Comparative Analysis between Micro and Macro Agent Evacuation Using PedSim Simulator

Hrishikesh Narayanankutty

Matriculation Number : 255670

A Dissertation

Presented to the Faculty

of University of L'Aquila

in Candidacy for the Degree

of Master of Science

Adviser: Professor Henry Muccini

Co-Supervisor: Mahyar Moghaddam

March 2019

Abstract

Disaster management pertains to coping with disaster, relief and evacuation in the event of a catastrophe. It is of paramount importance as it pertains to the protection of lives and property during the time of calamity. To mitigate the damage of such incidents, simulations are performed in order to formulate an efficient and optimized exit strategy for the people who are struck by such unfortunate incidents and to evacuate them to a safe zone as quickly as possible. The simulation of an evacuation of the participating agents broadly falls under three categories - Macro Agent Simulation, Micro Agent Simulation and/or a combination of Macro-Micro Agent Simulation (grouping according to age, gender, social ties such as family, friendships etc.). Although extensive work has been carried out to simulate disaster scenarios comprising of a massive number of agents with an aggregate set of characteristics (Macro Agent Simulation), very little work has been done so far to simulate realistic social behaviour of agents during a disaster scenario, especially pertaining to micro agent simulations. Realistic human and social behaviour characteristics are possible only through micro agent-based simulations as complex psychological and sociological paradigms can be mapped and hence dynamic real-time strategy decisions can be better understood, especially during a crisis. Through this thesis, I aim to present the simulation and comparison of various micro and macro agent scenarios.

In order to run the aforementioned simulations, an open source microscopic pedestrian simulator - PedSim is used. This tool not only simulates the various complex scenarios. it also provides visual feedback in real time. PedSim is a crowd simulation library capable of analysis of real time pedestrian flow rate. This agent-based model (ABM) tool is a class of computational models for simulating the actions and interactions of autonomous agents (either individual or collective entities such as organizations or groups) with a view to assessing their effects on the system as a whole. It combines elements of game theory, complex systems, emergency, computational

sociology, multi-agent systems, and evolutionary programming. Although there are many proposed agent modelling simulators that are available, many if not most are with commercial license and do not support real time data flow. This tool is also customized and further extended in order to take an optimized routing algorithm for agents. The PedSim simulator is modelled for both indoor and outdoor areas (parking lots, forests etc.). The simulation tool is able to take sensory based data and apply them to the modelling agents/nodes to simulate real/design time analysis. The main advantage of this library is the architecture that enables visibility of users live using tcp/stream-based output through batch processing. The implementation is pure in C++ with minimal external dependencies like Qt Framework. The output of PedSim can be translated using a graphics engine to provide visually appealing realistic render of a walking person. The code is modular, scalable and open source available under GPL license.

The main goal of this thesis is to exhibit an objective performance analysis and a comparison between an existing macro agent simulation algorithm against an optimized algorithm coupled with realistic constraints, testing them against a massively populated macro and micro agent model for real-time dynamic evacuation.

Acknowledgements

To my parents and my dear brother,

I am grateful for your undying love and support through the challenges I face. I am the person I am today because of you. I shall always cherish your wisdom and presence in my life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Due to increasing topological changes in urban environment, the human civilization has been subjected to increasing risk of disasters due to natural or artificial, and/or a combination of the two causes, since the better part of the modern era. Hence it has become an ever increasing need to design infrastructure to handle such disasters in the event it may occur. But even more so, the safe evacuation of people and personnel with the premises of the affected infrastructure takes precedence when dealing the necessary mitigation and disaster risk management. The evacuation time of people from a scene of disaster is extremely crucial in the case of an emergency in disaster situation. In order to reduce the time taken for evacuation, better and more robust exit strategy evacuation algorithms are developed which are used to model participating agents for their exit patterns and exit strategies and clock them based on performance, efficiency and practicality. In order to evaluate such parameters, evacuation simulation models have been developed and constantly tested over the years to investigate the emergency egress capabilities of the built environment for a variety of reasons including: difficulty in conducting real evacuation tests (drills); aiding building design and confirming conformity to building regulations; and even determining optimal evacuation routes for the building occupiers [4]. Computational models were

first introduced in the early 1980s since the inception of computers. During the 90s, academic researchers aimed to improve these the capabilities of these models in order to optimize and improve the pathfinding performance of the evacuees and their corresponding movements [15]. Over the course of many years, these algorithms have become the industry standard to perform exit evacuation time and performance analysis. The purpose of this thesis is to study the state of the art algorithm and to draw comparisons between micro model agent simulation with respect to its counterpart, the macro model agent simulation.

1.1   Motivation

Realistic simulations involve complex relationships between an individual and the surroundings. Three types of interactions are possible whilst the individual performs complex decision making during an evacuation scenario. Through this thesis I aim to optimize these three types of interactions through various constraints and real time elements. The following encounters may be classified as below [4]:

- People-people interactions - interactions between participating evacuees.

- People-structure interactions- interactions within the enclosed topology.

- People- environment interactions- interactions with quarantined atmosphere (fire, smoke etc.) and possible debris.

During disaster scenarios standard evacuation pathways are often rigid and cannot autonomously provide modification for an exit strategy as the disaster ensues. The evacuees often find themselves in situations that force them to rely on general guidelines about how to react in emergency evacuation [3]. While such dynamic hazards cannot all be dealt with, using the traditional approach, Lujak, M., Billhardt H., Dunkel, J. et al [19] attempts to help the evacuees adapt to the changing topography

of the environment due to hazard dynamics by updating a real time monitoring system using an IoT architecture built in place within the premises of the said topology of the building. The above mentioned work uses a combination of IoT devices to sense and identify and provide real-time monitoring for the participating agents. Our work also obtains data from real time sources and then it is simulated with both macro and micro based agent models and then used to analyse and draw conclusions from the resulting data.

1.2 Agent Based Model Simulation

The present thesis is a design of a computational software using Agent Based Model (ABMS) to help speedy evacuation in emergency situations. Our software architecture help optimise the navigational flow rate in cases of real-time disaster management. The scenario is that of many victims are struck up in a very large building in the event of an occurrence of a disaster like fire, earthquake, poisonous chemical gas leakage, imminent bomb attack, potential imminent building collapse (similar to that of twin towers of 9/11). As a side benefit the experience acquired gives valuable information in the case of architecture design at the time of building construction itself. By conveying a suggestive path for each individual in the building at the time of disaster, the panic, erratic and groping movements are reduced to minimum helping to achieve a streamlined flow. Bottlenecks in the flow are avoided by redirecting people to alternate paths. The benefit of the overall perspective of the scenario and informed management is instantaneously conveyed to each and every individual in real time. The floor capacities and width of the doors and passageways form part of the constraints in the modelling. This is scalable and generic version.

