Agent-based Crowd Simulation, Analysis and Optimization of Pedestrians Flow during Hajj

By

Muhammad Haris

00000118910

Supervisor

Dr. Imran Mahmood

Department of Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology

Approval

It is certified that the contents and form of the thesis entitled "Agent-based Crowd Simulation, Analysis and Optimization of Pedestrians Flow during Hajj "submitted by Muhammad Haris have been found satisfactory for the requirement of the degree.

Advisor: Dr. Imran Mahmood

Signature: ____________________

Date: ____________________

Committee Member 1:

Dr. Muhammad Moazam Fraz

Signature: ____________________

Date: ____________________

Committee Member 2:

Dr. Asad Waqar Malik

Signature: ____________________

Date: ____________________

Committee Member 3:

Dr. Muhammad Muddassir Malik

Signature: ____________________

Date: ____________________

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Muhammad Haris

Signature: ____________________

Acknowledgment

I pay my gratitude to Allah almighty for blessing me a lot and without His guidance I couldn't complete this task. I am thankful to Dr. Imran Mahmood for guiding and encouraging me to complete my thesis. His timely and efficient contributions helped me to shape the thesis into its final form and I express my gratefulness for his sincere supervision all the way.

I am also thankful to Department of Computing, and the teachers for providing me with an academic base, which enabled me to complete this thesis.

I'd like to thank my parents who have provided me every comfort of life and their utmost affection and support made me reach where I am today.

Publications

Conference Publications

I. Mahmood, M Haris, H. Sarjoughian, "Analyzing Emergency Evacuation Strategies for Mass Gatherings using Crowd Simulation and Analysis framework: Hajj Scenario", *Proceedings of the Proceedings of the 2017 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, May 2017, NTU Singapore.

Table of Contents

List of Abbreviation

PSS	Public Safety and Security
CA	Cellular Automata
SFM	Social Force Model
ABM	Agent-based Modeling
GA	Genetic Algorithm
HS	Harmony Search
HMM	Harmony Memory Matrix
SRD	Shortest Regional Distance
CGP	Cartesian Genetic Programming
ANN	Artificial Neural Network
APL	Anylogic Pedestrian Library

List of Tables

List of Figures

Abstract

Hajj is one of the largest mass gatherings where Muslims from all over the world congregate in Makah each year for pilgrimage. Such large gatherings always have risk of confrontation with disasters, either natural or man-made. In order to perform Hajj rituals, Pilgrims move around the various religious sites (Mina, Safa-Marwah, Arafat and Muzdalifah). In the past few years, thousands of casualties have occurred due to stampede or chaos during different Hajj rituals, especially during the Circumambulation of Kaba (Tawaf) and stoning the devil in Mina. In order to make these events safe and risk free, we have to analyze and propose an appropriate evacuation strategy to alleviate the risk of causalities. However, it is a daunting research problem to identify optimal course of actions during emergency disaster management when several constraints are involved, especially when dealing with a crowd of massive size.

We propose an Agent-based Crowd Simulation & Analysis framework that incorporates Anylogic simulation environment to design a mass gathering scenario e.g. Hajj and simulate large-scale crowd movements to evaluate different evacuation strategies for mass gatherings. The key features of proposed framework include (i) design of spatially explicit environments such as buildings, stadiums and mosques and incorporating obstacles and constraints of the crowd movements (ii) simulating large-scale crowd behavior using agent-based modeling paradigm (iii) interoperability with external optimization modules to integrate different optimization algorithms with simulation model to determine the optimal evacuation plan. We present a case study of Hajj scenario as a proof of concept and a test bed for identifying and evaluating different optimal strategies for large-scale crowd evacuation.

Keywords: *Public safety & security, Mass gatherings, Emergency Evacuation Strategies, Agent-based Crowd Simulation, Optimization, Genetic Algorithm, Harmony Search Algorithm*

Chapter 1

Introduction

This chapter provides the information about the topic which is being studied. It describes the definitions and concepts of crowd simulation and some preliminaries to get understanding about this thesis.

Crowd simulation is playing vital role to simulate large number of entities or characters. With the advancement in crowd simulation, mass gatherings such as sports events, religious obligations and cultural events can be modeled to analyze the different factors for crowd convenience and safety. Such crowd simulations help in analyzing security risks, pedestrians' behavior and building risk assessment in order to mitigate tragedies.

1.1 Mass Gathering

A mass gathering, as defined by World Health Organization is: "A planned or unplanned event at a specific location, attended by a huge number of people for a common purpose. This number is sufficient enough to strain the planning and response resources of the community, state or a nation hosting that event" [1]. There are different mass gathering events which are organized every year or after a specific period of time across the country or abroad. Some events are organized periodically at the same location and some on different locations. For instance, Muslims religious obligation Hajj and national parades are the recurring events organized at the same places. Whereas Sport events like Olympics, annual dinners, celebrity shows, political rallies and social protests are mass gatherings that occur at different places. These special events are attended by thousands of people and are always vulnerable to incidents such as terrorist attacks, crowd crushes and natural disasters. Due to these incidents a lot of people can be vanquished by death or injured. Thus, these events require contingency planning and emergency support from Public Safety & Security (PSS) organizations. In general, special events or mass gatherings should be planned in advance for better management. Event organizers and responsible of public safety need to plan together and define the functional stakes (Emergency management, rescue and fire, public help) in the event. While event managers work for the promotion and planning of an event whereas security agencies confirm the success of events by making them safe and secure.

Hajj is one of the largest recurring mass gathering, where about 3 million Muslims congregate in Makkah (Saudi Arabia) every year, from over 140 countries, with five to 40 days span [2]. As pilgrims inflow is increasing every year so Makkah and Madinah are becoming the site of extreme crowd densities. Such mass gatherings are quite vulnerable to serious disasters including stampede, construction failures, fires and communicable hazards [3]. Tawaf is one of the Hajj ritual where mass gathering can be observed and simlarly stoning the devil at Jamart bridges also involves large number of people and have chances of encountering stampede or crowd crushes. These incidents are occured due to some factors: high densities above 7 or 8 [4], persons per square meter, complex motion flows, extended stays at Hajj sites and suffocating enviroment [5]. These factors restrict the movement of people in concentrated regions and due to this immobility stampede can occur which can cause hundreds of deaths [6]. Moreover, in Hajj or Umrah season few pilgrims are ambling along beacuse of health issues which makes them prone to the mortality risks. In the past, a large number of disasters took place during Hajj; few of them are described here. In July 1990, due to improper crowd control, 1426 people were killed in a crowd crush. In 1997, 1500 injuries were observed along with 343 deaths due to fire explosion. Similarly, in 2006, more than 300 deaths took place in stampede during stoning the devil at Mina valley. In 2015, during Hajj preprations crane was toppled in the courtyard of Masjid-Al-Haram (great mosque of Makkah) causing a total of 512 casualties. Again, in September 2015, more than 2000 pilgrims were crushed due to stampede at Mina while stoning the pillars [3]. These incidents can be avoided through proper coordination with international and local Public Safetry and Security (PSS) organizations. PSS is mission of government to ensure the safety of mass gatherings, organizations and other institutions against threats for their well-being. *Pulic Safety* refers to the welfare and protecting people from natural or man-made disasters or environmental hazards whereas *Public Security* ensures protection of socities against terrorism and other crimes. Plans carried out by PSS teams during event management must be shared with security teams and reinforced through trainings. Figure 1 presents an overview of different roles of PSS organizations.

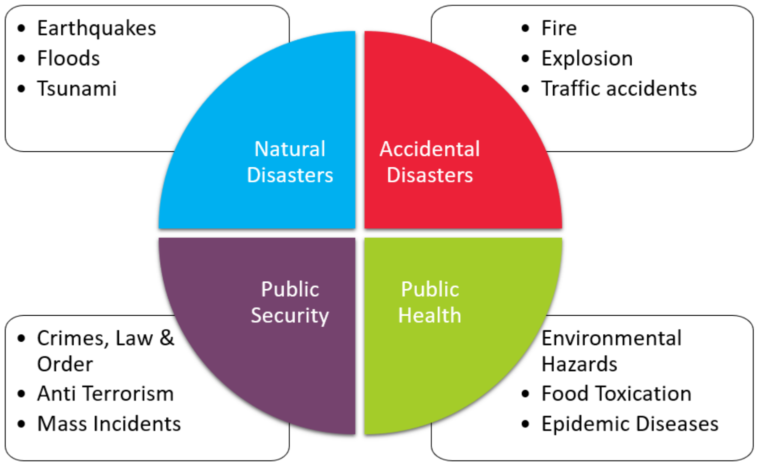


Figure 1: Roles of PSS organizations

In order to alleviate the risk and make mass gatherings successful there should be disaster management plan which comprises of 5 major parts [7]:

1. Monitoring
2. Mitigation
3. Preparedness
4. Incidence Response
5. Recovery

Figure 2: PSS crowd control and management steps

1.1.1 Monitoring

Monitoring is a continuous process of observing real-world activities. It includes regular observation and gathering information of different activities in different places i.e. crowd flow. Monitoring also includes systematic process of collecting, analyzing and using information to track an event's progress towards reaching its objectives and to guide management decisions. Monitoring helps event organizers to experience from past and take necessary steps to overcome the issues in future.

1.1.2 Mitigation

Mitigation includes identifying the vulnerability of an event and plan ahead to minimize the risk. In this phase, necessary measures are taken to reduce the impact of disaster or emergency before its occurrence. These plans can be security planning, evacuation management and building risk assessment. Mitigation involves government, army services and other security agencies to evaluate what tools and trainings are required to make the risk-free environments.

1.1.3 Preparedness

Preparedness addresses the strategy getting prepared in advance and implements the operational response immediately after occurrence of emergency. Preparedness activities include managing the teams and roles to aid people during disaster, formulating evacuation plans and providing training for disaster responders. It also includes guiding the public before an event and plan accordingly to respond during hazard and not to get panic.

1.1.4 Incidence Response

Response deals with the disastrous situation to provide support at the affected place. This phase especially deals with saving lives, firefighting, search and rescue operations and prevention from huge loss. Operational plans made during mitigation and preparedness phase are to be implemented at this stage. Due to the practical implementation during hazards, this phase becomes more crucial than previous phases. Numerous safety organizations, such as army forces, ambulances and fire brigades are deployed to combat and overcome the situation immediately. The main goal of these disaster management teams is to prevent further damage or injuries and planning of recovery activities.

1.1.5 Recovery

When an incident takes place, it can be calamitous and long-lasting. Recovery phase, which comes after response, deals with restoration at concerned area and rehabilitation of the loss observed during disaster. Local governments and event organizers make necessary arrangements to repair social, physical and economic damage. Recovery also refers to long term reconstruction of healthcare facilities, buildings and other communication facilities.

1.2 Challenge

Mass gathering, such as Muslims religious obligation, Hajj, sports events like Olympics, annual dinners and political rallies are always vulnerable to incidents and can claim thousands of lives due to lack of management. In past few years, a lot of incidents have been observed during large events. Such as stampede, fire explosion and construction failures have claimed thousands of lives. Such disasters are inevitable and need to be mitigated to minimize the casualties. Considering the above issues, provision of safety for mass gatherings is a challenging issue. The need of the time is to develop a system which helps emergency crowd evacuation in order to minimize fatalities.

1.3 Problem Domain

As the size of crowd is increasing in large events in different areas day by day, such places are becoming progressively vulnerable to hazards. Due to overcrowding, high densities of people are observed at big events as number of people per square meter increases to 7-8 which is normally 4 people per square meter. Therefore, formulation of security plans to determine the solution of emergency situations in reality without any prior planning for crowded areas is a daunting problem. Real time validation of security plans for disastrous situations is costly, leads to wastage of resources and can pose risk to human lives. Moreover, real world experiments can't provide us deep insights of system and thus are not feasible to experiment different solutions for future planning due to its high-risk factor.

1.4 Problem Statement

From above discussion we can conclude the problem statement as:

"The need of the time is to model large-scale crowd in a spatial environment with real dimensions as baseline to evaluate and compare different evacuation strategies which helps in evacuating people during emergency situations in order to alleviate the causalities."

1.5 Solution Domain

Following research area gives brief overview of approaches used in this thesis.

1.5.1 Modeling and Simulation

M&S is playing vital role for PSS management and decision-making purposes. Disastrous or mission critical situations can't be experimented in real, so simulation models are becoming useful in replicating real scenarios in the virtual world for analysis purposes.

1.5.2 Crowd Simulation

Crowd simulation is playing pivotal role to simulate large-scale pedestrians and their behavior in different fields including urban planning, evacuation handling of different areas and military simulations. We develop a spatial environment according to the real-scale dimensions for large-scale crowd using agent-based modeling.

1.5.3 Optimization Algorithms

Different optimization algorithms will be used for crowd simulation model to formulate the optimized evacuation plans for crowd safety.

1.6 Solution Statement

It is impossible to experiment disastrous situations in real to analyze different factors for safety measurements. Such experiments are highly risky, unsafe for humans and too costly. Modeling and simulation plays essential role to model the real-world scenarios in virtual world with some abstractions to analyze key factors for future plans.

We proposed an agent-based crowd simulation and analysis framework that incorporates Anylogic space markup and pedestrian library to design spatial environment and simulate large-scale crowd movements to evaluate different evacuation strategies for mass gatherings. Proposed framework consists of three layers (i) design of spatial environment and crowd simulation model (ii) provides interoperability to integrate simulation model to run different evacuation strategies (iii) ability to implement and integrate different optimization algorithms with simulation model in order to determine the optimal evacuation plan.

1.7 Key Contributions

We propose an agent-based crowd simulation and analysis framework that encompasses comprehensive approach (i) to model large number of pedestrians and different spatial environments such as stadiums, shopping malls and religious places (ii) to simulate high density crowd and their movements in spatial environments (iii) implementing and analyzing different crowd management strategies such as crowd evacuation during hazardous situations.

1.8 Research Impact

Our proposed framework is a key concept to formulate crowd evacuation and management strategies for overcrowded areas. Providing better evacuation or strategies, in order to mitigate possible causalities caused by overcrowding at large public events. Our solution can be deployed at any PSS organization and can benefit to develop crowd management plans.

1.9 Thesis Organization

Rest of the thesis is organized in following chapters

1.9.1 Chapter 2: Background

Chapter 2 provides brief overview of crowd simulation and evacuation and explains its impact on real life. This chapter also explains different tools which can be used to simulate crowd in complex environments. Moreover, few preliminary concepts, used in methodology chapter, have also been discussed.

1.9.2 Chapter 3: Literature

This chapter explains the work done so far related to crowd simulation and evacuation and summarizes the systematic approaches of crowd simulation and evacuation and also formulates research directions for this dissertation.

1.9.3 Chapter 4: Methodology

Our proposed crowd simulation and analysis framework has been presented in this chapter. As we have used different libraries to build our proposed framework, so technical background of these libraries has been discussed first. Furthermore, working of different optimization algorithms used for crowd evacuation have been discussed.

1.9.4 Chapter 5: Simulation and Results

This chapter is dedicated to demonstrating the functionality of our proposed framework. A case study of Hajj scenario has been used to provide the proof of concept. The chapter is concluded by results and their detailed discussion.

1.9.5 Chapter 6: Model Evaluation

This chapter deals with the evaluation of the model which has been presented in chapter 5. We have performed automated unit and integration testing to verify the functionality of our framework.

1.9.6 Chapter 7: Conclusion and Future work

The work accomplished in this thesis has been concluded in this last chapter. Moreover, this chapter also describes the future directions which can be done ahead to this work.

