC13] Giuseppe Vizzari, Lorenza Manenti, and Luca Crociani. Adaptive pedestrian behaviour for the preservation of group cohesion. *Complex Adaptive Systems Modeling*, 1(1):7, Mar 2013.

[WHWL08] A. Wegener, H. Hellbruck, C. Wewetzer, and A. Lubke. Vanet simulation environment with feedback loop and its application to traffic light assistance. In *2008 IEEE Globecom Workshops*, pages 1–7, Nov 2008.

[WPR$^+$08] Axel Wegener, MichałPiórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. Traci: An interface for coupling road traffic and network simulators. In *Proceedings of the 11th Communications and Networking Simulation Symposium*, CNS '08, pages 155–163, New York, NY, USA, 2008. ACM.

[XD10] Song Xu and Henry Duh. A simulation of bonding effects and their impacts on pedestrian dynamics. 11:153 – 161, 04 2010.

[yle] Web page of Ülemiste center. `http://www.ulemiste.ee/`. [Online; last accessed 21-05-2018].

[ZCC$^+$10] Suiping Zhou, Dan Chen, Wentong Cai, Linbo Luo, Malcolm Yoke Hean Low, Feng Tian, Victor Su-Han Tay, Darren Wee Sze Ong, and Benjamin D. Hamilton. Crowd modeling and simulation technologies. *ACM Trans. Model. Comput. Simul.*, 20(4):20:1–20:35, November 2010.

[ZZL09] Q. Zhang, G. Zhao, and J. Liu. Performance-based design for large crowd venue control using a multi-agent model. *Tsinghua Science and Technology*, 14(3):352–359, June 2009.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Annika Laumets-Tättar**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
    1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
    1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

    of my thesis

    Modelling and Simulating Indoor Pedestrian Movement Behaviour and Displacement

    supervised by Dr. Amnir Hadachi

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 21.05.2018