1.3 PedSim Simulator

PedSim is a pedestrian simulator tool designed as a front end to facilitate disaster management scenarios. Disaster management plans are multi-layered and topology specific. Tools catering to disaster management require the users to input topographic information and also provide crowd/agent location information. This aggregate of data is then processed initially to create a layout of the premises. The layout defines several rules and boundary conditions imposing restrictions for the crowd to navigate. Such information can be used to instantly model very specific scenarios such as evacuation of specific floors of a building. Since, PedSim enables input of design specific topology, protocols can be established quickly for speedy evacuation. The second part includes behaviour analysis of crowd during different disaster patterns. The social behaviour of victims in disaster afflicted scenarios have been of keen interest to researchers over the recent years. Disaster affected regions often portray victims who exhibit severe trauma or stress leading to emotional/psychological shut down or present themselves making erratic decisions etc. This tool enables a platform for crowd flow modelling. Erratic movement patterns can be modelled, simulated and visualized to study the impact and speed of evacuation. Such processed information can be used to provide instructions or navigation guidelines for optimal evacuation. PedSim also allows for real time monitoring of crowds providing instant feedback on the visual front end. This enables redirection of crowds for balancing congestion while developing exit strategies. Each exit can also be associated with restrictive exit capacity and corresponding flow rate ensuring crowd balancing. The architectural design of the tool itself takes into account various factors such as streaming and batch processing of time critical data within permissible thresholds.

1.4　Objectives and Approach

My work pertains to application of PedSim on our university building to implement and analyze the impacts of crowd evacuation during a disaster. The university building taken into consideration for our present study is level 3 of building (Coppito 0). The area under consideration is divided into 18 blocks. Each block consists of a set number of cells. Each cell represents a cubicle or a room. There are 26 cubicles/rooms in all. There is also one conference hall and two generic multipurpose areas used for allied purposes. In addition, there are several hallways and pathways for navigation. This area has four key exits distributed across the different boundary walls of the premises. At any given time, each cell hosts atmost twenty agents, reaching a maximum during peak hours of 10am to 2 pm Italy time. The premises often on average hosts about 100 people. We architected pedestrian flow simulations during a generic disaster which can include earthquake, fire and chemical leakage. The data is used to generate crowd clusters during events other than disasters such as workshops, cafeteria gatherings, conference gatherings etc. The figure 1.1 shows a typical population cluster during a conference gathering.

This thesis presents a case study where the Coppito 0 building is simulated during a disaster scenario using grid geometry based navigation algorithm as presented in the An IoT Software Architecture for an Evacuable Building Architecture by H.Muccini et.al [20].
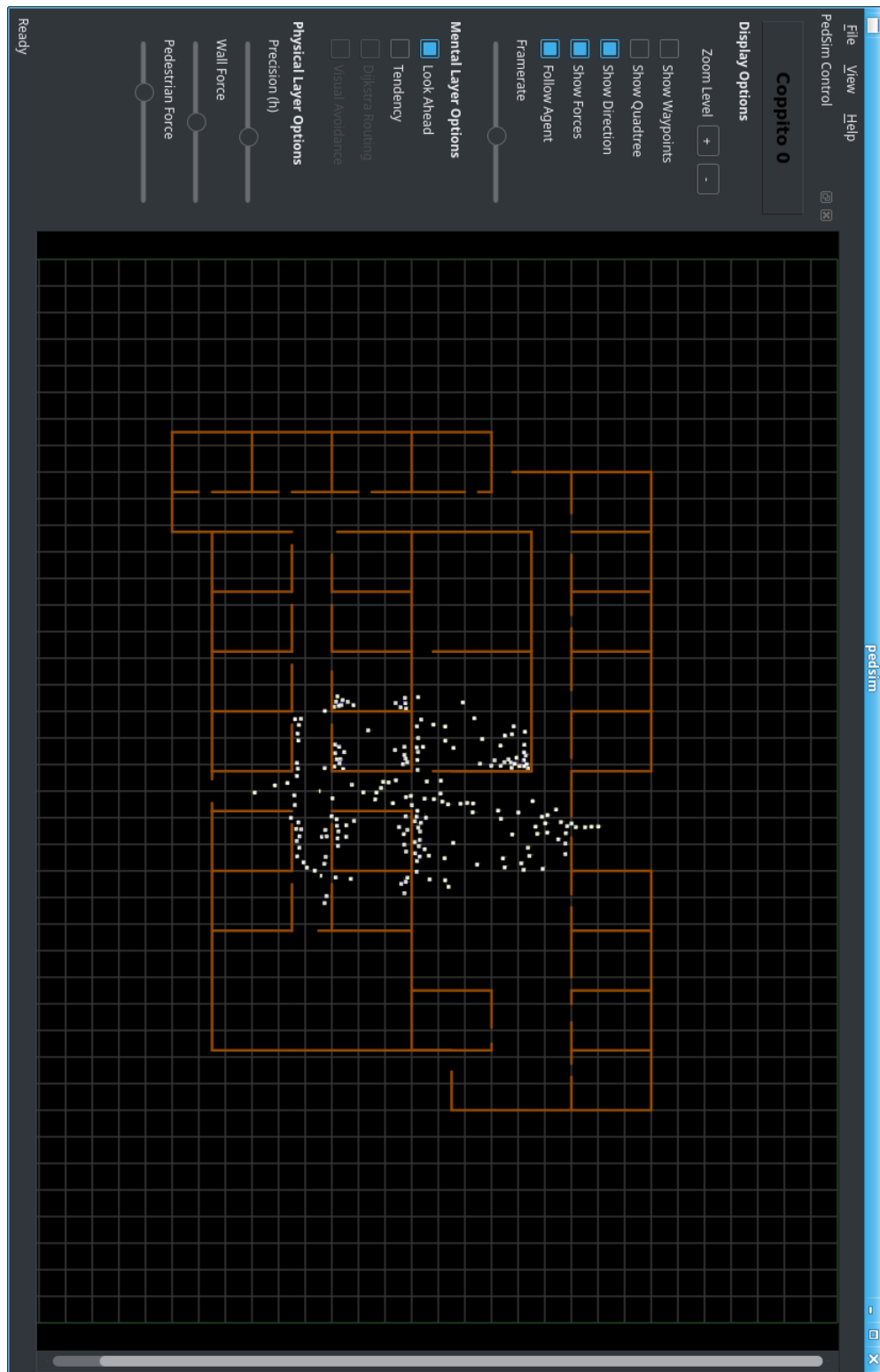
Figure 1.1: Agent Cluster Formation in Coppito 0 Building

The work also involves in expanding the capabilities of the existing PedSim simulation tool to include the new topology and the also to incorporate the algorithm and also further extended developed from the aforementioned paper and perform a real time analysis between micro and macro agents with the following criteria:

- specifying the cells by social distances

- respecting the area capacity and doors capacities constraints

- varying movement speed according to various groups

- simulating social attachment among some agents

As it is evident from the above constraints, the goal of this algorithm and this simulation is to make exit strategies as realistic as possible. By grouping certain agents together for instance, we can achieve realistic movement of participating agents, as in reality, people usually move in groups. families, friends etc. Furthermore, congestion is a big part of the evacuation scenario, as it is congestion that determines how fast or efficient the evacuation of agents occur and also how it would take to move groups of people. With groups of people clumped together, the next most natural constraint would be to assign varying speeds as not all agents and not all groups of agents travel with the same speed across the topology of the building. There is also one other constraint which determines the cell capacity, which determines how spaced out and occupied agents are in a room, a hall or a passageway. This also help determine how many can get through a single doorway in order to transition from one passageway to another. This of course leads to previously mentioned issue of congestion.

All these constraints are hence taken into account as the simulation is performed in order to clock and test for the performance time and efficiency of the algorithm used for micro and macro model simulation.