Chapter 2

Background

This chapter provides the information about the topic which is being studied. It describes the definitions and concepts of crowd simulation and some preliminaries to get understanding about this thesis.

2.1 Crowd Simulation

"Crowd simulation is the process of simulating the movement (or dynamics) of a large number of entities or characters" [8]

In the past decades, efforts have been made to simulate human motion and behavior to study crowd dynamics. Virtualization of human characters or crowd simulation is widely used to create virtual reality games, animated movies, and is also used for infrastructure designing. It has become popular area of interest among researchers and management companies for taking security measures and evacuation planning in case of mass gatherings. Crowd simulation is also playing important role to virtualize pedestrians at large-scale and simulate their behavior in different fields including urban planning, evacuation handling of different areas and military simulations. Traditional planning of urban areas depends upon hand crafted maps and sketches but advancement in crowd simulation tools has made the development of urban environment and planning robust. Armed forces are utilizing the crowd simulation for potentially analyzing the combatant situations and military trainings. Construction of crowd movements and their behavior in virtual environments can be applied to identify the possible risks and test evacuation plans for crowd safety.

Although crowd simulation deals with modeling of individuals on abstract level, but individual's characteristics and behavior are important factors to develop realistic models. In dense crowd, individual's behavior may be affected by other pedestrians and external factors. These external factors have significant importance in analyzing behavioral aspects [9].

2.1.1 Crowd Simulation Models

Crowd simulation approaches describe the way how simulated individuals will perform in an environment. Three crowd modeling approaches have been discussed below.

2.1.1.1 Flow-Based Approach

In this approach crowd is considered as a whole (homogenous entities) and abstracts from individual's characteristics and behavior. This approach is mainly used to simulate dense crowd in different environments to analyze the pedestrians' flow in complex situations such as evacuation process [10] e.g. macro crowd simulation.

2.1.1.2 Entity-Based Approach

Individuals are modeled as homogenous entities using entity-based approach. This approach also incorporates social, psychological and global/local laws which affects the motion of entities in physical world such as Helbing's Social Force Model e.g. meso crowd simulation [11].

2.1.1.3 Agent-Based Approach

Agent-based approach enables simulation of autonomous entities in interacting environment. Entities modeled using this approach are intelligent and communicate with each other. Agents can react to the situation on their own and make decisions according to the set of rules. Using this approach, modelers are given complete freedom to simulate attributes and behavior of each individual e.g. micro and meso crowd simulation [9].

2.2 Crowd Simulation Tools

Different tools have been developed to simulate crowd for complex scenarios. These tools help modelers to design different spatial environments for large-scale crowd for analysis purposes.

2.2.1 Pedestrian Dynamics

Pedestrian Dynamics crowd simulation software is used to analyze and optimize large crowd flows. Its key features include:

- Support large-scale crowd simulation
- Integrated 2D&3D models with fast simulation runs
- Import of industry standards (CAD, XML and many more) [12]

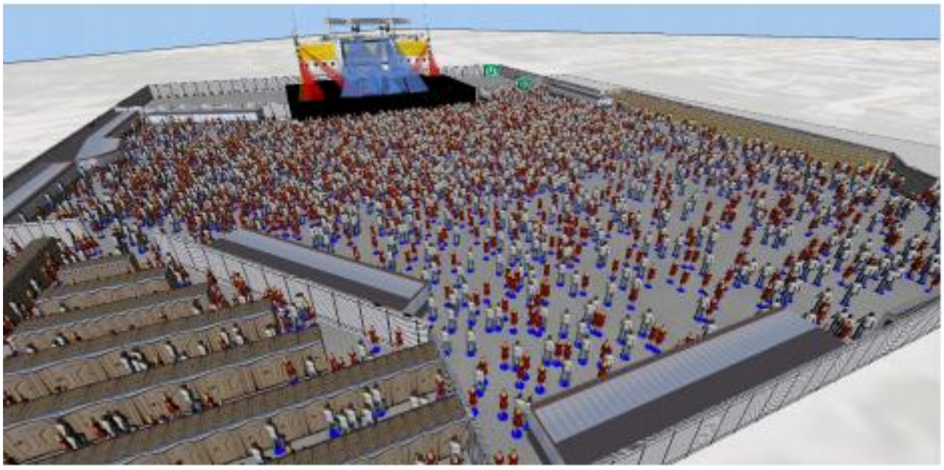


Figure 3: Crowd Simulation using Pedestrians Dynamics Software [12]

2.2.2 PedSim

PEDSIM is a microscopic pedestrian crowd simulation software. It is suitable to develop different crowd simulations such as indoor evacuation simulation and large scale outdoor simulations. It is also used for quantitative measurements like pedestrian density or evacuation time.

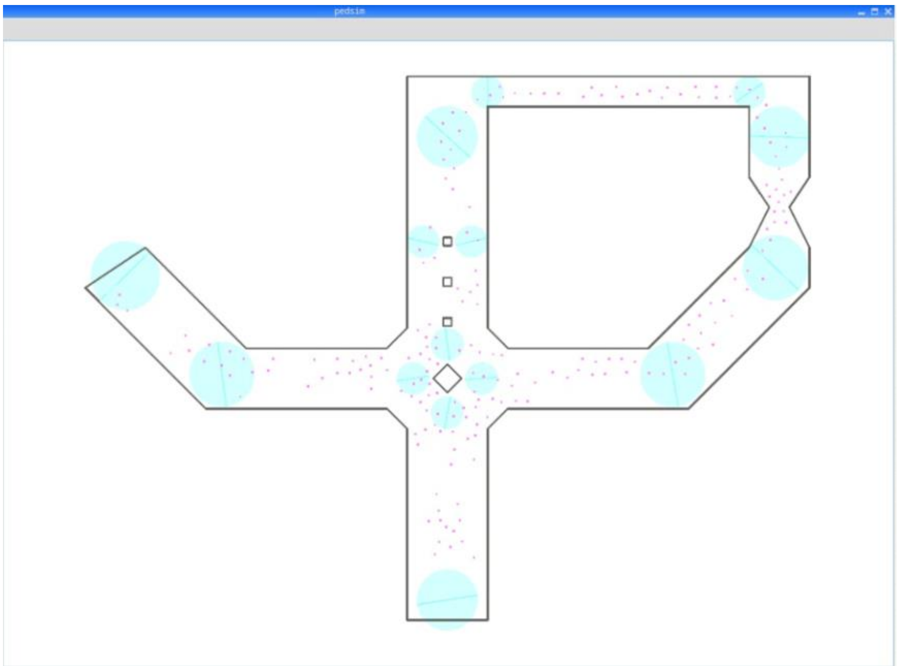


Figure 4: PedSim Crowd Simulation [13]

2.2.3 SimWalk

This is an agent-based simulation environment used for pedestrian simulation products for public transport, aviation, sports venues, architecture, urban planning and evacuation.

Figure 5: Pedestrian Modeling using Simwalk

2.2.4 AnyLogic

AnyLogic [14] is a versatile simulation tool with graphical interface that allows modeler to quickly model complex environments. Anylogic Simulation environment provides both a user friendly Integrated Development Environment (IDE) as well as an efficient simulation engine that allows the modelers to quickly create and simulate high fidelity models of complex systems. It supports different modelling techniques such as Discrete Event, Agent Based and System Dynamics. We can develop complex hybrid models with the combination of discrete event, agent-based and system dynamics. Anylogic provides user-friendly interface, Java-based development environment and a set of multipurpose component libraries, which all together help to robust the modelling process [15]. It also facilitates modeler to integrate simulations with external environments.

Figure 6: Pedestrian Simulation using Anylogic

In this research we are using Anylogic simulation software for the development of proposed framework. Its rich features help us to build complex simulations robustly. We can design custom agents in Anylogic using agent-based paradigm whereas pedestrian dynamics and PedSim doesn't provide agent-based modeling approach.

2.3 Crowd Evacuation

Extreme crowd conditions can emerge due to mass gathering, particularly in suffocated environments. Normally evacuations take place smoothly and in a polite manner but there are chances of haphazardness leading to chaos. In these state of affairs, term "panic" comes across our mind however, from a scientific perspective this term might be challenging. But our focus will be on the issue of crowd refinements at high solidities and under anxiety, rather than on answering whether panic actually occurs or not. Mostly the normal behavior of pedestrians vanishes when they face emergency situations. As people try to leave the incident site as fast as possible, the desired velocity increases which leads to some characteristic formations. As nervousness increases there is less concern about comfort zone and finding the most convenient and shortest way so people, not knowing the structure of incident site, try to leave it. They would run and try to exit from the inlet which they used as an entrance before without knowing that other exits might be in their close proximity than the entrance [16].

2.4 Optimization

Optimization is a method for discovering the best feasible solution. Optimization is concerned with finding the values of input through which the "best" output can be achieved. The definition of the term "best" can vary in different problem domains, however in mathematical terminology, it is said that by varying the input parameters one can maximize or minimize the objective functions.

Search space is made up of all the possible solutions or values which the input can take. In this pool of search space there lies a point or set of points which give the best solution. The main objective of optimization is to find that point or set of points that give the best solution [17].

2.4.1 Genetic Algorithm

A genetic algorithm (GA) is a meta-heuristic search based on biological inspired process used to solve optimization problems. This algorithm works according to the Charles Darwin theory evolutionary genetics and natural selection. GA was proposed by John Holland in 1960's and his students assisted him for development of newly invented approach. In GA, a population of candidate solutions (called individuals) is evolved towards better solutions. The process starts from an initial population of individuals also known as generation which is randomly generated and evolves in multiple iterations for better solution. These individuals are composed of set of bit strings known as genes. The set of genes is called chromosome which is basically a candidate solution [18].

Once the population of chromosomes is generated the fitness of each individual is evaluated. The fitness can be defined as a value of the objective function of optimization problem. This objective function needs to be minimized or maximized according to the problem's requirement (i.e. minimizing evacuation time). The better fit solutions are selected from the population of each previous generation as parents of next generation and these parents are passed through the process of cross over and random mutation to produce offspring for new generation. The new generation of candidate solutions is then used in the next iteration to evaluate the fitness and the same process continues until maximum number of generations has been produced, or a termination condition has been reached.

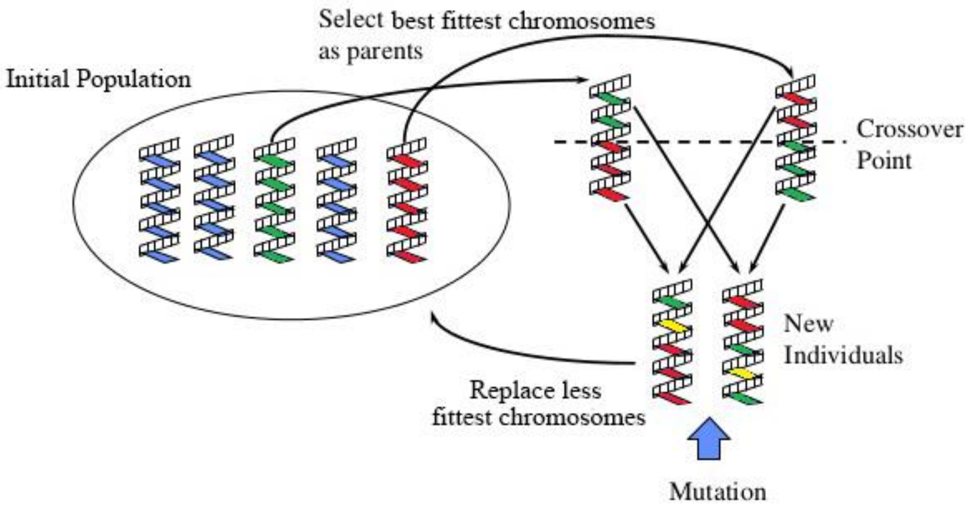


Figure 7: GA Crossover Example

2.4.2 Harmony Search

Harmony Search (HS) is a metaheuristic approach inspired from musical instruments and try to find the best state of harmony. In musical process, harmony can be defined as multiple pitches being played at a time. Musical performances obtain the best state of harmony which is determined by aesthetic principles, just as optimization problems obtain the global optimum solution which is determined by objective function [19].

Chapter 3

Literature Review

This chapter helps in explaining how similar is the work with that of others. Moreover, it contributes in the understanding and development of the area of research.

Table 1 summarizes the literature work into different categories whereas details of these papers are described later.

Table 1: Literature Review

Author (s)	Category	Paper Description	Key Features
[20]	**Crowd Simulation**	Explained the advantages of crowd simulation	Identified crowd simulation usage in real-life problems and explained benefits of crowd simulation for mass gatherings
[21]		Explained models for micro crowd simulation	Described cellular automata (CA), Social Force Model (SFM) and Velocity-based model (VBM)
[22]		Provided basics of crowd simulation models	Described that CA and ABM are widely used approaches for crowd simulation
[23]		Focused on crowd simulation model	Explained the working of CA for crowd simulation
[24]		Described drawbacks of CA	Identified that CA is unable to simulate large-density crowd, improper pedestrians motion and obstacles placement due to fixed number of cells
[11]		Provided explanation of Social Force Model (SFM) and its usage with crowd simulation model.	Described that SFM is realistic than CA, in which crowd moves in continuous space and also reacts to different external objects
[25]		Explained Agent-based Modeling (ABM)	Enabled integration of human decision-making phenomena in crowd models. Used ABM help modeler to create autonomous entities, which can adopt and react according to the situation
[26]		Based on micro Crowd Simulation	Simulated the crowd in Makkah performing Tawaf using CA
[27]		Provided micro Crowd Simulation	Provided simulation of crowd performing Tawaf using multi-agent system
[28]	**Crowd Evacuation**	Proposed evacuation framework	Proposed evacuation framework based on ABM and GA

[29]		Proposed evacuation framework	Proposed crowd evacuation framework using ABM and optimization approach
[30]		Crowd evacuation using wearable device	Introduced wearable device life belt to guide people for exit route selection during disaster or emergency
[31]		Crowd evacuation with leader-follower approach	Proposed leader-follower approach to guide people during disaster and exit gates are determined using A* algorithm
[32]		Crowd evacuation during fire incident	Developed an evacuation system using Artificial Neural Network (ANN)
[33]		Emergency disaster planning	Proposed intelligent shelter allotment system for disaster planning
[34]		Evacuating crowd using multi-agent system	Crowd evacuation using Belief-Desire-Intention framework and ABM

3.1 Crowd Simulation

Crowd simulation can be applied to solve several real-life problems. In dense crowd environments (shopping malls, concerts or Tawaf movement in Holy city), uncertainties can always occur that can lead to disasters or chaos. Public Safety and Security (PSS) organizations should maintain a high level of preparedness in order to mitigate the disasters. As emergency situation drills need the exposure of real people and the actual environment, so, it is very challenging and risky to experiment such sort of situations in real to formulate the evacuation plans for public safety. If we actually experiment such serious situations, then it can be detrimental for human lives and also wastage of useful resources. Thus, using simulation we can model crowded environments, experiment and analyze real-time problems to take the safety precautions which takes less time, utilizes cost-effective resource and is also non-adverse to human lives [20].

Crowd simulation model can be categorized in three levels: microscopic, mesoscopic and macroscopic.

3.1.1 Micro Crowd Simulation

The widely used category is micro crowd simulation which simulates individual's behavior and their interactions. This category is commonly used in evacuation scenarios to observe the individual's behavior in detail.