1.5 Thesis Outline

The structure of the thesis is formulated as follows: chapter 2 provides a detailed literature and background work pertaining to the area of agent based simulation models and exit strategies. Chapter 3 focuses on the implementation details of the algorithm and specific scenario details that are incorporated in order to extend the algorithm in order to adapt it to a more realistic setting and also technical details regarding the PedSim simulation environment. Chapter 4 presents the results of the comparisons of the optimized(realistic) algorithm between the macro and micro agent simulation setting and to chart out the various performance metrics that are obtained. Chapter 5 attempts to draw conclusions based on the obtained performance metrics and also present the future work related to the aforementioned. This thesis also provides a technical appendix for reference for the simulation of various micro and macro agent scenarios within the environment of PedSim.

Chapter 2

Related Work

In this section the background literature and the work related to agent based simulation is dicussed.

2.1 Pre-Modern Disaster Preparedness, Management and Relief Techniques

Although disasters have existed throughout the history of the universe for all living organisms, human beings are perhaps the only one to be affected on a massive scale as a collective group as we are vulnerable and weak compared to other animals that live and co-exist with us. Hence human beings have taken very many steps over the years fortifying and defending ourselves as best as we can through the various buildings and escape routes we design. The early inhabitants of mankind were not idle and did not become easy victims. There is historical evidence that the early man took various measures in order to cope with, reduce and mitigate risks. The mere fact that they chose to inhabit caves are a testament to this theory [9].

The ancient man also used other natural techniques for disaster preparedness. One of the prominent ones include the observation of animals since they are very perceptive

to natural disasters, animal behavior can and has been used for the better part of the millenia to predict the onset of a natural disaster such as floods, earthquakes etc. One of the earliest works to be done in this field involves in the observation of the behavior of fish, specifically the catfish, as they were found to exhibit a definitive behavior in advance of the occurence of an earthquake [16]. Other experiments and observations pertaining to such natural responses include the obeservation of ground electric field effects on behavior of Albino rats, Mongolian gerbils (sand rats), hair-footed Djungarian hamsters, guinea pigs, and red sparrows [17]. A summary of such animal behavior and work is presented by Neeti Bhargava et. al. [5].

To combat various natural disasters that affect us, mankind has tried for over a millenia to improve and adapt buildings, and entire cities to cope during the time of a catatrosphe. There are two broad categories to the mitigation of these disasters that have been long employed [10]:

- structural based mitigation

- non-structural based mitigation

Since my work pertains to indoor simulation and evacuation management, the discussion for non-structural mitigation forms here is considered unnecessary and beyond the scope of this thesis. Structural mitigation involves the presence of a building or a man-made mechanical or technological adaptations performed to reduce hazards and mitigate risks. The following sub-section expands more on the more modern techniques that have been developed over the course of the modern era and their implications. Although better and state of the art systems have been developed over the years, these systems have complications of their own. Failure of the perfect working of an evacuation system leads to the possible death of all the agents who are in need of evacuation.

2.2 Routing Algorithms and IoT Based Evacuation Management

Routing algorithms and maze solving algorithms have especially been applied to the structural domains for the real time analysis of shortest paths and obstacle avoidance by both machine and human agents since the time the earliest shortest paths algorithms were developed. The development of the shortest path algorithm by Dijkstra in 1956, especially has seen a plethora of applications in various fields. As one can guess most of its applications have been catered to graph based problems and obstacle avoidance. Shortest paths exit strategy formation approach is a good way to analyse a safe egress as it provides obstacle avoidance as well. How can it successfully be applied to a real disaster scenario? Before the advancements were made to IoT based infrastructure, maze solving algorithms were used to successfully analyze a safe egress.

IoT and computation systems have brough about a tremendous potential to perceive the environment and then form the necessary strategies that are required to safely escort the human agents to a safe exit point during the time of disaster. Modern evacuation systems have sensors in place to allow for better perception of the world during a disaster scenario. For instance the work by Kobes et al. determined that during fire related disaster scenarios, 56.3% of the participating agents were able to determine the exit points based on the exit signs when there was no smoke present whilst 81.8% was able to identify exit points only based on the exit signs when their vision was imparied due to smoke [18]. Based on such data for instance, it greatly enhances our need to analyse the fastest and shortest egress path so that all human agents in the vicinity can get to the exit points. The shortest path algorithm - Dijkstra's algorithm is currently used as a great tool to provide the shortest exit points as it is demonstrated in the work by Jehyun Cho [7]. His research pertains to the

dynamic analysis of a shortest path algorithm based on information obtained from the sensors and the smart infrastructure.

In order to address the issue of agent(s) and infrastructure mapping during a disaster event, state of the art sensors are placed all through the infrastructure in order to perceive the environment and the surrounding structures in order to analyze a safe egress. Motion detectors for instance are used to understand how many people are still trapped inside the building during a catastrophe. One example of using IoT technologies is proposed by Prasad Annadata et al. using multiple WIFI channels and the aforementioned motion detectors [2]. Based on the statistics gathered heuristic solutions are proposed in order to detect the number of personnel inside the infrastructure as well as perceive the environment.

Although the proposed methods and techniques are helpful in determining the shortest path to an exit point, they rarely can be used in a realistic scenario due to its simplistic nature. A real world scenario is more complex due the social dynamics of people. Such complexities are better understood while simulating agents in real and static simulations and then performing an analysis based on the observed stats. Such is what I hope to achieve through this thesis.

Existing literature on surveying personnel and the surrounding infrastructure for assistance based support for immediate first responders include the work the done by Palmieri et al., who proposes a hybrid cloud architecture to manage and store the necessary required resources to command and control activities during emergencies [21]. However the work done by H. Muccini et al. [20] aims to improve this work by adapting geolocation of first responders to track people during a disaster for evacuation. Furthermore W.Choi et al. [8] proposes to model building evacuation by dynamic flow maximization and by considering variable capacities on some arcs as a function of flows in incident arcs. For the purposes of modeling our topology we have used a similar arc based geometry for determining the topological capacity of a

particular cell into which the entire infrastructure is divided into. Chen et al. [6] in addition to the above, proposes a flow control algorithm that calculates evacuation paths depending on building plan and total number of evacuees. Computation in this case aims at minimizing total evacuation time and assigning an optimal number of evacuees to each evacuation path. However [5] provides a robust solution by architecting an IoT system to monitor and update dynamically the topology of the environment.

Another important aspect that remains unaddressed is that in real disaster scenarios, over crowding and congestion of certain pathways and exit points are bound to occur frequently. The issue of congestion is serious and perilous and is arguably pointed out by a case study performed by John El Khoury [12] and analyses the dangers due to high traffic intensity during a disaster event. Although the analysis is performed on a wide scale city range, the implications of congestion and other such constraints are just a applicant to a much smaller scale infrastructure. To dynamically reallocate personnel to routes that are not just shortest paths but optimal paths is what we try to achieve through these simulations and added constraints. The work done by Antoine Desmet et al. [11] addresses the congestion issue by a "self adaptation" algorithm much inspired by the computer network routing algorithm - The Cognitive Packet routing algorithm by E. Gelenbe [13]. A robust evacuation with optimized routing solution proposed by H.Muccini et al. [5] are as follows:

- Optimal solutions that can be continuously updated, so evacuation guidelines can be adjusted according to visitors position that evolve over time.

- Paths that become suddenly unfeasible can automatically be discarded by the system.

- The model can be incorporated into a mobile app supporting emergency units to evacuate closed or open spaces.

The aforementioned algorithm is still incomplete as dynamic and realistic constraints are yet to be added. The constraints are 4 fold as mentioned in the introduction section. Over the course of chapter 3, the details of the algorithm is explained and how these constraints provide an even more optimized route since it considers congestion, grouping, varying age and speeds in addition.