In past few years, a lot of models have been proposed to simulate crowd motions. Dorine and Winnie describe few of the models to simulate pedestrian movements. They focused on Cellular Automata (CA) models, Social Force Models (SFM), Velocity-Based Models (VBM) and Behavioral models [21]. Whereas cellular automata, social force and agent-based models are commonly used approaches for complex crowd simulations [22].

CA was invented by von-Neumann in 1966. CA is a discrete space model and is represented in the form of grids. Each cell of the grid is occupied by one person and that person can move to any of the 8 adjacent cells. CA approach is widely used in simulating large-scale crowd movements in shopping malls, airports and evacuation processes. Different objects or obstacles can be designed using CA in the defined cells. CA are preferred due to their representation of the crowd motions in a simple way in the form of fixed size cells [23]. But this method also has some limitations for example, unable to simulate large crowd density, improper pedestrians motion and obstacles placement due to fixed number of cells. As CA is a discrete space so individual's diameter should be according to the cell's size in order to observe the accurate motion behavior [24].

SFM is realistic than CA, in which crowd moves in continuous space and also reacts to different external objects. In pedestrian models, several external forces affect individual's movement. Helbing describes that motion of pedestrians is affected by other pedestrians and objects (walls, obstacles) as there is a repulsive force and every individual tries to keep certain distance from strangers and obstacles [11].

Agent-based modeling enables us to integrate human decision-making phenomena in crowd models. Agents are the autonomous entities; they can adopt and react in the situation accordingly and can communicate with other agents [25]. ABM can be integrated with CA and SFM in order to define the pedestrian's movement mechanism. ABM works more appropriately with SFM as agents move in continuous space and react to the other pedestrians and obstacles.

Talib and Haron [26] proposed a model using cellular automata approach to simulate the pilgrims in a circular Tawaf movement. Moreover, their proposed model was capable of handling thousands of pedestrians.

Sarmady and Haron [27] also simulated the crowd movements in the Tawaf area. In this model they have also focused on the characteristics and behaviors of individual pedestrians. In order to simulate pedestrians using multi-agent methods we need to develop a model of human movement process. For the development of human movement process, psychological processes must be taken into account to simulate pedestrians' behaviors.

3.1.2 Macro Crowd Simulation

Macroscopic models simulate crowd's behavior on abstract level and also enable us to simulate large density crowds. These models consider crowd behavior as a whole and don't incorporate their individual attributes or characteristics and even agents can't communicate with each other.

3.1.3 Meso Crowd Simulation

Mesoscopic models fill the gap between micro and macroscopic models. This category deals with large-scale crowd simulation and it incorporates indvidual's interaction and behavior at lower level of detail. In this category agents are represented in the form of groups such as families, friends and people with same age.

3.2 Crowd Evacuation

As we discussed earlier there is always a risk of stampede or crowd crush in large events so, there must be a mechanism to evacuate the crowd in case of emergency to minimize the loss.

A simulation framework is composed of agent-based models and Genetic Algorithm (GA) was proposed for crowd evacuation. The key idea was to partition the whole region into sub-regions and divide the crowd in regions based on their coordinates. GA was used to determine the optimal gate assignments for each region whereas crowd simulation model has been used to evaluate the GA's result. Crowd evacuates from the gate which is allocated to their region by GA. Later on, time sequence concept was introduced in the algorithm to minimize the overcrowding at exit gates. Each region was assigned a starting time and people of that region will start evacuating at defined time. Thus, time slots for each region will minimize the overcrowding at exit gates [28].

An evolutionary-algorithm based framework has been proposed for crowd evacuation planning. The proposed framework is comprised of agent-based simulation model and optimization approach - Cartesian Genetic Programming

(CGP) to obtain the heuristic rule (evacuation plan). The main idea was to divide the whole area into sub-areas and use heuristic rule dynamically to allocate the exit gates to each region. Pedestrians of each region will follow the assigned exit gate to evacuate from the scene. The framework has been trained and tested on different evacuation scenarios to obtain the generic heuristic rule [29].

Frescha and Zia proposed a wearable device named life belt. Authors have demonstrated life belt functionality by evacuating crowd from building. This device contains sensors which are used to detect the current location of each pedestrian and transmits it to central system. It then manipulates the current position and guide the next move to each person towards shortest exit. Life belt incorporates the situation dynamically and recommends exit gate to each pedestrian according to their current location [30].

Figure 8: Life Belt

JI and GAO proposed evacuation model with leader-follower approach and demonstrated its functionality using simulation model. A hall with different roomettes and obstacles was modeled and proposed approach was used to evacuate people from hall. When evacuation starts, leaders are decided and assigned to crowd dynamically whereas A* algorithm finds the exit paths. These paths are conveyed to each leader according to their location and rest of the people follow their assigned leaders to evacuate from the hall. Authors have proved that evacuation using leaders is efficient and reduces evacuation time as compared to the scenarios without leaders. But evacuation with too many leaders might not be efficient as it should be [31].

An evacuation system based on soft computing technique has been proposed to guide people to safest location during fire explosion. Artificial Neural Network (ANN) was used to determine the evacuation plan for crowd safety. Different scenarios (around 100) were used to train ANN and then trained network was tested on multiple floor plans to prove the reliability of proposed method. There were different smoke sensors used to indicate the direction to people towards their exits [32].

A novel approach, named, Crowd-separated Allocation of Routes, Exits and Shelters (CARES) has been proposed to avoid anomalies during crowd evacuation at Makkah. Another approach known as intelligent shelter allotment (ISA), plays an important role in emergency or disaster planning. But some of the issues, such as shelter allotment and transportation-network bottleneck, made it challenging. Authors have compared CARES with ISA and proved that CAERS outperformed the ISA. CARES addresses the limitations of ISA and provides faster evacuation and also eliminates transportation choke-point issues [33].

Seungho, Young and Judy [34] proposed evacuation model using Belief-Desire-Intention (BDI) architecture. The extended BDI framework was demonstrated using agent-based modeling and simulation. The approach used in the research was to represent the characteristics of the human decision-planning and decision-making process under selected evacuation scenarios. The developed simulation enables us to observe crowd behaviors under evacuation scenarios.

3.3 Crowd Model Verification and Validation

A list of seventeen test cases were presented for verification and validation of crowd evacuation models. Authors have performed different experiments for building fire evacuation and presented generic set of test cases which are useful for any simulation model. These recommended test cases help modeler to validate simulation models to make it bug free and also ensure its applicability in real world.

Different methods were presented to validate the simulation models. The main focus of this work was on crowd circumambulation in Makkah and Jamaraat bridges. Results obtained from simulation model was compared with empirical data to validate the model. In order to acquire the real-world data different footages were taken of Mataf and Jamaraat bridges. Later on, simulation results were compared with the results obtained from footages to validate the models.

3.4 Our Position in the State of Art

Most of the simulation models focus on the pedestrians movements in large areas but they lack the ability to incorporate areas according to real dimensions. Few researchers have modeled large-scale pedestrians in different areas i.e. Tawaf movement but these models do not take into account continuous space.

In this work, we try to fill this gap by designing complex spatial environment of Masjid-Al-Haram according to the real dimensions which incorporates large-scale pedestrians in continuous space. Our work provides generic solution to handle emergency situations for crowd safety. Moreover, our proposed solution incorporates different optimization strategies in order to obtain the optimal solution for crowd evacuation. Optimization is performed by running the simulation in the loop with varying parameters for assessment of algorithms.

Chapter 4

Methodology

In this chapter we propose an agent-based crowd simulation and analysis framework comprising of different evacuation strategies for crowd management and safety.

Our proposed framework is composed of three layers as shown in Figure 9.

1. Simulation Layer
2. Interface Layer
3. Optimization and Analysis Layer

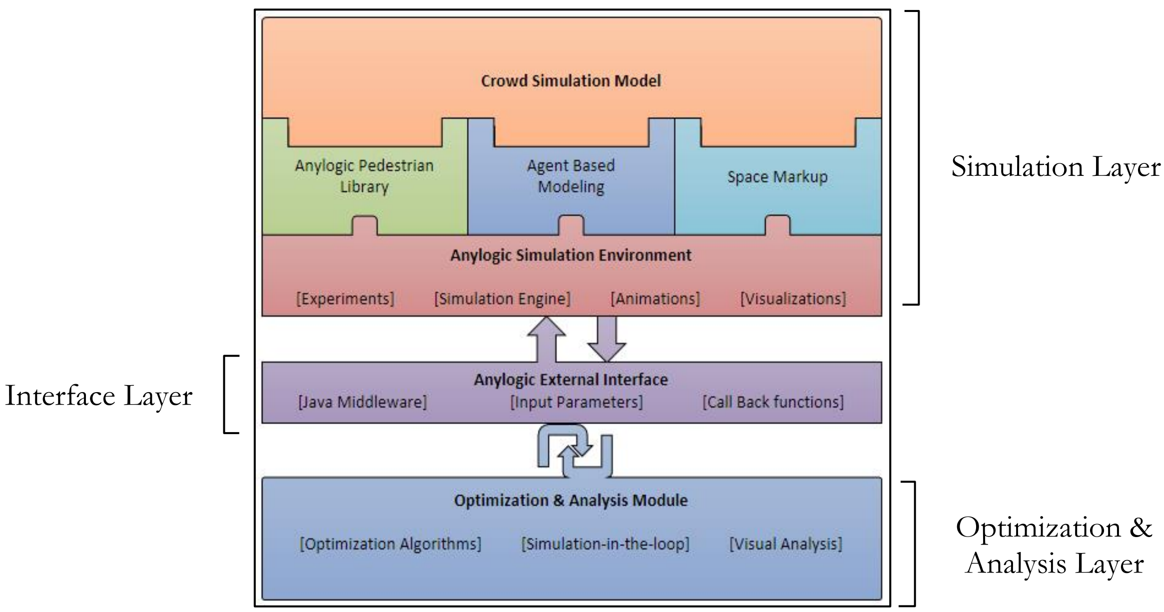
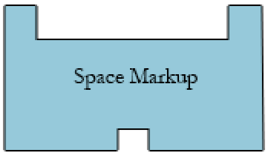


Figure 9: Crowd Simulation and Analysis Framework

4.1 Simulation Layer

In this layer we use Anylogic simulation software for development of our proposed framework. Anylogic provides different libraries such as Pedestrian, Agent-based and Space markup to build complex simulation environments. The main aim of this layer is to build a crowd simulation model on top of the above-mentioned libraries.

4.1.1 Space Markup Library



Different primitive elements of Space Markup Library such as, walls, polylines and pathways can help to make rich floor plan for pedestrian models. Table 2 describes the space markup elements used to design the simulation environment [35].

Table 2: Space Markup Elements

	Walls	Used to draw walls and obstacles in pedestrian models. Walls can be drawn in rectangle, circular or any shape. These are the objects which pedestrians can't cross during their movements.
	Target Line	Used to define source and destination of pedestrians in simulated area. Target lines can also be used to represent the entrance or exit gates of any building or room.
	Area	Area represents a floor or courtyard. It is a place where pedestrians can actually appear, walk or wait. Area can be defined in arbitrary shapes either rectangular, circular or any polygonal shape.
	Polyline	Polylines can be used to draw representation of models in 2D and 3D space.
	Attractor	Attractors are the elements used to specify the waiting locations of individuals in an area. It can be used to simulate waiting halls or class rooms. Attractors can be drawn using polylines and points.
	Scale	Scale is used to define suitable ratio for modeling purposes. Scale is a markup element which allows us to design the objects according to the real ratio.

Anylogic also facilitates to import AutoCAD drawings and Geographic Information System (GIS) shape files to build more flexible models in physical space. Anylogic AutoCAD feature enables us to import different drawings using .DXF file format. CAD file contains different layers which are easily customizable with different coloring schemes [36]. These types of designs are much smaller in size and are also rendered robustly during run time.

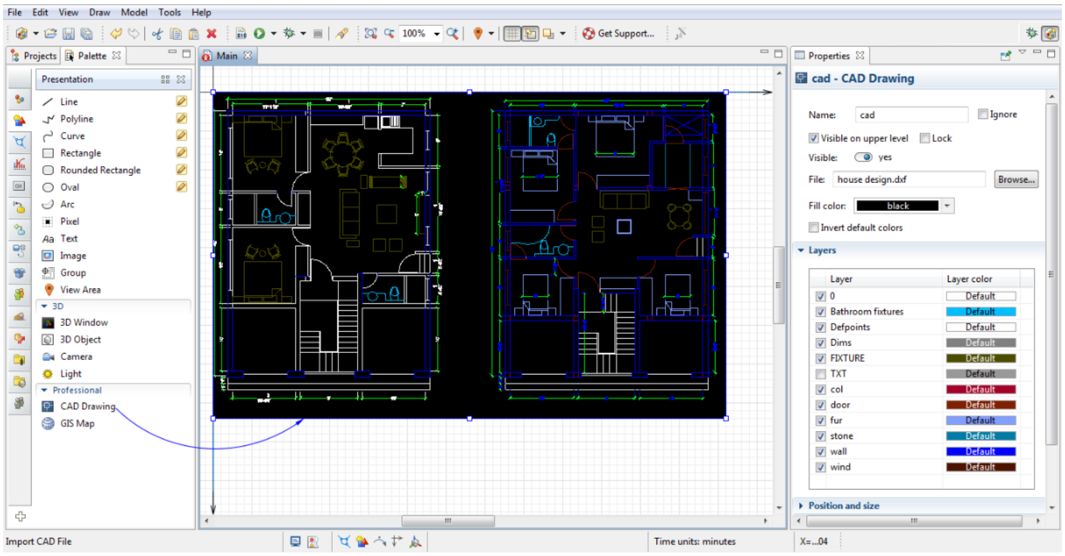
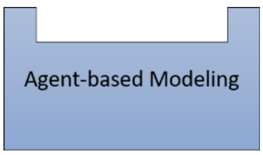


Figure 10: Loading AutoCAD drawing in Anylogic environment

This tool enables us to design and import rich layouts of buildings or different floor plans according to the real-world scale. CAD drawings are suitable during the modeling of pedestrian dynamics in complex environments, such as, shopping malls, subway area and hospitals. Once the CAD drawing is imported, obstacles, walking areas, source and destinations can be traced using space markup elements for pedestrian models.

4.1.2 Agent-based Modeling



Modeler can use agent-based modeling approach to simulate the pedestrians in the spatial environment. Agent-based modeling is an appropriate paradigm to simulate agents and their interactions by defining individual’s characteristics or attributes (age, name and weight) [37].

4.1.2.1 Agent Design

Anylogic provides multiple ways to create agents in spatial environment. Modeler can simulate agents and their behavior either using default agents or by creating new type of agents. New types of agents with custom physical and behavioral structure can be modeled through AnyLogic's agent-based library. Agent's physical structure includes attributes such as: age, gender, height, weight etc. whereas behavior includes states such as: normal, injured, and critical. Anylogic provides several probability distributions to initialize a population of agents and randomly generates initial values of these attributes according to the selected probability distribution. The behavior of agents is controlled using state charts with time triggered or event triggered transitions.

4.1.2.2 Agents Physical Behavior

As explained earlier agent's physical behavior can be modeled using agent-based library. It provides the ability to characterize the behavioral states of agents in the form of state charts. State chart is a generalized approach to depict the human behavior during external events (stampede and incidents). Agents' different physical behavior such as: normal, injured, unconscious, and deceased can be modeled easily in the form of states and transitions.

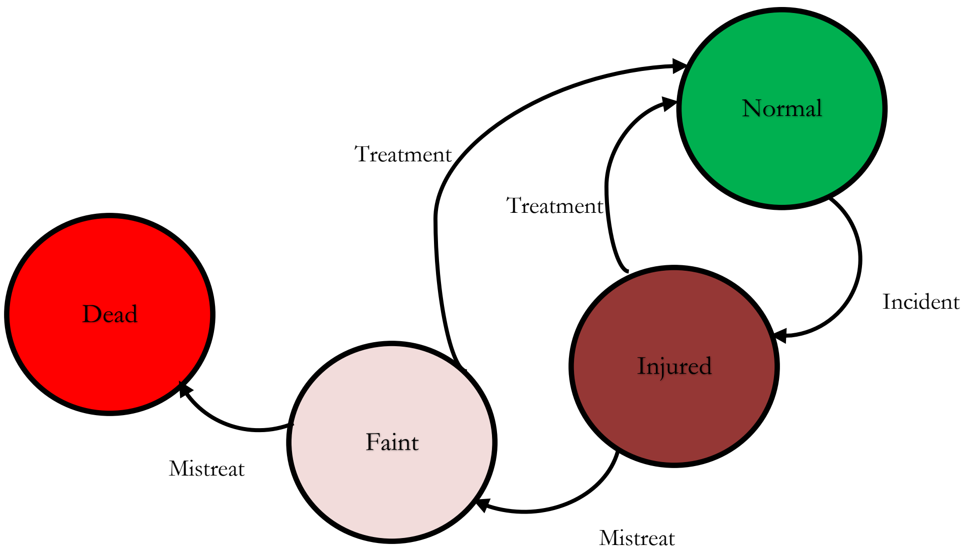


Figure 11: State Chart of Agent's Behavior during Incident

Figure 11 is describing the agent's behavior in the form of different states when incident occurs. Initially, agent is at normal state and later on when the incident occurs, the state changes in accordance with the severity level of situation. Some people get injured and some become unconscious whereas few people are expired. People in injured and unconscious state can get normal again with some proper assistance.

4.1.2.3 Agents Relationships

Agent-based library also provides the mechanism to define agent's communication and their relationships with each other. The relationship of agents can be implemented using Agent Network API.

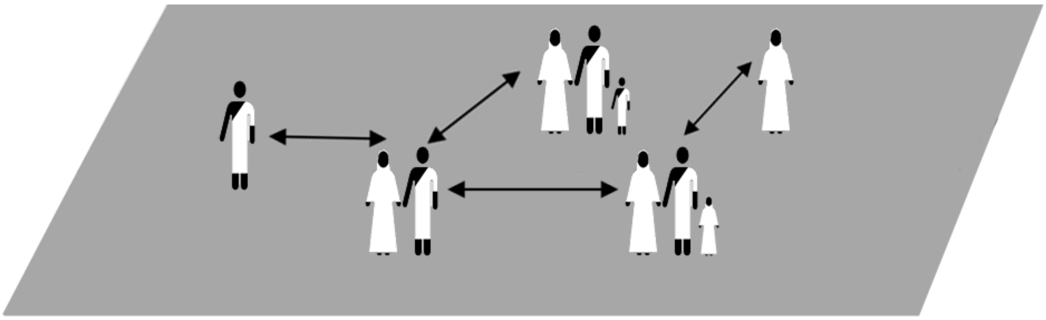
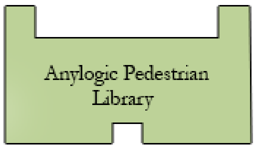


Figure 12: Agents relationship and their communication

Figure 13: Agents Relationship in terms of Nationalities

The relationships can be defined in the form of child, parents, friends and families. Agents living in same environment can communicate with each other through message passing. Anylogic provides built-in functions to send and receive messages from other agents. As shown in Figure 13, agents can also be related to each other in terms of nationality. While creating large number of agents with their complex behavior, modelers need to consider the computational power required to execute the models. A population of more than thousand individuals is not easily executable on desktop PC therefore a cloud computing platform is required.

4.1.3 Anylogic Pedestrian Library (APL)



AnyLogic Pedestrian Library is used to simulate pedestrians flow and their behavior in a physical environment [37]. APL provides friendly interface for robust development of pedestrian models in the form of flowcharts. It permits making pedestrian buildings models e.g. subway stations, concert halls and streets. Pedestrian models allow us to collect statistics on pedestrian density in different areas, measuring safety precautions and time estimations on services and queues.

Pedestrians' models consist of two parts (i) environment (ii) behavior. In these models, environments can be designed using space markup elements such as walls, obstacles, attractors etc. and pedestrians' behavior can be modeled using basic building blocks [38] provided by APL as described in Table 4. Crowd modeled using APL, moves in continuous space and reflects the reality i.e. reacting on different obstacles, walls and avoid collision with other pedestrians. These models can also assist in analyzing throughput capacity and in optimizing time schedules. Moreover, in case of disaster or terroristic attack, evacuation analysis and security plans can be made for crowd safety and risk evaluation.

Table 3: Pedestrian Library Settings Blocks

	PedConfiguration	Used to configure important parameters for pedestrian objects.
	PedGround	It is used to draw the floor where pedestrians can walk and also used to define walls and obstacles which pedestrians can't cross during walking.

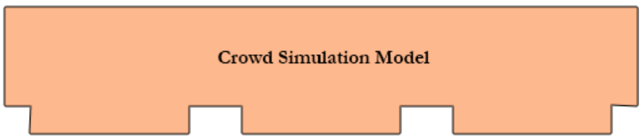
	PedArea	Represents walking area for pedestrians and is also used to modify their walking speed.
	PedAttractors	It Allows to control pedestrians' location in an area. Attractors define the waiting points where pedestrians will go and wait inside the area.

Table 4: Pedestrian Library Blocks

	PedSource	It is the starting point of pedestrian model and is used to generate agents with specified rate. Pedestrians are generated either from target line or are randomly populated in the space markup area. Modeler can use built-in class type *Ped* or custom defined agents to create the pedestrians. There are numerous ways to create agents such as: arrival rates, manual call inject function and inter arrival time. PedSource also facilitates to specify the walking speed, diameter and animation shapes of every pedestrian.
	PedSink	It is the final block of pedestrian model and is used to dispose the pedestrians. When pedestrians enter the PedSink they vanish from the simulation environment.
	PedGoto	PedGoto is used to direct the crowd towards specified location which can be defined using area, target line or attractor. Using this block pedestrians can be directed to follow the defined route i.e. circular path.
	PedWait	It causes pedestrians to wait for a specified time at a specified location. The waiting time of pedestrians can be cancelled by calling *free ()* function or when delay time is expired. Pedestrians waiting locations can be defined with or without attractors.

	PedSelectOutput	It is used to define the conditional statements in the model. It routes the pedestrians to one of several output ports according to the specified conditions or probabilities. PedSelectOutput work flow based on probabilities: pedestrians will exit through the ports based on the probabilities. If there are total 5 ports and we want to direct the equal number of pedestrians through each port, then assigned probability of each port will be 0.2. If we intend to increase the flow on one port as compared to that of others then we can increase its probability accordingly but net probability should be equal to 1. PedSelectOutput conditional work follow: initially, first condition is checked if it is true, pedestrians leave through first output port. Otherwise, next condition is evaluated and pedestrians will exit through the port 2 if condition is true. Same criteria will be followed for the rest of the ports.

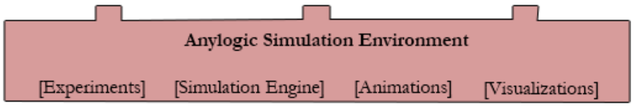
4.1.4 Crowd Simulation Model



Crowd simulation model has been developed using the following steps:

1. Design the external courtyard of Great Mosque of Makkah using AutoCAD in .DXF format.
2. Import the .DXF file in Anylogic environment using CAD feature.
3. Trace the obstacles and entrance/exit gates using space markup elements.
4. Use APL blocks to generate pedestrians in the courtyard.

4.2 AnyLogic Simulation Environment



4.2.1 Simulation Engine

Simulation engine controls the execution of model in a variety of ways. It provides different modes to execute the model either in virtual or real mode. Real-time mode is used when the model's presentation is required to appear as in real life whereas virtual model is used to make the model execution as fast as possible without mapping the model time to real time. This mode is useful when we need to execute our model for a long period of time. Simulation engine also defines the stopping criteria of model execution. Its different modes such as: start time, stop time, start date and stop date help modeler to analyze the model in multiple ways [39].

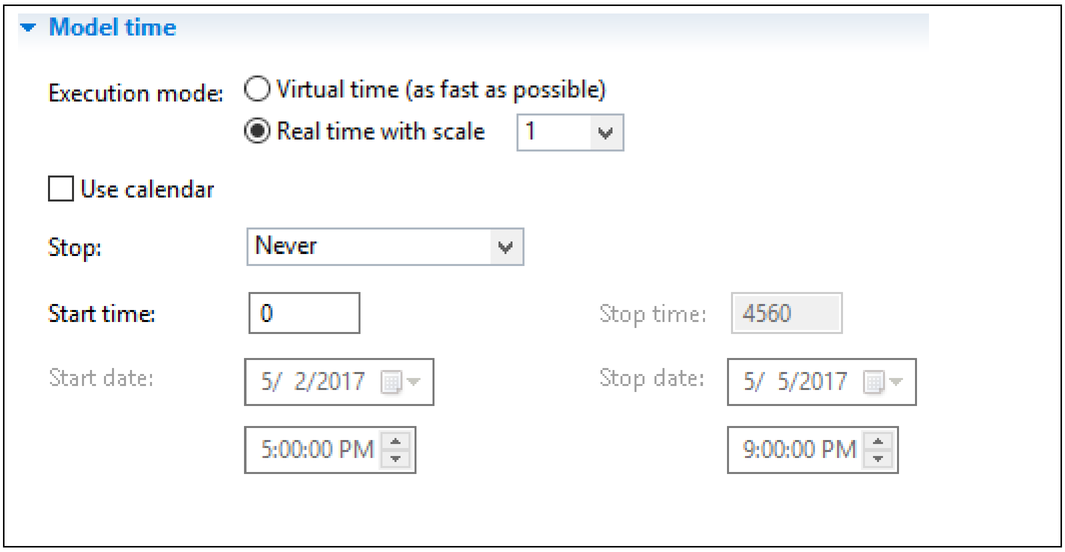


Figure 14: AnyLogic's Simulation Engine

4.2.2 Experiments

Anylogic provides several types of experiments for different simulation experiments [40].

Table 5: AnyLogic Experiment List

	Simulation	When a new model is created, default experiment named Simulation is always created automatically. This experiment is used to animate and debug the model. It provides different

		parameters to calibrate model settings.
	Parameter Variation	Anylogic provides Parameter Variation experiment to execute the model with varying parameters at run time. This experiment allows us to execute the model automatically for different parameters and compare them during execution time rather than executing them separately and storing the results to make comparison. Parameter Variation also facilitates analysis of effect of different parameters on the model behavior. Output of multiple executions can be displayed on single graph to analyze the model behavior against each parameter.
	Optimization	Optimization experiment is used to run the simulation model in loop to find the best possible solution by varying parameters.
	Monte Carlo	This experiment allows us to run the simulation a number of times to get the different results at each run and display results using histogram.
	Compare Runs	This experiment allows us to execute the model for different parameters and compare the results during execution time.
	Sensitivity Analysis	The experiment runs the model multiple times by varying one of the parameters and shows how the simulation output depends on it.
	Calibration	This experiment is used when we have multiple parameters and want to tune them to match the historical pattern. Calibration helps to reduce the variation between simulation results and historic data.
	Custom Experiment	Custom experiment is entirely managed by modelers and doesn't provide graphical interface to animate the model. It provides option to write the Java code in Anylogic environment and also allows use of different functions such as *start()*, *stop()* and *run()* etc. to handle the model execution.

4.2.3 Animations

AnyLogic enables users to create 2D and 3D animations of their models. As AnyLogic provides the space markup elements to construct the 2D design similarly these elements can be used for 3D animations. 3D animation is the most realistic and natural way of visualizing the simulation process [41] which can be enabled by dragging the 3D window  from Anylogic palette on the graphical editor. Other than space markup elements, external 3D objects can also be imported to create 3D animations. For example, some complicated objects such as house shapes, people and airplanes, etc. are usually not drawn in AnyLogic, but these shapes can be imported from external sources using the special 3D object  element with X3D and VRML formats [42].

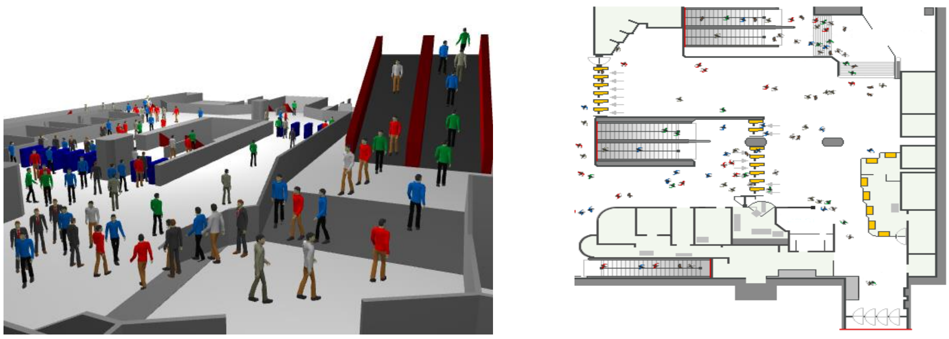
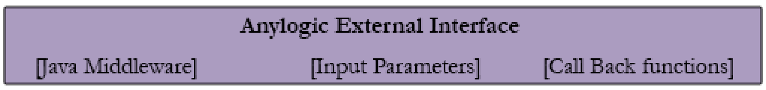


Figure 15: 3D and 2D view of Anylogic Model (Figure courtesy: Anylogic)

4.2.4 Visualizations

AnyLogic includes set of charts to plot the simulation output data at run time. These components can be edited graphically in AnyLogic editor, linked to parameters, variables and the data analysis objects (data sets, statistics, histograms), and displayed at runtime as a part of the model UI [43].

4.3 Interface Layer



This layer enables integration of the simulation layer with external programs. Upon the completion of simulation layer, it can be integrated with different optimization algorithms through interface layer. During the model development, user exposes certain simulation parameters which can be accessed through interface layer and are adjustable according to each algorithm.

4.3.1 Java Middleware

A Java based middleware has been developed that interacts with simulation layer and configures the parameters to run the model. Later on, this layer interacts with optimization and analysis layer which implements the different optimization algorithms.

4.3.2 Input Parameters

Few model parameters are exposed as public to control the simulation from external environment.

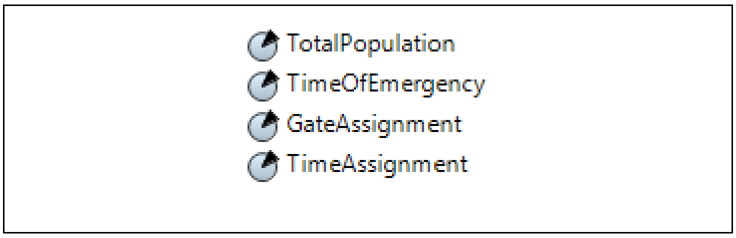


Figure 16: Simulation Model Input Parameters

These parameters are used in Interface layer and adjusted at run time during the optimization process. These parameters will be assigned varying values each time when simulation will run in the loop.