2.3   Simulation Tools and Constraints

In order to accomodate for accurate results and model precise human behavior, agent based modeling and simulations have become increasingly popular as computational expenses became cheaper and ever more accessible. In addition to developing state of the art routing algorithms and other obstacle avoidance mechanisms, it was also absolutely important to understand human behavior and the complex decisions that agents make during a catastrophic event. To understand the implications of human behavior and complex real-time decision making and other realistic constraints, this thesis presents various scenarios that are depicted in a small to a large scale microscopic agent based fashion in hopes to draw out some interesting conclusions based on the obtained results. One such example of the kind of work that we are hoping to perform is the impairment of certain agents based on the circumstances that are met with in real time during the disaster event. The work presented by Selain Kasereka et al. [14] offers a unique insight into modeling agents for a fire disaster and simulating the case for some interesting results. It should be mentioned that the case presented in the current thesis however is modeled for a generic scenario. In the work presented, the smoke and fire can affect the agent, thus reducing their potency and their ability to escape from danger. This damage to potency can be essentially translated to a reduced movement speed and a severe impairment to their cognitive ability to perform real time decisions.

Another really interesting thing to point out is about psychology and human behavior during ground zero scenarios. Human panic and confusion is generally a chaotic element that must certainly be infused into such simulations as they provide further insight into human behavior and particularly decisions that lead to increase or the decrease in congestion. Based on certain decisions and paths, it may cause discomfort or physical pain to some and many agents within the promixity of such an incident. Ashutosh Trivedi et al. [22] provides a detailed analysis and where different strategies are evaluated and the corresponding evacuation time and physical discomfort caused to the agents are observed. He justifies the effect of social forces based on cohesion which, in my thesis, is also considered whilst perfoming simulation for microscopic agents. Our work revolves very closely the work done by A. Trivedi et al. as it also deals with social forces, panic and other realistic constraints. Through this thesis, experimental analysis for these sophisticated constraints are performed for both macro and micro agents in hopes for better understanding and formulating a safe egress for the agents.

Simulation tools have been used since the inception of computers and computation technology since the early 1970s. Modern simulations for evacuation and relief management includes the simulations of pedastrians and agents within the domains of the inscribed topology, with start and end points. These simulations are then run against various parameters and scenarios thus obtaining various results. Through this thesis the various experiments are simulated within the environment of PedSim, a massive microscopic real time simulator. The in depth analysis of large number agents based simulators are provided by the extensive and comprehensive survey presented in "Agent Based Modelling and Simulation tools: A review of the state-of-art software" by Sameera Abar et al. [1]. The survey is based on the following criteria which is presented as follows:

- License/Availability

- Source Code in which the simulator is written in

- Type of Agent based on its interaction and behavior

- Coding language or application programming interface(API) for Model Development; Integrated Development Environment(IDE) used

- Compiler/Operating System/Implementation Platform

- Model Development Effort

- Modelling Strength/Simulation Models' Scalability Level

- ABMS scope or Application Domain

From the extensive survey provided, it was easily inferred that PedSim, pedastrian microscopic simulator was the best suited for testing the various scenarios as it could support both micro and macro based agent modelling scenarios and grouping. It must also be mentioned that the tool is entirely written in C++, extremely light-weight and highly scalable. It also works well on development platforms and can be compliled easily on all operating systems. Its simple and easy to use, as complex simulations can be run for massive number of agents and the results can be visualized in real time. For all these reasons I have decided to use PedSim exclusively as my test bed tool whilst modelling such scenarios.

In the following section the scenario and the implementation details of my work will be discussed.

Chapter 3

PedSim Environment and Setup

This section briefly explains the underlying PedSim library modules and the basic algorithm that is presented in [20], based on which more constraints were added for realism and optimization; used for the simulation of pedastrian agents. The following subsections will describe the environment and the topology of the building, the algorithm that is used for the movement of participating human agents and the necessary changes that were made to make the algorithm more realistic and the technical implementation details of the PedSim simulation environment.

3.1   Scenario Description

The work involves simulation of level 3 of Coppito 0 building during a disaster scenario such as a fire. The building topology consists of 26 rooms and cubicles connected by several hallways and pathways for exit and is depicted in 3.1. There are four exits in total. Two in the middle top and bottom denoted by a the red font as **E** and **E"**. There are two more on the left and right which is depicted by **E'** and **E"'** also in red. The entire premises is divided into cells which form the basic unit of analysis. Each room is made up of one or more cells. For instance the room at the top left corner is made up of cells 96 and 97. Dark blue lines indicate wall boundaries. Each cell for

our scenario has a dimension of 4.5x3 in meters. The doorways provide a separation of 1 meter each. Roughly at the center of the building, two conference rooms and a hallway provides a substantial gathering area.

The work here caters to simulating the flow of crowd during the event of a fire across the four exits. Restrictions such as exit clogging due to intense smoke/fire can be imposed redirecting most of the traffic across specific other exits. Also crowds from crowded exits can be siphoned off by redirecting them to an alternative less crowded exit. The data flow density of population which used for simulation in Pedsim is measured using an IoT framework of smartphones, cameras, RFIDs and motion sensors that enable localization of victims in the room. The tool was thus used as a framework to simulate evacuation algorithms in Coppito 0.

The given figure 3.1 is the graphical description of Coppito 0 building and is referred from the outline and scenario as stated in [20].
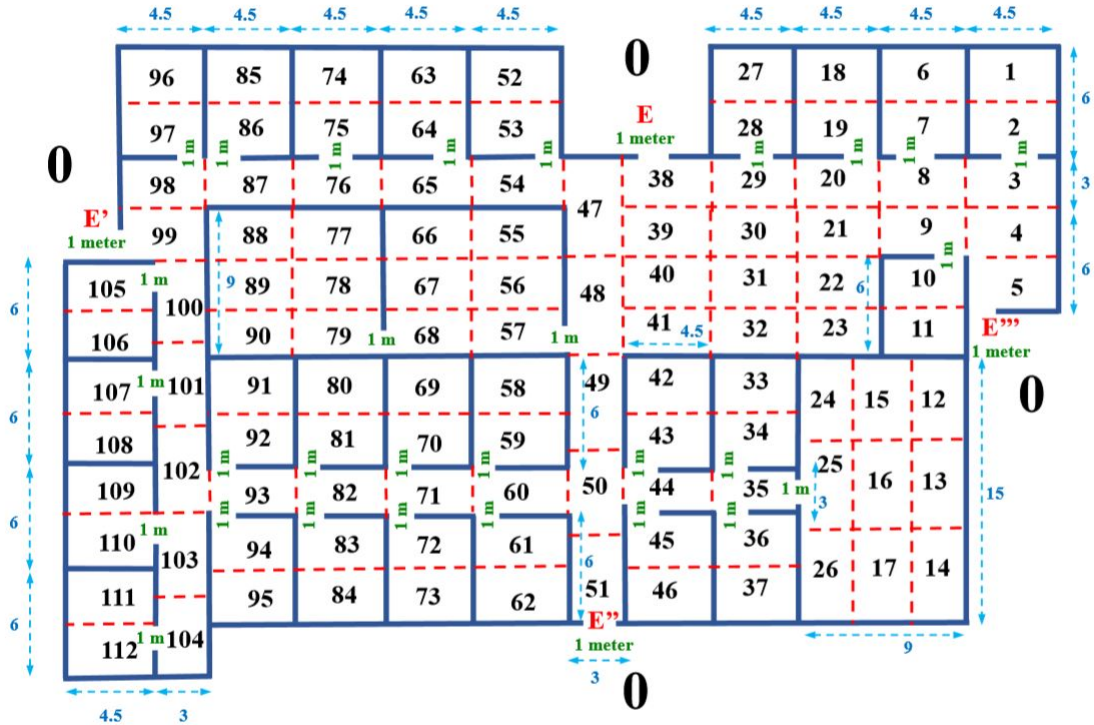

Figure 3.1: Coppito 0 Topological Layout

A detailed description of adaptation of the tool to suit the given scenario is described in the next section.