4.3.3 Callback Function

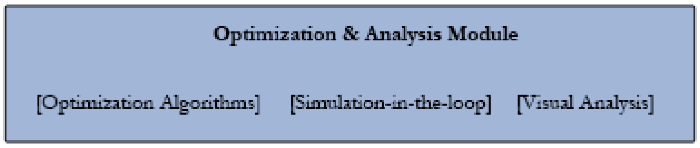
Simulation layer also contains the callback interface which defines the *onFinish()* function and is actually implemented in interface layer. This layer defines the body of *onFinish()* function, which is used to access model variables and objects at any particular event or through out the simulation lifetime. Once the simulation is finished, callback function is called to retrieve the simulation results. Depending on the need, user can create multiple callback functions and call them from the Interface layer. Structure of Callback interface is given below.

```
public interface Callback
{
        void onFinish(Main root);
}
```

The *OnFinish* function is actually implemented in the interface layer which is called after the completion of each simulation run and *root* parameter is used to gain access of the total evacuation time when all the pedestrians exit from the courtyard.

```
simulation.callback = new Callback() {
    @Override
    public void onFinish(Main root) {
      System.out.println(root.EvacuationTime);
    }
};
```

4.4 Optimization and Analysis Layer



This layer is used to implement and integrate optimization algorithms. This layer interacts with interface layer which sets the simulation parameters and runs the model to obtain the desired results. It provides such flexibility that user can implement different optimization algorithms such as: Simulated annealing, Hill climbing and Harmony search etc. After the implementation of optimization approaches, this layer communicates with interface layer to run the simulation model.

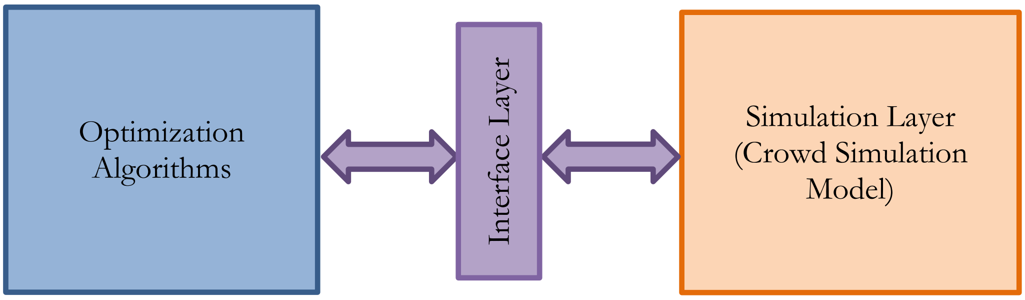


Figure 17: Integrating Optimization Algorithms with Simulation Model

As this layer is used for optimization purpose it runs the simulation in the loop without graphical interface and obtains the results through callback function. Thus, using this layer we can execute our simulation model for each algorithm to obtain the optimized evacuation plan for crowd safety.

We have implemented Genetic Algorithm (GA) and Harmony Search (HS) in order to obtain the optimized evacuation plan for crowd safety. Figure 18 describes the optimization process of GA and shows how this layer will interact and run the simulation model with varying parameters. At first, initial population is generated and for each candidate solution simulation is executed which interacts with interface layer and sets the parameters of each candidate to run the simulation model.

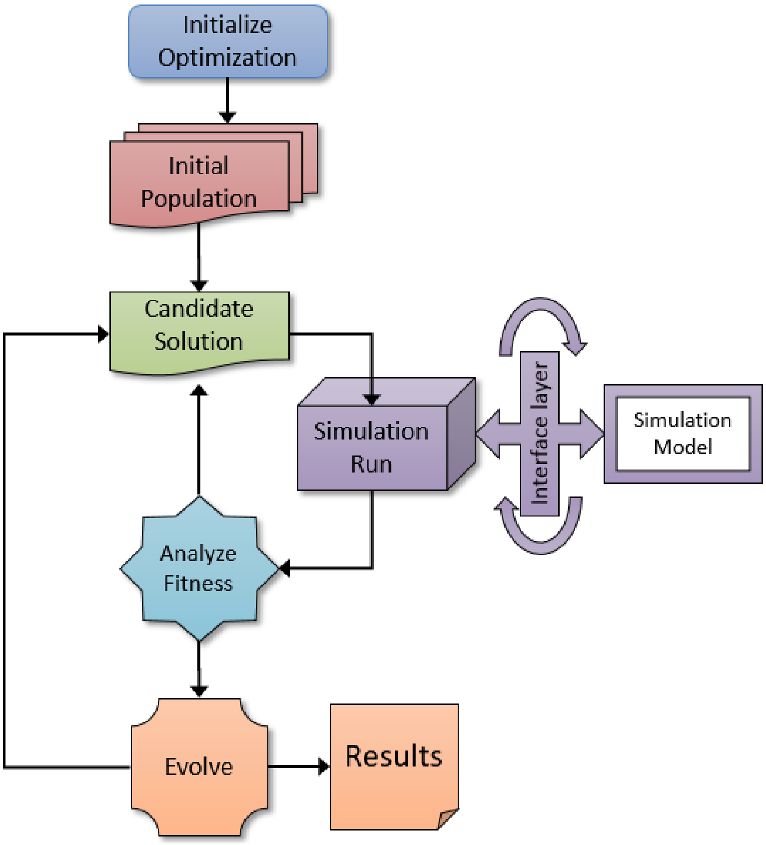


Figure 18: GA Flow

This task is done in the loop to generate each candidate solution and interface layer will create the new instance of simulation model for each candidate. After finishing each simulation run, callback function will be invoked and return the simulation result. In Analyze fitness module, the fitness of each candidate is compared with fittest solution and best candidate is further evolved according to GA mechanism. This process continues until the terminate condition is reached or best solution is found.

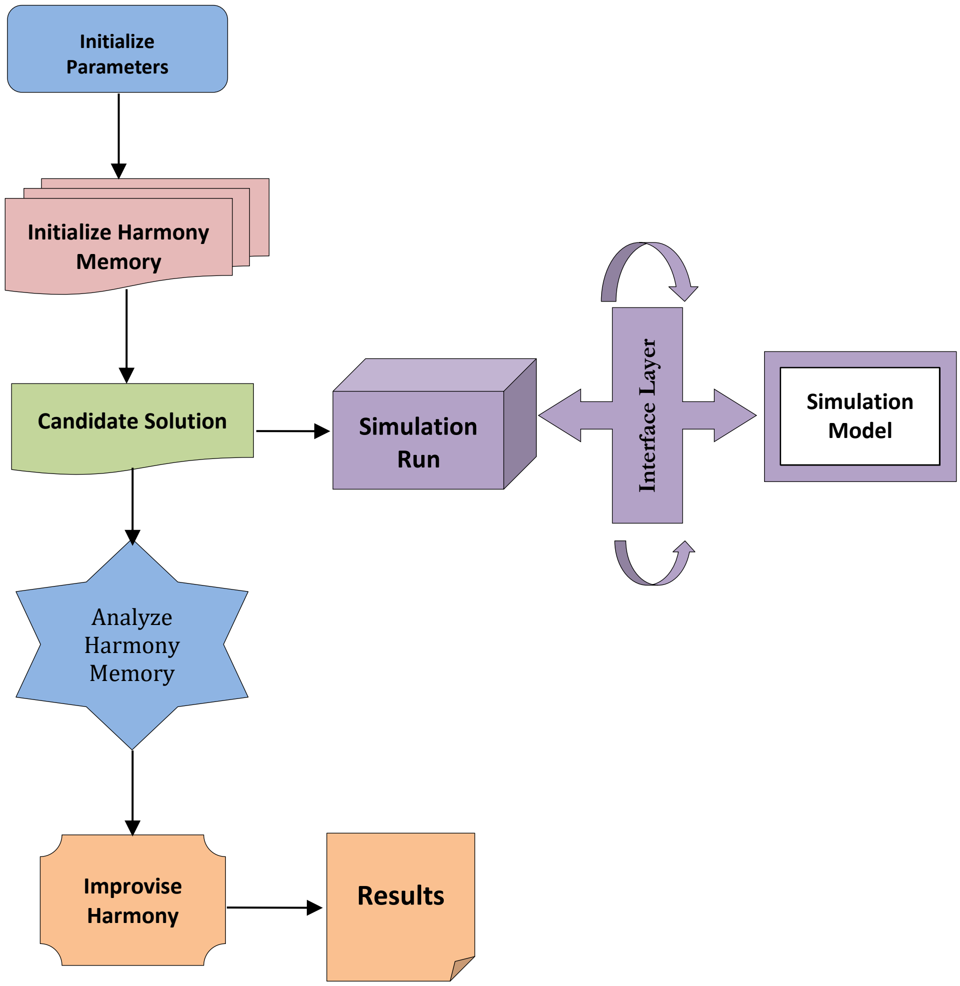


Figure 19: HS Flow

4.5 Emergency Evacuation Strategies

4.5.1 Random Crowd Evacuation

In this approach crowd randomly selects the exit paths to evacuate from the scene. When emergency occurs, pedestrians start vacating without any guidance. This strategy depicts the scenario when there is no supervisory control or assistance for people and they evacuate as they perceive the situation. The selection of exit gates is done on the basis of probability and equal probabilities has been assigned to all gates.

Table 6: Random Crowd Evacuation Algorithm

```
Algorithm I: Random Crowd Evacuation
Input: Total Population, Time of Emergency
Output: Evacuation Time

Initialize Population ⊳ using Ped Source
    While(time<delay)
        ⊳ current time is less than ped Wait delay
     Population Waits in the area ⊳ Using PedWait
    If Time of Emergency event has occurred then
     Start Time ← Current Time
     Cancel Wait
     ⊳ Causes Pedestrians to Start Evacuating
     for each Ped ∊ Population do
         Gate ← Uniform Probability (1 to 12)
         Goto ⊳ Using PedGoto
         Remove Ped from population ⊳ Using PedSink
     end for
     Evacuation Time = Current Time – Start Time
    end If
  end While

Return Evacuation Time
```

Table 6 explains the working of random crowd evacuation algorithm. First the *PedSource* has been used to generate the population in the courtyard and keep them waiting in the area using *PedWait* till the current time is less than the mentioned delay. After the occurrence of emergency event, pedestrians wait is cancelled which causes the pedestrians to evacuate and start time has been noted. Start time is the time when pedestrians actually start evacuating from the scene. *PedSelectOutPut* is used to select the exit gate for each pedestrian. This approach is completely random and

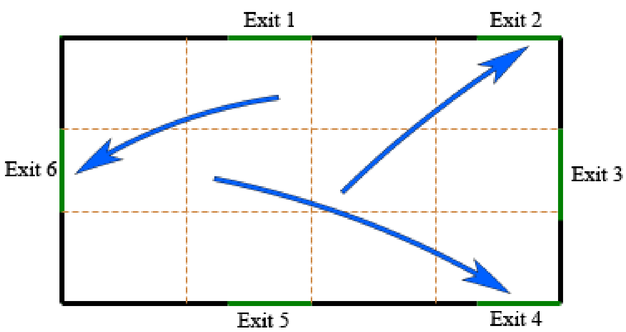


Figure 20: Random Crowd Evacuation Example

pedestrians might not choose their nearest exits. Gate assignments are done on the basis of probabilities and equal probabilities has been allocated to each gate. Example in Figure 20 shows that there are total 6 gates and equal probability (1/6) is assigned to each gate. Then *PedGoto* has been used which represents each exit gate and directs the pedestrians to move towards their exits. Finally, *PedSink* is used to dispose the pedestrians when they reach their exits and in the last step total evacuation time has been calculated.

4.5.2 Random-Time

When people start evacuation, it is expected that jamming can occur at exit gates and will cause delayed evacuation of the crowd. In order to minimize overcrowding at exit gates, time concept has been introduced. Different time slots are allocated for each region and pedestrians of that region start evacuating according to the allocated time sequence. Thus, it will reduce the congestion at gates.

4.5.3 Shortest Regional Distance (SRD)

This approach decides evacuation strategy for pedestrians using their minimum distance from exit gates. At first, the whole region is partitioned into logical sub-regions and the crowd is divided into regions based on their {x, y} positions. Figure 21 shows an example of SRD evacuation where the whole area is divided into 12 sub-regions (R1, R2, R3… R12) and there are 6 exit gates (Exit1, Exit2…Exit6). Each region will be assigned closest exit gate to evacuate during stampede or disaster.

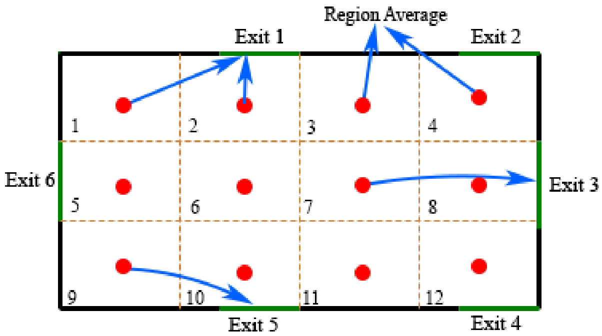


Figure 21: SRD Example

The pedestrians will follow the path and move to that exit which has been allocated to their particular region. Table 7 explains the working of SRD algorithm. Initial steps are same as explained in random crowd evacuation. When emergency event occurs, pedestrians wait has been cancelled which cause them to evacuate and start time has been recorded. Line 8 calls *GetRegion (Ped)* procedure which acquires the region of each pedestrian based on their {x, y} coordinates. When regions of whole population have been discovered the mean position of all pedestrians is calculated which is basically representing the average of a particular sub-region.

Table 7: Shortest Distance of the Region (SDR) Algorithm

```
Algorithm II: Shortest Regional Distance
Input: Population, Time of Emergency
Output: Evacuation Time

Initialize Population ▻ Using PedSource
    While(time<delay)
        ▻ Current time is less than ped Wait delay
        Population Waits in the area ▻ Using PedWait
        If Time of Emergency event has occurred then
            Start Time ← Current Time
            Cancel Wait
            ▻ Causes Pedestrians to start evacuating
            for each Ped i ∊ Population do
                Region ← Get Region (Ped)
                MeanXY ← Average (Ped.X, Ped.Y)
                Gate ← Nearest gate from MeanXY
                ▻ Gets the gate with shortest distance from the mean position to a region
                Goto Gate (Gate) ▻ Using PedGoto
                Remove Ped from Population Goto Gate (Gate) ▻ Using PedSink
            end for
            Evacuation Time = Current Time – Start Time
        end If
    end While
Return Evacuation Time
```

The average value of each region is calculated using Equation 1.

$$Region\ average = \frac{1}{n}\sum(\text{Ped.X}, \text{Ped.Y}) \text{….Equation 1}$$

After obtaining the mean position of sub-regions, their distance from each exit gate has been calculated and the gate having minimum distance is allocated to that region. Then *PedGoto* is used to direct the crowd to follow the exit gate which has been allocated to their region to evacuate from the courtyard. When all pedestrians reach their exit gates the total evacuation time is calculated.

4.5.4 Shortest Regional Distance (SRD)-Time

Similar to Random-Time, time slots are also allocated in this approach to minimize the overcrowding at exit gates. Different time slots are allocated for internal and external regions. People located in areas near to the exit gates will evacuate without any wait whereas pedestrians who are in the internal regions will have to wait until their time slot expired.

4.5.5 Genetic Algorithm

We propose the use of GA as an optimization approach to obtain the optimal evacuation solution in a given scenario.

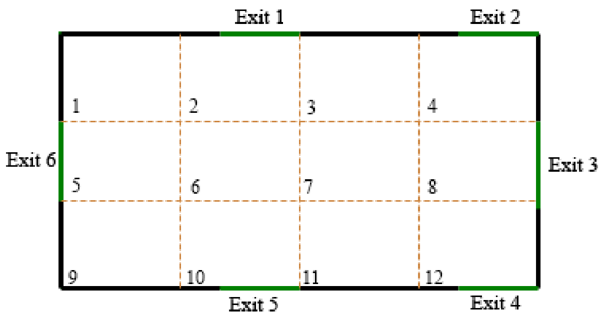


Figure 22: Evacuation Scenario

Following steps describe GA's working:

1. At first, the population of chromosomes is to be generated according to the suitable encoding. There are different encodings to represent the chromosomes such as: binary, real value, integer and permutation encoding. According to the problem's nature relevant encoding scheme can be selected to generate the chromosomes. In our scenario integer encoding has been used as shown in Figure 23. We need to find the optimized gate assignment to evacuate crowd in minimum time.

Gene

E1	**E3**	**E4**	**E5**	**E6**	**E2**	**E2**	**E2**	**E5**	**E3**	**E5**	**E6**	**E1**
R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**

Figure 23: GA'S Chromosome Structure

Figure 23 shows the structure of chromosome according to the given scenario. This chromosome represents the regions and assigned gates. In this chromosome all the regions are fixed and randomly gates have been assigned to each region.

2. In each generation, the population consisting of chromosomes is generated according to the specified size. This size can be varied according to the problem's nature and can be increased for complex problems to obtain the better solution.
3. Evaluate the fitness of each candidate according to the objective function i.e. minimizing the evacuation time.
4. Select the best chromosomes as parents and perform crossover to produce offspring and add them to the generation to keep the size preserved. These parents can be selected according to the different schemes such as tournament, rank and roulette wheel selection. An offspring is generated by randomly shifting the genes of both chromosomes. There are different crossover operators available to generate the offspring, for example, single-point, two-point, uniform and half uniform.
5. Figure 24 shows the parent chromosomes and their crossover operation to generate the offspring.
6. Mutate the offspring with specified probability to maintain the diversity from one generation to another generation. Mutation is performed by replacing the older genes with new genes and can be done using different operators.
7. Figure 25 explains the mutation procedure in which values of highlighted genes are randomly changed in offspring 2.
8. As generation evolves in each iteration few candidates don't further evolve and needs to be removed to maintain the diversity. Elitism is a process which decides the chromosomes to be kept and those to be discarded from the generation.

Parent 1

E1	E3	E4	E5	E6	E2	E2	E2	E5	E3	E5	E6	E1
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13

Parent 2

E6	E4	E3	E1	E2	E3	E5	E5	E4	E2	E3	E3	E6
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13

Offspring 1

E1	E3	E4	E5	E6	E2	E2	E5	E5	E2	E3	E3	E6
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13

Offspring 2

E6	E4	E3	E1	E2	E3	E5	E2	E5	E3	E5	E6	E1
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13

Figure 24: Crossover Operation and Generating two Offspring

E6	E4	E3	E4	E2	E3	E5	E3	E5	E3	E5	E6	E1
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13

Figure 25: Mutation of Offspring 2

Table 8: Genetic Algorithm

```
Algorithm III: Genetic Algorithm
Input: N ▷No of individuals in a generation
Output: Individual with Optimal Evacuation Time

Population ← Generate Population(N)
    for Generation i=1 to Maximum Generations
      Fittest Individual ← Best Individual in the population
      ▷ With minimum fitness value i.e. minimum evacuation time
      Population[0] ← Fittest Individual
      ▷ Pick the best from previous generation
        for j = 1 to N ▷ Evolve population
          {Group A} ← Tournament Selection
          {Group B} ← Tournament Selection
          ▷ Tournament selection randomly picks individuals from previous populations into groups
          New Individual ← Crossover
          ▷ Crossover combines a set of genes from a parent of Group A and a parent of Group B
          New Individual ← Mutate
          ▷ With a certain mutation probability the mutate function flips certain genes with new random values
          Population[j] ← Fittest Individual
        end for
    end for
Return Fittest Individual
```

```
Procedure: Generate Population
Input: N
Output: {Population}

for k = 1 to N
Individual ← Generate Chromosome(length)
      Fitness ← Calculate Individual Fitness
      ▷ Run simulation model for the individual with assigned gates for each region and returns Evacuation Time as fitness value
      Population[k] ← Individual
end for
Return {Population}
```

```
Procedure: Generate Chromosome
Input: Length
Output: Individual

for k = 1 to Length
Gene ← Random gate ← {Gates}
  ▷ Randomly selects a gate with uniform distribution
      Individual[k] ← Gene
end for
Return Individual
```

4.5.6 Harmony Search

Another optimization strategy, Harmony Search, has been implemented in Optimization and Analysis layer to obtain the optimal evacuation plan for the given scenario.

Working of HS is explained in following steps:

1. In the first step, optimization problem and main parameters of algorithm are defined which are used to optimize the results. These parameters include harmony memory size (HMS) which is also known as the number of solution vectors in harmony memory, harmony memory consideration rate (HMCR), pitch adjusting rate (PAR), number of improvisations (NI) and stopping criteria. Harmony memory (HM) is a memory where all solution vectors are stored. These solution vectors are similar to the chromosome representation in GA whereas HMCR and PAR are used to improve the solution vectors to find the optimal solution. More details of these parameters are given in next steps.
2. Figure 26 shows harmony memory matrix, defined as randomly generated solution vectors according to the HMS.

$$\begin{bmatrix} a_1^1 & a_2^1 & \cdots & a & a_N^1 \\ a_1^2 & a_2^2 & \cdots & a_{N-1}^2 & a_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1^{HMS-1} & a_2^{HMS-1} & \cdots & a_N^{HMS-1} & a_N^{HMS-1} \\ a_2^{HMS} & a_2^{HMS} & \cdots & a_{N-1}^{HMS} & a_N^{HMS} \end{bmatrix}$$

Figure 26: Harmony Memory Matrix (HMM)

According to the scenario as show in Figure 22, HMM can be defined as:

$$\begin{bmatrix} 6 & 5 & 2 & 3 & 5 & 1 \\ 2 & 4 & 5 & 3 & 1 & 1 \\ 1 & 3 & 2 & 5 & 6 & 4 \\ 3 & 3 & 5 & 6 & 3 & 1 \\ 2 & 3 & 4 & 5 & 6 & 1 \end{bmatrix}$$

Figure 27: HMM Example

3. After defining the HM matrix, each vector needs to be evaluated or calculate its fitness according to the objective function.
4. Once the fitness of each vector is determined HM should be improvised according to the parameters defined in step 1. A new harmony vector can be improvised in 3 ways: (i) memory consideration (ii) pitch adjustment (iii)

random selection. In memory consideration, value of decision variable for new vector is chosen from already defined HM range. The value of HMCR is varied from 0 to 1. HMCR defines ratio of selecting the value from HM and (1-HMCR) is the rate of randomly selecting value from possible range of values. Each value determined by memory consideration should be pitch-adjusted. This operation can be done using PAR parameter. For example, a PAR of 0.4 indicates that 40% of decision variables will be pitch adjusted while the remaining will not be pitch adjusted. Third way of improvising harmony is to select random values from defined range and add them to the newly generated vector.

5. After generating the new vector its fitness is calculated and compared with the worst harmony in HM. If newly generated vector is better, it is replaced in HM and worst harmony is excluded.
6. The stopping criteria needs to be checked after each iteration. If maximum number of improvisations has been reached, then it should be stopped otherwise steps 4 and 5 will be repeated.

Table 9: Harmony Search

```
Algorithm IV: Harmony Search
Input: N ⊳No of vectors/harmonies in a harmony memory
Output: Harmony with Optimal Evacuation Time

Initialize HS parameters HMS, HMCR, PAR, NVAR, NI
{Harmony Memory} ← Generate Harmony Memory (N)
    while Iteration < Maximum Iterations
      for i = 1 to NVAR do
        if Random Value < HMCR
          NewHarmony[i] ← memoryConsideration
          if Random Value < PAR
             NewHarmony[i] ← PitchAdjustmentRate
          end if
        end if
       Else
           NewHarmony[i] ← randomSelection
       end else
     Current Fitness ← calculate fitness(NewHarmony)
          ⊳ Run simulation model for each harmony with assigned gates for each region and
returns Evacuation Time as fitness value

          ⊳ update harmony memory using current fitness
    end while
          ⊳ get fittest harmony i.e. harmony having minimum value
 Return Fittest Harmony
```

```
Procedure: Generate Harmony Memory
Input: HMS, NVAR
Output: {Harmony Memory Matrix}

for k = 1 to HMS
  for j = 1 to NVAR
  HM[k][j] ← Get Gate(length)
      HM[k][NVAR] ← Calculate Fitness(HM[k])
          ⊳ Run simulation model for each harmony with assigned gates for each region
and returns Evacuation Time as fitness value
  end for
 Return { Harmony Memory Matrix }
```

```
Procedure: Get Gate
Input: Length
Output: Gate

Gene ← Random gate ← {Gates}
    ⊳ Randomly selects a gate with uniform distribution
 Return Gate
```

Chapter 5

Simulation and Results

In this section, we demonstrate the functionality of our proposed framework. We simulate the Hajj scenario to analyze different evacuation strtegies to alleviate disaster risk for pilgrisms in Great Mosque of Makkah.

5.1 Model Development (Simulation Layer)

Simulation layer presents the spatial model of Great Mosque of Makkah (Masjid-Al-Haram) which has been developed in two sections.

5.1.1 Spatial Environment

Firstly, the physical environment of Masjid-Al-Haram has been developed for crowd simulation using AutoCAD and this design has been imported in Anylogic using CAD feature. The mosque is divided into two parts, inner region and outer region. Inner region, where Khanna Kaba with Hateem and Maqam Ibrahim is situated. Kaba is located in the center and people circumambulate (Tawaf) around it seven times whereas the outer region which is the mosque building is the praying area which is connected with outer main gates. There are five main gates from where people can enter and exit. These are the gates people make their entrance to the courtyard to perform Tawaf.

Once the mosque design has been imported in Anylogic environment, different space markup elements are then used to build the walls, obstacles and enter/exit gates for creating spatial environment.

Figure 28 shows the top view of courtyard of Masjid-Al-Haram, where black border lines are denoting the walls and 12 entrances and exits are depicted in green color. The area has been designed in the courtyard where pedestrians will move around Khanna Kaba. Kaba and Hateem structure is designed using the space markup walls and represented with black color. Next to the Kaba is Maqam Ibrahim which is represented by a black solid point. These are all restricted areas which pedestrians can't cross during their movement in the courtyard.

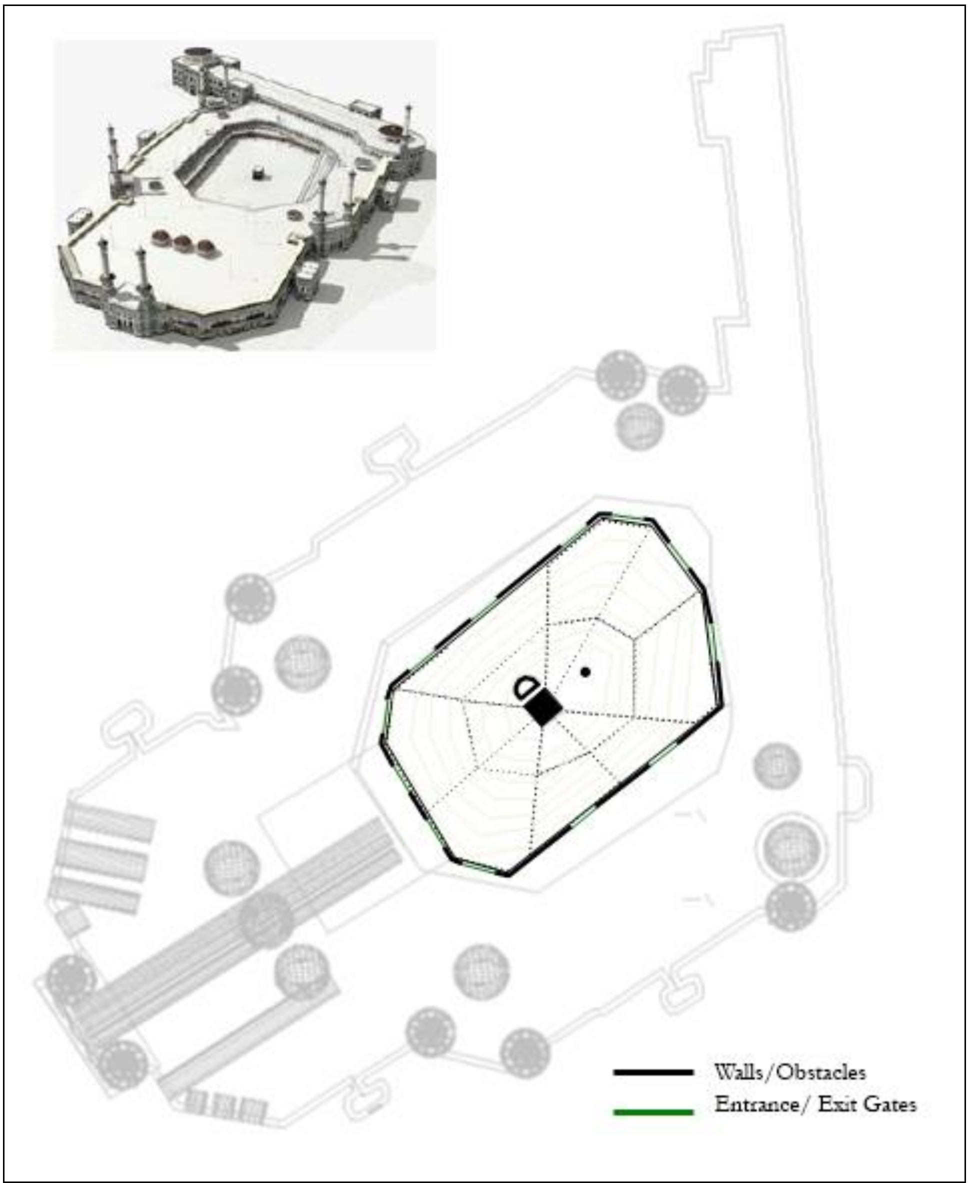


Figure 28: Anylogic Spatial Model of Great Mosque of Makkah (Top View)

5.1.2 Pedestrian Behavior

After the development of spatial environment of Masjid-Al-haram, modeler can use Anylogic Pedestrian Library (APL) to simulate the pilgrims in courtyard. At first, some environment control blocks, also known as pedestrian settings blocks, are used to make necessary configurations.