3.2 PedSim Library Modules

The Pedism library allows for the use of pedestrian dynamics into our own software. The libpedsim simulation rendering engine can be extended, modified and modeled to suit specific behavior patterns and scenarios. The above disaster considered is fire and through this thesis, we aim to demonstrate the capabilities of this microscopic simulator, modeling crowds during an emergency evacuation using the above mentioned building topology.

The implementation of the various scenarios(which will be explained briefly in the following section) using Pedsim is developed on Ubuntu 18.04 using libpedsim version 2.4.2. The library itself contains many subsections and modules which handle specific tasks such as agent movement, topology description etc. The various modules of the pedsim library is described below.

The libpedsim tool can be broadly classified into 2 sections - the simulation of pedestrians and graphically rendering the simulation process onto a QT based graphical window to depict the flow of agents in real time process.

The following modules are part of rendering output to a graphical window:

1. agent.h

2. agent.cpp

3. cell.h

4. cell.cpp

5. config.h

6. config.cpp

7. control.h

8. control.cpp

9. control.ui

10. grid.h

11. grid.cpp

12. mainwindow.h

13. mainwindow.cpp

14. moc_loadscene.cpp

15. moc_control.cpp

16. moc_mainwindow.cpp

17. moc_scene.cpp

18. moc_predefs.cpp

19. qrc_application.cpp

20. scene.h

21. scene.cpp

22. style.h

23. style.cpp

24. tree.h

25. tree.cpp

26. ui_control.h

As the above modules are used to render the algorithm to a graphical output, most of the modules as mentioned above need very little to no modification whatsoever. It is also worth noting that the modules agent.h and agent.cpp contains the actual definitions for the behavior of pedestrian agents. These behavior functions are implemented by the author of the tool as a generic inter-agent based interaction and is developed keeping in mind to add further and more complex behavioral functionalities. These functions can be extended in the ped_agent.h and ped_agent.cpp respectively. The following modules below represent the core modules that we extend and modify to suit the scenario at hand:

1. coppito.h

2. coppito.cpp

3. loadscene.h

4. loadscene.cpp

5. ped_agent.h

6. ped_agent.cpp

7. main.cpp

The first two modules - coppito.h and coppito.cpp is an external module that is incorporated into the source of the libpedsim library for the specific purposes of modeling the building Coppito 0, as the name suggests. The detailed description of coppito.h and coppito.cpp will be listed after the brief description of the other mentioned modules.

loadscene module is used to generate and extract the specific topology of the building. The exact dimensions of the building is stored on an .xml file which is then

fed into the loadscene module to incorporate into the library source for graphical representation. This .xml file has specific tags which can be used to not only describe the dimensions of the building but also to specify the number of agents and the path of their trajectory - mentioned as waypoint within the .xml file. This information is then retrieved by the loadscene module, extract information from the various mentioned tags, and generating the mentioned number of total agens on the graphical output, adding the constrained waypoints to the agents, and most importantly, to gather the information required to generate a graph that depicts the nature of the building in description.

The ped_agent module describes the behavior and movement of the agent that is to be rendered to the graphical window. This module consists of behavior functionalities that typically incorporates social forces, obstacle forces, look ahead force, desired force and my force. The definition of these forces (generic navigation constraints and behaviors) are explained below:

1. my force:

   - myForce() is a method that returns an "empty" force (all components set to 0). This method can be overridden in order to define own forces. This can thus be used to model more complex human navigation/decision making patterns.

2. lookahead force:

   - This calculates the mental layer force of the strategy "look ahead". It is implemented here in the physical layer because of performance reasons.

3. obstacle force:

   - This calculates the force between this agent and the nearest obstacle in this scene.

- It iterates over all obstacles.

- Hence the complexity of this module is equal to $O(N)$.

4. social force:

- This module calculates the social force between this agent and all the other agents belonging to the same scene. It iterates over all agents inside the scene.

- Hence it has a complexity of $O(N^2)$.

5. desired force:

- This module calculates the force between this agent and the next assigned waypoint. If the waypoint has been reached, the next waypoint in the list will be selected.

- At the moment, a visited waypoint is pushed back to the end of the list, which means that the agents will visit all the waypoints over and over again.

The main behavioral functionalities to be incorporated is thus implemented in the ped_agent module. The agents are made to move according the complex grid geometry based architecture called *Alan Turing Building Architecture* which is implemented in the coppito source module.

As aforementioned, the .xml file contains a generic description of the topology of the building that is to be modeled and used for rendering simulation. The coppito.xml is the file used in our case for the modeling of our present scenario. This information is retrieved by the loadscene module to extract the exact graphical plot points to draw the building.