Initially *PedConfiguration* is used to tune some important parameters related to the objects of pedestrian library. Then *PedGround* is used to define the floor where pedestrians walk and it is also used to define the walls and obstacles which pedestrians can't cross during walking. After configuring the *PedGround*, *PedArea*

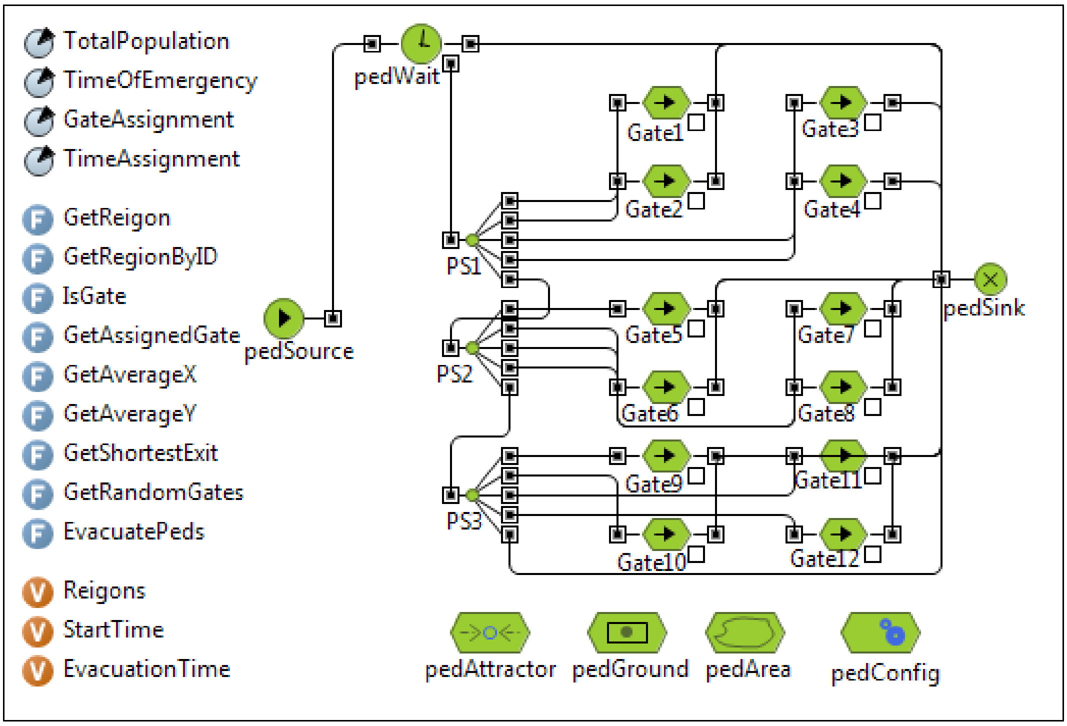


Figure 29: Pedestrians Flow Controls

and *PedAttractors* are used to control the walking speed and movement of pedestrians in the courtyard. Attractors can be lines or polylines which are used to control the flow of pedestrians, such as circular movement around Khanna Kaba. When all the configuration blocks have been defined, *PedSource*, which is the starting point of the simulation model, is used to generate the pedestrians in the courtyard. In order to make model more realistic we need to adjust some PedSource parameters, for example, pedestrians' arrival rate, walking speed and maximum arrivals. The arrival rate of crowd is defined as 5000 peds/second whereas initial walking speed of each pedestrian is 1.2 m/s (meters per second) and comfortable speed on which pedestrians will move in courtyard is defined by triangular distribution. Then *PedWait* is used to keep the pilgrims in the courtyard for large time (100 minutes) and they wait there till the occurrence of emergency event. Once the emergency event occurs, *PedWait.cancell* function causes the pedestrians to move to the *PedSelectOutPut* (PS1, PS2 and PS3). PedSelect is a gate selection block which directs the crowd towards their *PedGoto* block, representing the exit gates (Gate1, Gate2….Gate12), and finally the *PedSink* is called to dispose the crowd.

5.2 Simulation Experiment (Interface Layer)

In order to achieve interoperability with external environments, Anylogic Custom Experiment has been developed which consists of the following parts:

5.2.1 Interface Parameters

Some of the model parameters are exposed as public to control the simulation from external environment. These parameters are *TotalPopulation*, *TimeOfEmergency*, *GateAssignment* and *TimeAssignment*. They are used by Interface layer and adjusted at run time during the optimization process. These parameters will be assigned varying values each time the simulation is run in the loop.

5.2.2 Call Back Functions

Callback function which is already explained in the section 4.3 defines the *OnFinish* function which is actually implemented in the interface layer. This function is called after the completion of each simulation run and *root* parameter is used to gain access of the total evacuation time when all the pedestrians exit from the courtyard.

```
simulation.callback = new Callback() {
    @Override
    public void onFinish(Main root) {
      System.out.println(root.EvacuationTime);

    }
};
```

5.3 External Modules (Optimization and Analysis Layer)

Main purpose of this layer is to implement the optimization algorithm. It enables us to implement any optimization algorithm in order to get the optimized exit plan to evacuate the crowd in minimum time. This layer interacts with the interface layer to set the simulation parameters and finally executes the simulation model in the loop.

5.4 Emergency Evacuation Strategies

5.4.1 Random Crowd Evacuation

Initially, the population is generated in the courtyard using *PedSource*. Pedestrians are represented with green color and it is pre-evacuation scenario when everyone is moving in the area normally. When emergency occurs, the pedestrians color is changed to red, and everyone starts evacuating according to the random gate assignments.

- *TotalPopulation = 10,000*
- *TimeOfEmergency = 1.3*
- *GateAssignment = Random*
- *Average Speed of the Crowd = Triangular distribution min=1.0 m/s, mostly likely=1.4 m/s, max=2.0 m/s*
- *No. of Simulation runs = 20*

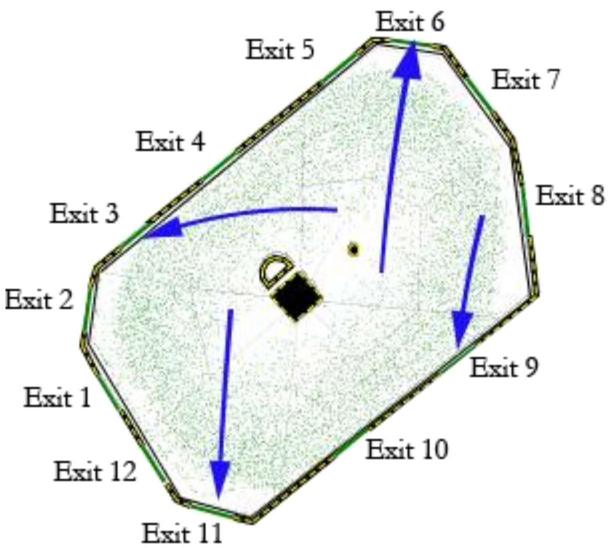


Figure 30: Random Crowd Evacuation

5.4.2 Random Time Evacuation (Random-T)

- *TotalPopulation = 10,000*
- *TimeOfEmergency = 1.3*
- *GateAssignment = Random*
- *Average Speed of the Crowd = Triangular distribution min=1.0 m/s, mostly likely=1.4 m/s, max=2.0 m/s*
- *No. of Simulation runs = 20*
- *Time Delay Assignment Probability = 0.5*

5.4.3 Shortest Regional Distance (SRD)

This approach evacuates crowd using the minimum distance from their exit gates. As mentioned in random crowd strategy, initially crowd is represented with green color. The whole Makkah courtyard is divided into 16 logical polygonal sub-regions (R1, R2, R3… R16) with 12 exit gates (Exit1, Exit2…Exit12). Each region will be assigned closest exit gate to evacuate during disaster. When emergency occurs, pedestrians color is changed to red and everyone starts evacuating from the nearest gate which has been allocated to their region.

- *TotalPopulation = 10,000*
- *TimeOfEmergency = 1.3*
- *GateAssignment = Nearest Exit*
- *Average Speed of the Crowd = Triangular distribution min=1.0 m/s, mostly likely=1.4 m/s, max=2.0 m/s*
- *No. of Simulation runs = 20*

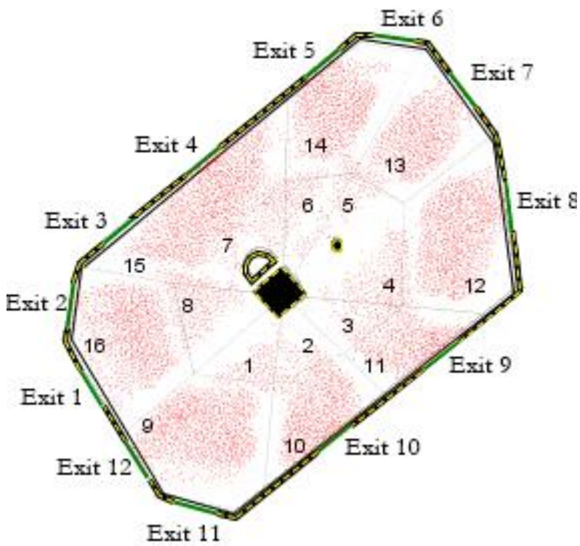


Figure 31: SRD Crowd Evacuation

5.4.4 Shortest Regional Distance Time (SRD-T)

- *TotalPopulation = 10,000*
- *TimeOfEmergency = 1.3*
- *GateAssignment = Nearest Exit*
- *Average Speed of the Crowd = Triangular distribution min=1.0 m/s, mostly likely=1.4 m/s, max=2.0 m/s*

- *No. of Simulation runs = 20*
- *Time Delay for all Outer Regions (R9-R16) = 0*
- *Time Delay for all Inner Regions (R1-R8) = 1.5*

5.4.5 Genetic Algorithm

- *Population = 10,000*
- *No. of Sub-regions = 16*
- *Time of Emergency = 1.3 minute*
- *Average Speed of the Crowd = Triangular distribution min=1.0 m/s, mostly likely=1.4 m/s, max=2.0 m/s*
- *Population N = 40;*
- *No of Generations = 50*
- *Cross Over Gene Selection = Uniform Distribution [0 to 6]*
- *Mutation Rate = 0.015*
- *Tournament Size = 5*
- *Elitism = true*

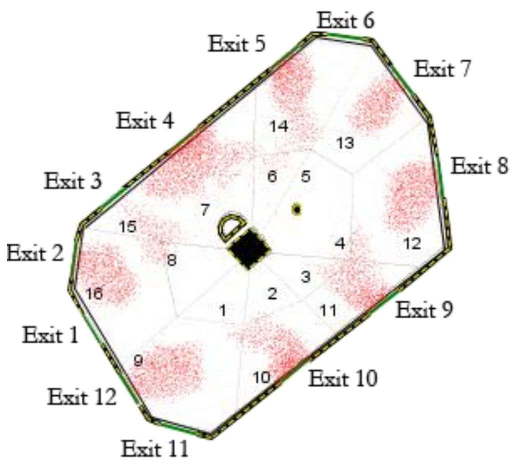


Figure 32: GA Crowd Evacuation

Optimal gate assignment determined using GA:

E3	**E10**	**E9**	**E4**	**E5**	**E6**	**E1**	**E3**	**E11**	**E10**	**E9**	**E7**	**E8**	**E5**	**E4**	**E2**
R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**	**R15**	**R16**

Figure 33: GA Gate Assignment of Makkah Courtyard

5.4.6 Harmony Search

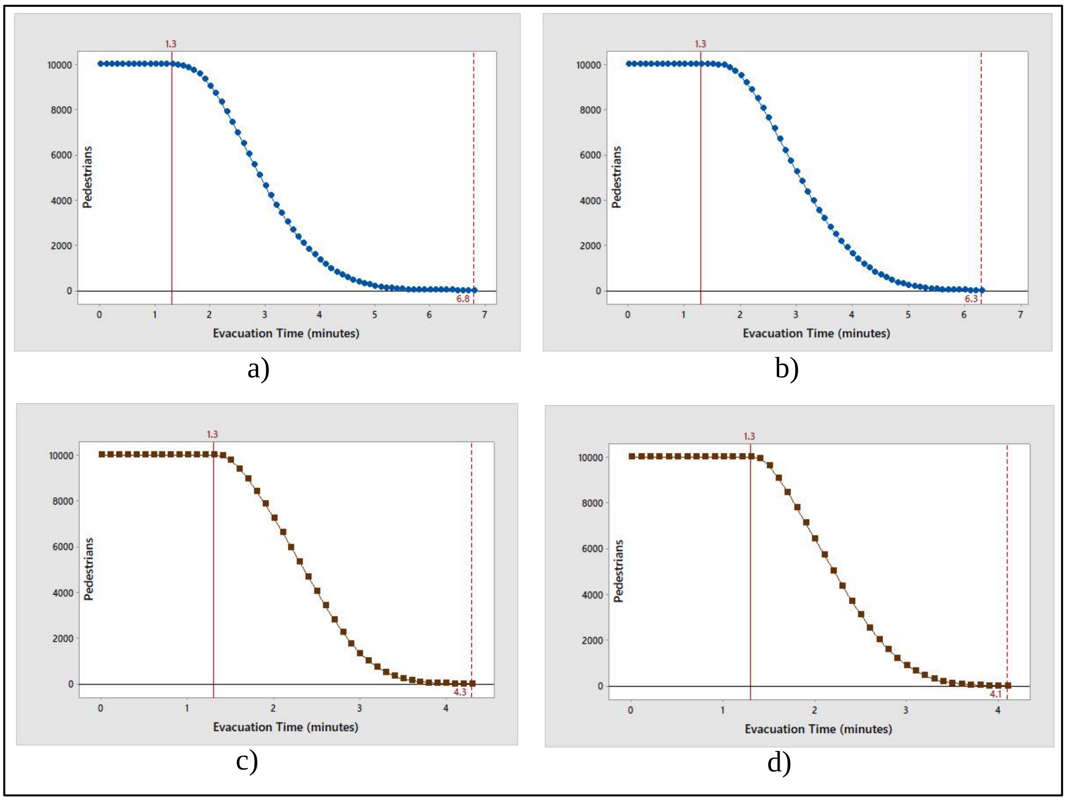
- *Population = 10,000*
- *No. of Sub-regions = 16*
- *Time of Emergency = 1.3 minute*
- *Average Speed of the Crowd = Triangular distribution min=1.0 m/s, mostly likely=1.4 m/s, max=2.0 m/s*
- *Harmony Memory Size (HMS) = 50;*
- *No of Iterations = 3000*

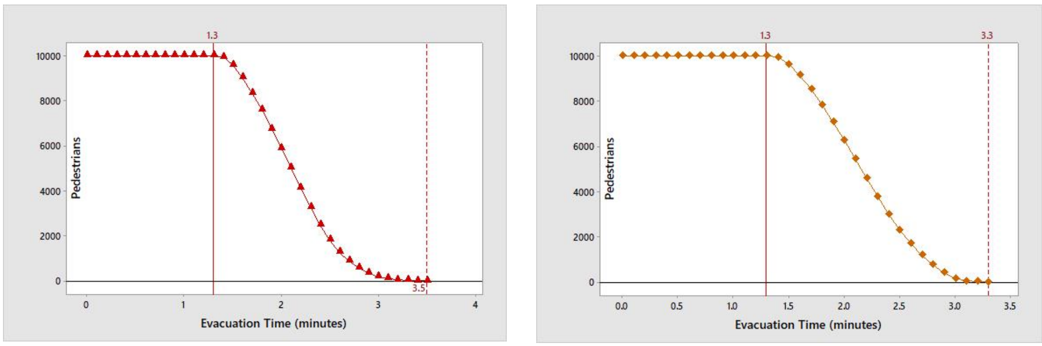
Optimal gate assignment determined using HS:

E3	E10	E11	E4	E9	E7	E1	E3	E11	E10	E9	E8	E8	E5	E4	E2
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16

Figure 34: HS Gate Assignment of Makkah Courtyard

5.4.7 Results and Discussion





e) f)

Figure 35: a) Random b) Random-T c) SRD d) SRD-T e) GA f) HS

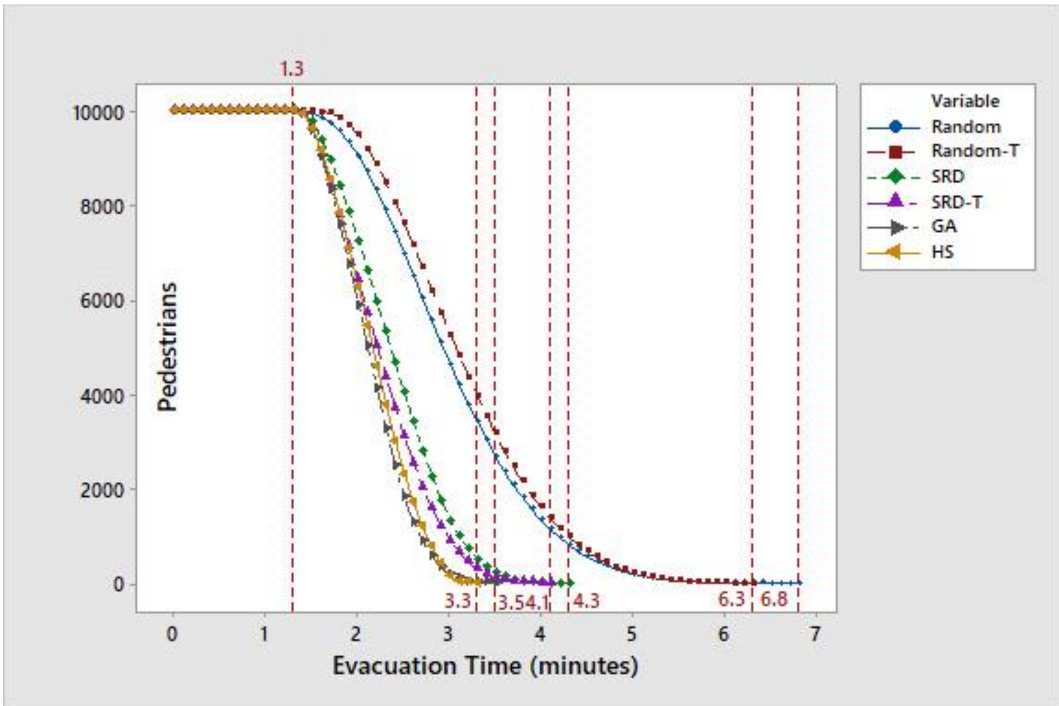


Figure 36: Evacuation Time plot

Graph shows the mean plots of four evacuation strategies and 6 scenarios. In all cases, we assume that the crowd has been uniformly distributed in the area with the average speed defined by triangular distribution. Different parameters have been initialized to run the simulation and changing the simulation parameters will produce variations in the results. Each strategy is executed 20 times with 10,000 population and average result of each strategy has been plotted. Red bold line represents the emergency which occurred at 1.3 minutes, indicating that pedestrians start evacuating after this time and emergency time is same for all strategies.

Random evacuation, which is also a worst-case scenario, people evacuated using this approach with the average of 6.8 minutes. Whereas Random-T performs slightly better than Random with the mean time 6.3 minutes because the time sequence helps to reduce the overcrowding at exit gates and thus reduces the evacuation time. SRD is much better than Random and Random-T with mean time 4.3 minutes as group of people (people of reach region) move towards their nearest exit. In SRD-T 10 seconds delay has been introduced for inner regions and it performed slightly better than SRD with mean time of 4.1 minutes. GA performed better than other approaches with mean time of 3.5 minutes and HS performed remarkably better than all other strategies with mean time 3.3 minutes because optimized gates have been assigned to each region.

Chapter 6

Model Evaluation

In this section, we explain functional and integration testing. We also discuss that how our model has been tested using these testing techniques.

6.1 Functional Testing

Functional testing is a testing which verifies that each function of a software or model is working according to the requirements. This type of testing lies in the branch of black box testing and doesn't concern about the source code of the application. Each function is given an appropriate input and results are verified by comparison with expected results. This testing can be done either manually or automatically [44].

6.1.1 Functional Testing of Model

In order to test our model, we have performed automated testing using Junit libraries. Each function is given appropriate input and comparison of its output with expected result is done to verify the functionality. Given below is the list of test cases to verify the functionality of the model.

Testing Functions	**Description**
GetRegion	Input: X, Y coordinates Output: Region no. This function is tested by passing x and y coordinates of pedestrian as input parameters and it will return region of the pedestrian based on the coordinates. Later, this output is compared with expected output to verify the result.

GetRegionByID	Input: ID Output: Region no. GetRegionByID takes pedestrian's ID as an input and returns the region in which that pedestrian resides. The retrieved region is compared with the expected output region to verify the working of function.
GetShortestExit	Input: Region no. Output: Nearest exit gate no. This function takes region no. as an input and returns the nearest exit gate among others.
IsGate	Input: Pedestrian ID, Gate no. Output: Boolean IsGate takes pedestrian ID and gate no. as input parameters and evaluates that this gate belongs to particular pedestrian or not.
GetAssignedGate	Input: Region no. Output: Region's gate GetAssignedGate takes region no. as an input parameter and returns gate number which has been allocated to the region. The gate number is compared with expected output to verify results.

Table 10: Testing of Model Functions

6.2 Integration Testing

Upon the completion of unit testing, multiple functions or modules are combined and tested as groups. The purpose of integration testing is to verify the functionality and reliability between the modules that are integrated. Integration testing helps developers to debug the modules such as, exposing problems with the interfaces among program components [44].

6.3 Integration Testing of Model

As our model is developed in three layers, so, integration testing will be performed by combining these layers to carry out the interface level testing. As mentioned above, simulation and optimization layers are communicating with each through interface layer with varying parameters and retrieving the results using callback function.

Chapter 7

Conclusion and Future Work

This chapter provides the discussion, conclusion and future work of the thesis.

7.1 Conclusion

In this research we have presented that mass gatherings and overcrowding can confront emergency situations either because of natural disasters or due to man-made chaos. We also discussed need of time is to provide such mechanism for public safety and security during large events. Crowd simulation is playing vital role for the development of crowd management and security plans. It helps to replicate the real-world scenario in virtual environment to perform different tests in order to get feasible solution (evaluating crowd evacuation strategies). We proposed an agent-based crowd simulation and analysis framework to evaluate and determine the optimal evacuation strategy for crowd safety. Proposed framework is composed of three layers (i) Simulation layer enable modelers to create crowd simulation model using Anylogic simulation software and using its different libraries (ii) Interface layers provides interoperability to integrate simulation model with external algorithms (iii) Optimization and Analysis layer enables to implement different optimization algorithms which can be integrated with simulation model through interface layer to obtain the optimal evacuation plan.

We have presented case study of Hajj scenario as a test bed for assessment of our proposed framework. We provided key steps to build the spatial environment of Makkah using AnyLogic's libraries such as APL, ABM and space markup. Proposed framework adopts Random, SRD and GA strategies for crowd evacuation. These approaches have been tested on Hajj scenario and performed different simulation runs with varying parameters to analyze and obtain the optimal evacuation plan for crowd safety. Later on, time sequence concept has been introduced to reduce the overcrowding at exit gates because, when evacuation starts everyone tries to reach the exit in minimal time and results in stampede or crowd crushes. Thus, after including delays in different regions of courtyard, timed strategies have performed better than without delays in inner region.

Precisely, our proposed framework can be deployed at PSS organizations to benefit and develop safety plans for mass gatherings.

7.2 Future Work

There are several future directions of proposed framework such as integrating different optimization algorithms with simulation model and comparing them with already existing models to obtain the optimal result. This work can also be extended to develop simulation model and evacuation plans for Jamaraat Bridge. Moreover, our meso crowd model is extendable to micro level in order to incorporate the attributes and physical states of each pedestrian which can provide detailed insights to observe human behavior during evacuation. We also aim to implement integration of real-world sensors and actuators for Real-world in the loop modes, where e.g., the GIS coordinates of a crowd is acquired for input and the actuators are used to manage the crowd flow for optimal evacuation.

References

[1] World Health Organization, "Communiable Disease Alert and Response for Mass Gatherings: Key Considerations," WHO press, Geneva, 2008.

[2] J. A. Al-Tawfiq and Z. A. Memish, "The Hajj: updated health hazards and current recommendations for 2012," *Eurosurveillance,* vol. 17, 2012.

[3] "Incidents during the Hajj," [Online]. Available: https://en.wikipedia.org/wiki/Incidents_during_the_Hajj. [Accessed June 2017].

[4] B. Zafar, "Analysis of the Mataf - Ramadan 1432 AH Technical report, Hajj Research Institute, Umm al-Qura University, Saudi Arabia," 2011.

[5] S. Curtis, S. J. Guy, B. Zafar and D. Manocha, "Virtual Tawaf: A case study in simulating the behavior of dense, heterogeneous crowds," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.

[6] M. Shamsul, S. M. Shahrizal and R. M. Hanifa, "A Review on Tawad Crowd Simulation: State-of-the-Art," *International Journal of Interactive Digital Media,* Vols. vol. 2, , no. no. 11, pp. pp. 1-6, 2014.

[7] S. P. Simonovi´c, in *Systems Approach to Management of Disasters Methods and Applications*, WILEY, pp. 31-41.

[8] D. Thalmann and S. R. Musse, Crowd Simulation, 2016.

[9] S. Zhou, D. Chen, W. Cai, L. Luo, M. Y. H. Low, F. Tian, V. S.-H. Tay and B. D. Hamilton, "Crowd modeling and simulation technologies," *ACM Transactions on Modeling and Computer Simulation (TOMACS),* vol. 20, no. 4, pp. 20:1--20:35, 2010.

[10] S. Chenney, "Flow Tiles," in *Symposium on Computer Animation*, 2004.

[11] D. Helbing and P. Molnar, "Social Force Model for Pedestrian Dynamics," *PHYSICAL REVIEW E,* vol. 51, pp. 4282-4286, 1995.

[12] "Using Pedestrian Dynamics," 2017. [Online]. Available: http://www.pedestrian-dynamics.com/pedestrian-dynamics/pedestrian-dynamics-features.html. [Accessed 2017].

[13] "Pedsim Demo Application," [Online]. Available: http://pedsim.silmaril.org/examples/. [Accessed 2018].

[14] "Anylogic Simulation Software," [Online]. Available: http://www.anylogic.com/. [Accessed January 2017].

[15] A. Borshchev, The big book of simulation modeling: multimethod modeling with AnyLogic, AnyLogic North America , 2013.

[16] D. Helbing and A. Johansson, "Pedestrian, Crowd and Evacuation Dynamics," 2010.

[17] D. E. Goldberg, Genetics Algorithms in Search Optimization & Machine Learning, 1989.

[18] M. Melanie, An introduction to genetic algorithms, MIT press,, 1998.

[19] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer methods in applied mechanics and engineering,* pp. 3902-3933, 2005.

[20] F. H. M. e. Aguiar, Crowd Simulation Applied to Emergency and Evacuation Situations, 2010.

[21] D. C. Duives, W. Daamen and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transportation Research Part C: Emerging Technologies,* vol. 37, pp. 193-209, 2013.

[22] Q. Sun, A Generic Approach to Modelling Individual Behaviours in Crowd Simulation, 2013.

[23] S. Wolfarm, Theory and Applications of Cellular Automata. World Scientific, 1986.

[24] N. Pelechano, J. M. Allbeck and N. I. Badler, "Virtual Crowds: Methods, Simulation, and Control. Synthesis Lectures on Computer Graphics and Animation,," 2008.

[25] J. Dijkstra, H. J. Timmermans and B. d. Vries, "Towards a Multi-Agent System for Visualising Simulated Behaviour within the Built Environment," in *Proceedings of Design and Decision Support Systems in Architecture and Urban Planning Conference: DDSS*, 2000, pp. 101-117.

[26] S. Sarmady, F. Haron and A. Z. H. Talib, "Multi-Agent Simulation of Circular Pedestrian Movements Using Cellular Automate," in *Second Asia International Conference on Modelling & Simulation*.

[27] S. Sarmady, F. Haron and A. Z.H.Talib, "Agent-Based Simulation Of Crowd At The Tawaf Area".

[28] A. Abdelghany, K. Abdelghany, H. Mahmassani and W. Alhalabi, "Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities," *European Journal of Operational Research,* vol. 237, no. 3, pp. 1105-1118, 2014.

[29] J. Zhong, W. Cai and L. Luo, "Crowd Evacuation Planning Using Cartersian Genetic Programming and Agent-Based Crowd Modeling," in *Winter Simulation Conference*, 2015.

[30] A. Ferscha and K. Zia, "LifeBelt: Silent Directional Guidance for Crowd Evacuation," in *International Symposium on Wearable Computers*, Linz, 2009.

[31] Q. JI and C. GAO, "Simulating Crowd Evacuation with a Leader-Follower Model," *IJCSES International Journal of Computer Sciences and Engineering Systems,*

vol. 1, pp. 249-252, 2007.

[32] K. V. Santhosh and P. Mohanty, "Soft Computation Technique based Fire Evacuation System," *Indian Journal of Science and Technology,* vol. 8, 2015.

[33] K. Yang, F. U. Rehman, H. Lahza, S. Basalamah, S. Shekhar, I. Ahmed and A. Ghafoor, "Intelligent Shelter Allotment for Emergency Evacuation Planning: A Case Study of Makkah," *IEEE Intelligent Systems,* vol. 30, no. 5, pp. 66-76, 2015.

[34] S. LEE, Y.-J. SON and J. JIN, "An Integrated Human Decision Making Model for Evacuation Scenarios under a BDI Framework," *ACM Transactions on Modeling and Computer Simulation,* vol. Volume 20, no. 4, October 2010.

[35] "Space Markup Palette," [Online]. Available: https://help.anylogic.com/index.jsp?topic=%2Fcom.anylogic.help%2Fhtml%2Fui%2FSpace+Markup+palette.html. [Accessed 2018].

[36] "CAD Drawing," [Online]. Available: https://help.anylogic.com/index.jsp?topic=/com.xj.anylogic.help/html/markup/Pedestrian_Markup.html. [Accessed 2017].

[37] "AnyLogic Pedestrian Simulation," [Online]. Available: https://en.wikibooks.org/wiki/Simulation_with_AnyLogic/Pedestrian_Simulation. [Accessed 2017].

[38] "Pedestrian Library Blocks," [Online]. Available: https://help.anylogic.com/index.jsp?topic=/com.xj.anylogic.help/html/_PL/reference/Pedestrian_Library_Objects.html. [Accessed 2017].

[39] "Simulation Experiment," [Online]. Available: https://help.anylogic.com/index.jsp?topic=%2Fcom.anylogic.help%2Fhtml%2Fexperiments%2FCreating+an+Experiment.html. [Accessed 2017].

[40] "Experiments," [Online]. Available: https://help.anylogic.com/index.jsp?topic=/com.xj.anylogic.help/html/experiments/About%20Experiments.html. [Accessed 2017].

[41] "Animating pedestrians," [Online]. Available: https://help.anylogic.com/index.jsp?topic=/com.xj.anylogic.help/html/_PL/reference/Animation.html. [Accessed 2017].

[42] "3D animation," [Online]. Available: https://help.anylogic.com/index.jsp?topic=/com.xj.anylogic.help/html/3d/3D.html. [Accessed 2017].

[43] "Charts," [Online]. Available: https://help.anylogic.com/index.jsp?topic=%2Fcom.anylogic.help%2Fhtml%2Fanalysis%2FCharts.html. [Accessed 2017].

[44] I. Sommerville, Software Engineering, 2009.

[45] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences,* vol. 99, no. suppl 3, p. 7280–7287, 2002.

[46] A. Borshchev and A. Filippov, "From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools," in *Proceedings of the 22nd international conference of the system dynamics society*, 2004.

[47] A. Borshchev, The big book of simulation modeling: multimethod modeling with AnyLogic 6, 2013.

[48] J. Banks, J. S. Carson, B. L. Nelson and D. L. Nicol, Discrete-Event System Simulation (5th Edition), Prentice Hall, 2009.