The following figure 3.2 depicts the environment of PedSim and the coppito 0 building that is modeled.
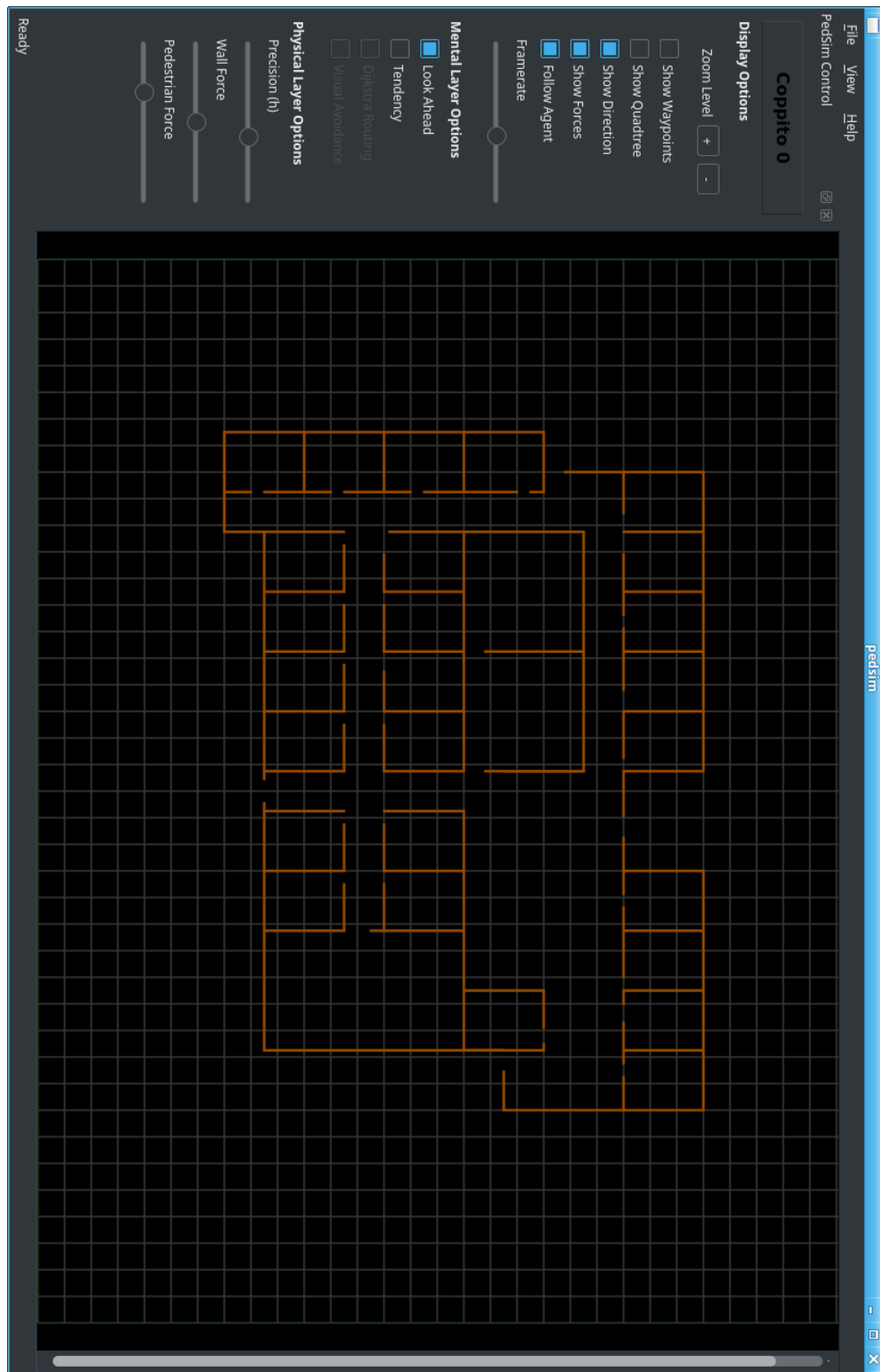
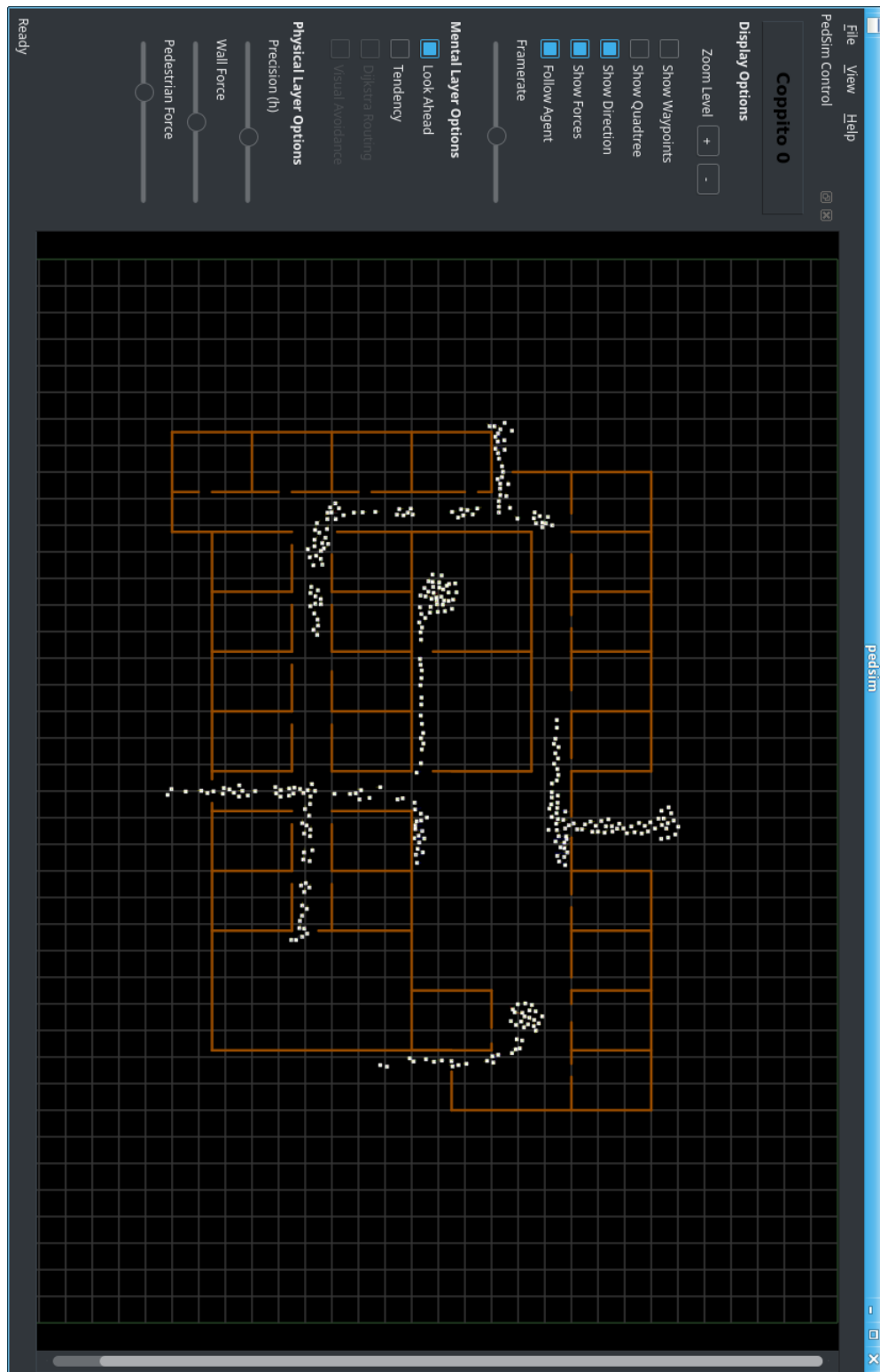Figure 3.2: Grid Representation of Coppito 0 within the PedSim Environment

Figure 3.3: Microscopic Agent Simulation in PedSim

The above figure 3.3 represents the microscopic agents being simulated within the PedSim environment, as the agents are subjected to the various constraints and are tasked to evacuate to the safe zone "**E**".

PedSim also provides a quick and easy way to change certain key constraints within its front end QT client. The following figure 3.4 depicts the variables that can be changed in real time whilst the simulation is being processed.
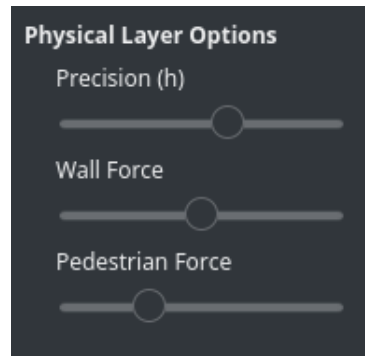

Figure 3.4: Real Time Variable Constraints in PedSim

The above figure depicts 3 variable sliders that can be applied to the simulation in real time. The three sliders are explained as follows:

1. precision(h) - this represents the time steps τ

2. Wall Force - this force determines how close to the obstacles they can tread, this is extremely useful in modelling situations like fire, heat etc., in cases where the walls of the topology should not approached due the nature of the disaster or situation.

3. Pedastrian Force - This force essentially determines the social forces among the participating agents. The more the pedastrian force the less intermixing happens between the agents. This variable constraint is especially important in modelling cell capacities and in determining flow dynamics.

The following subsection describes in detail the algorithm that is used in the simulation of the various scenarios.

3.3   Algorithm Description

All the agents within the PedSim environment are modeled and simulated according to certain realistic constraints. These constraints form the backbone that defines how the agents interact with one another and the environment that surrounds them. To make things simpler, the modeled topology is subdivided into grids. These grids are further divided into atomic square divisions of certain length and breadth. For convenience, we term these square divisions as cells. Constraints such as capacity, flow etc. are imposed on these cells to create a flow dynamic that is closer to the realistic simulation of an emergency evacuation. Within the premise of the running program, these are plot points which are manipulated by the program.

The coppito module pipelines this plot information for further processing. The basic movement of agents are strictly modeled according to the algorithm presented in [20]. Various constraints are mentioned in the paper for the anaylsis of a shortest egress path during design time. However these constraints alone are not enough to make the simulations close to be realistic. Once the basic algorithm is explained, further conditions will be introduced and explained.

In the work by H.Muccini et. al [20], discussion of the linearization of the constraints for effectively reducing the evacuation time of the agents is mentioned. However since its redundant to use linearization within the pedsim framework to simulate for the scenario, we mainly consider only 3 of the below mentioned constraints.

$$y_j^t - y_j^{t-1} - \sum_{i:ij \in A} x_{ij}^{t-1} + \sum_{i:ji \in A} x_{ji}^{t-1} = 0 \tag{3.1}$$

$$j \in V, t \in T, t > 0$$

$$0 \le x_{ij}^t + x_{ji}^t \le c_{ij} \tag{3.2}$$

$$t \in T, ij \in A$$

$$0 \le y_i^t \le n_i \tag{3.3}$$

$$t \in T, i \in V$$

Where,

1. $T = \{0, 1, ..., \tau\}$, set of unit of time slots.

2. y_i^t = state of cell $i \in V$ at time $t \in T$, that is, the number of persons that occupy i at t.

3. n_i = capacity of cell i; it measures the maximum nominal amount of people that i can host at any time.

4. x_{ij}^t = how many persons move from cell i to an adjacent cell j in $(t, t+1]$; this gives the average speed at which the agent flow proceeds from cell i to cell j.

5. c_{ij} = capacity of the passage between cell i and cell j; this is the maximum amount of people that, independently on how many persons there are in cell j, can traverse the passage in the time unit.

The above conditions are a set of mandatory inclusive pre-requisites required to be included from *An IoT Software Architecture for an Evacuable Building Architecture* by H. Muccini et. al [20], since the experimentation of various scenarios are based on the aforementioned set of equations and conditions. Through this thesis we introduce 4 more constraints to be included along with this base algorithm. The specific nature and details of the additional constraints will be discussed in the next chapter, where the various different experiments and scenarios are carried and are analyzed. It must

also be mentioned that in order to be coherent with the mentioned work and also to analyze optimal egress paths the following condition is also considered in addition to the above 3:

$$\max y_0^\tau \tag{3.4}$$

The above condition is obviously included in order to maximize the number of evacuees during the disaster scenario.

3.4 Topology Formulation

The topology of the building as given in 3.1 is modeled according to the *Alan Turing Building Architecture* and as shown in the figure the whole Coppito 0 building is subdivided into cells of equal length and breadth. Each cell also has a cell capacity as defined above and is assumed that inter-connected cells have full passage, if not bound by walls or walls with reduced passage capacity(doors).

The coppito source module divides the entire topology into such grids that are composed of cells, whose initial plot point is taken from the loadscene module, which in turn takes its information from the coppito.xml file. To translate simple plot points to a complex cell based grid structure, a highly complex user defined data structure was developed to store this information.

The plot points that are stored into the .xml file is randomly plotted with the exception of the first starting point in order to facilitate a quicker render of the entire building. However for the processing of a cell based structure, this information cannot be random and hence the information is then once again converted to form a structured cell geometry.

Thus a complex user defined data structure is defined to hold all this information in place. For the sake of convenience and optimal storage of data, the coppito 0 building is divided into regions called blocks. Each block contains a set number of

cells and naturally have a variable number of cells. The whole topology is divided into 18 blocks and consists in total of 119 cells. Each cell is also of a specific dimension and has a certain cell length and a cell width. To understand what a block is, it can be easily explained diagrammatically as follows:
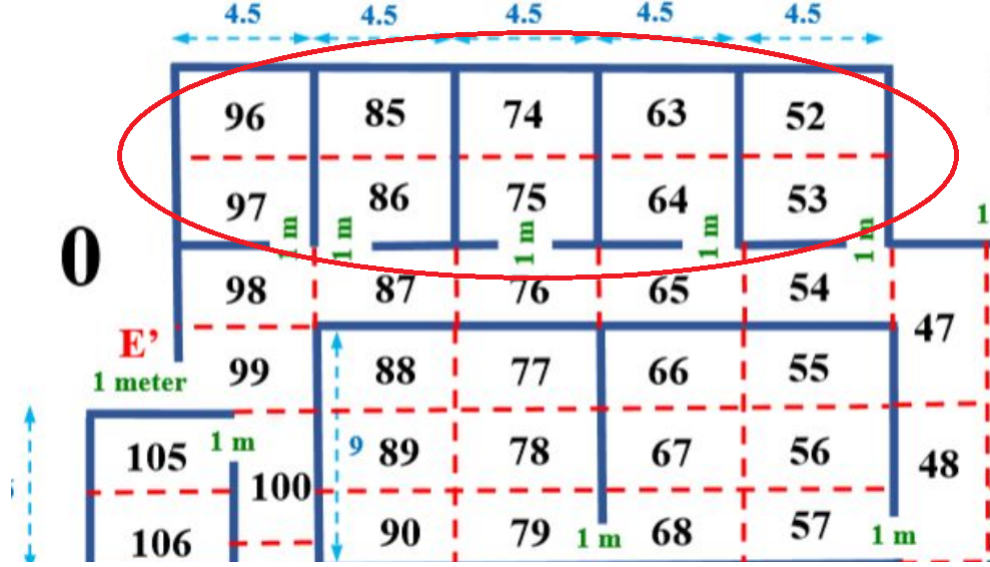

Figure 3.5: Groups of Cells Making a Block Structure

From the above figure 3.5, the red encircled circled area which includes the cells numbered 96, 85, 74, 63, 52, 97, 86, 75, 64, 53 form a block. Although a certain number of cells make up a block, the number of cells that make a block is often variable. Essentially one section of the topology is considered a block. This helps segregate different areas of the premises and identify them easily.

Further decomposing the cell, subdivides it into 4 edge lines, each edge line further decomposes into 4 vertices, an end to end x and y plot points. Along with these information are stored if the edge lines are walls and if they incorporate doors. This is highly useful for creating a tree structure pattern for path finding, used by agents during evacuation simulation. The final complex structure that is defined to hold this cell structure is explained in detail in the Appendix section. In order to dynamically

30

store the data of the cell information, the data structure used in place is a custom user defined 7 layered structure.

There are several sub-modules implemented in this custom made source file. The following are the following methods that are implemented as part of the module:

1. divide_cells

2. cell_structure_allocation

3. vertical_cell_allocation

4. non_standard_vertical_allocation

5. wall_allocation

6. wall_division_horizontal

7. wall_division_vertical

8. door_allocation

9. graph_tree_structuring

10. print_block

11. print_walls

12. print_door

13. print_graph

Further details regarding the technical and programmatic implementation of these modules are provided in the Appendix section of the thesis. The following section discusses the modification to the existing unoptimized algorithm from [20]. The different simulation results comparing macro and micro agent simulations are presented and analyzed. The resulting the implications of these observationss are discussed.

Chapter 4

Experimentation and Analysis

This chapter is intended to provide the experimental proof of the contribution. Through this chapter, the detailed explanation of the additional constraints that are added to the unoptimized algorithm from the previous chapter is provided. Next micro and macro agent simulations are presented pertaining to certain cases and scenarios to compare between the unoptimized and the optimized algorithms. The resulting observations and statistics are then presented for further analysis and the implications are then discussed.

4.1   Optimization through Constraints

From the equations 3.1, 3.2 and 3.3, the basic conditions for agent occupancy, flow control and cell capacity is discussed. However these conditions are hardly realistic when compared to modelling a real crowd of agents. In realistic scenarios, groups of people have altered behavior, confusion panic etc. There will be different forms of social constraints, varying movement speeds, and due to the impaired cognition, perceptions can change leading to different decisions that are made during a catastrophe. To model some of these realistic scenarios, it became evident that the inclusion

of certain more properties to the scenario was mandatory. The following conditions as mentioned in sub-section 1.4 is analyzed here:

1. Cell capacity specification by defining social distances

   - Although PedSim comes by default with a cell class, it does not go coherently with the topology, as cell divisions are highly subjective and can change their length and breadth according to the defined scenario. To cater for this, manipulation of the social forces constraint causes agents to maintain a certain "social distance" from each other. This can be used to our advantage to simulate cell capacities based on these social distances.

2. Definition of total area capacity and doors flow capacity constraints -congestion control

   - The algorithm presented in sub-section 3.3 unfortunately does not handle congestion. Providing additional constraint for area capacity, door flow capacity and passageway capacity can help reduce congestion and enable the re-routing of agents to alternate exits or routes in real time.

3. Simulating social attachment among some agents

   - With the introduction of the group constraint, social attachments can be modeled for a more realistic approach. For instance friends move together, a mother will most likely not be separable from a child etc. These groups can be defined during the simulation and then observed for the various real time decisions that these groups of agents make.

4. Setting the speed accordingly for various groups

   - By default PedSim models for microscopic agents and hence the entirety of the agents are considered as a single group. The movement speed for

these agents are varied across a distribution and the average speed set for the microscopic group of agents. By setting a variable speed constraint and with the introduction of groups, we model PedSim to cater for macro and micro agent simulation and thus support varying speeds according to the different group formations that can be defined during the simulation. In other words v_max is not fixed across all agents.

The PedSim library is extended to be inclusive of the algorithm from the sub-section 3.3 and the aforementioned constraints in order to simulate the various scenarios for micro and macro agents. The following sub-section provides a detailed analysis of the various scenarios that are simulated within the PedSim environment.

4.2   Simulation and Scenario Analysis

Bibliography

[1] Sameera Abar, Georgios K Theodoropoulos, Pierre Lemarinier, and Gregory MP OHare. Agent based modelling and simulation tools: a review of the state-of-art software. *Computer Science Review*, 24:13–33, 2017.

[2] Prasad Annadata, Wisam Eltarjaman, and Ramakrishna Thurimella. Person detection techniques for an iot based emergency evacuation assistance system. In *Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, MOBIQUITOUS 2016, pages 77–82, New York, NY, USA, 2016. ACM.

[3] William H Avery and Jamie Soo. *Emergency/disaster guidelines and procedures for employees*. CCH Canadian Limited, 2003.

[4] Georgia Bateman and Arnab Majumdar. An assessment of the simulation modeling of emergency evacuations in sports stadiums. 01 2018.

[5] Neeti Bhargava, V Katiyar, Mukat Sharma, and P Pradhan. Earthquake prediction through animal behavior: A review. *Indian Journal of Biomechanics: Special Issue*, 78, 01 2009.

[6] Po-Han Chen and Feng Feng. A fast flow control algorithm for real-time emergency evacuation in large indoor areas. *Fire Safety Journal*, 44(5):732–740, 2009.

[7] Jehyun Cho, Ghang Lee, Jongsung Won, and Eunseo Ryu. Application of dijkstra's algorithm in the smart exit sign. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 31, page 1. IAARC Publications, 2014.

[8] Wonjoon Choi, Horst W Hamacher, and Suleyman Tufekci. Modeling of building evacuation problems by network flows with side constraints. *European Journal of Operational Research*, 35(1):98–110, 1988.

[9] Damon P. Coppola. Chapter 1 - the management of disasters. In Damon P. Coppola, editor, *Introduction to International Disaster Management (Third Edition)*, pages 1 – 39. Butterworth-Heinemann, Boston, third edition edition, 2015.

[10] Damon P. Coppola. Chapter 4 - mitigation. In Damon P. Coppola, editor, *Introduction to International Disaster Management (Third Edition)*, pages 224 – 274. Butterworth-Heinemann, Boston, third edition edition, 2015.

[11] Antoine Desmet, Huibo Bi, and Erol Gelenbe. Simulation of congestion management for emergency evacuation.

[12] John El Khoury, Jean-Paul M Arnaout, and Caline El Khoury. Evacuation of a highly congested urban city: the case of beirut. 2015.

[13] E. Gelenbe, , and E. Seref. Cognitive packet networks. In *Proceedings 11th International Conference on Tools with Artificial Intelligence*, pages 47–54, Nov 1999.

[14] Doungmo Goufoc, Abiola P Chokkid, and Maurice V Yengoa. Agent-based modelling and simulation for evacuation of people from a building in case of fire. *Procedia Computer Science*, 130:10–17, 2018.

[15] Steve Gwynne, Edward R Galea, M Owen, Peter J Lawrence, and L Filippidis. A review of the methodologies used in the computer simulation of evacuation from the built environment. *Building and environment*, 34(6):741–749, 1999.

[16] Shinkishi Hatai and Noboru Abe. The responses of the catfish, parasilurus asotus, to earthquakes. *Proceedings of the Imperial Academy*, 8(8):375–378, 1932.

[17] Motoji Ikeya, Hiroshi Furuta, Naemi Kajiwara, and Hiroko Anzai. Ground electric field effects on rats and sparrows: Seismic anomalous animal behaviors (saabs). *Japanese journal of applied physics*, 35(8R):4587, 1996.

[18] Margrethe Kobes, Ira Helsloot, Bauke De Vries, Jos G Post, Nancy Oberijé, and Karin Groenewegen. Way finding during fire evacuation; an analysis of unannounced fire drills in a hotel at night. *Building and Environment*, 45(3):537–548, 2010.

[19] Marin Lujak, Holger Billhardt, Jürgen Dunkel, Alberto Fernández, Ramón Hermoso, and Sascha Ossowski. A distributed architecture for real-time evacuation guidance in large smart buildings. 2017.

[20] Henry Muccini, Claudio Arbib, Paul Davidsson, and Mahyar Tourchi Moghaddam. An iot software architecture for an evacuable building architecture. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[21] Francesco Palmieri, Massimo Ficco, Silvio Pardi, and Aniello Castiglione. A cloud-based architecture for emergency management and first responders localization in smart city environments. *Computers & Electrical Engineering*, 56:810–830, 2016.

[22] Ashutosh Trivedi and Shrisha Rao. Agent-based modeling of emergency evacuations considering human panic behavior. *IEEE Transactions on Computational Social Systems*, 5:277–, 03 2018.