



Bat Optimization For Crowd Evacuation Simulation in Classroom

Amylia Binti Mohammad Hasly

Bachelor of Computer Science with Honors (Software Engineering)

2025

**BAT OPTIMIZATION FOR CROWD EVACUATION SIMULATION IN
CLASSROOM**

AMYLIA BINTI MOHAMMAD HASLY

This project is submitted in partial fulfilment of the requirements for the degree of
Bachelor of Software Engineering with Honours

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

2025

BAT OPTIMIZATION FOR CROWD EVACUATION SIMULATION IN CLASSROOM

AMYLIA BINTI MOHAMMAD HASLY

Projek ini merupakan salah satu keperluan untuk Ijazah Sarjana Muda Kejuruteraan
Perisian Dengan Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE **BAT OPTIMIZATION FOR CROWD EVACUATION
SIMULATION IN CLASSROOM**

ACADEMIC SESSION: 2025

AMYLIA BINTI MOHAMMAD HASLY
(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☐	CONFIDENTIAL	(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)
☐	RESTRICTED	(Contains restricted information as dictated by the body or organization where the research was conducted)
☒	UNRESTRICTED	



(AUTHOR'S SIGNATURE)

Validated by



(SUPERVISOR'S SIGNATURE)

Permanent Address

No.188 lot.668 Jalan ketitir 3a
Everbright estate 93250 Kuching
Sarawak

Date: 10 July 2025

Date: 9 July 2025

Note * Thesis refers to PhD, Master, and Bachelor Degree

** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I, Amylia Binti Mohammad Hasly hereby affirm that the work provided in this final year project titled Bat Optimization for Crowd Evacuation Simulation in Classroom is entirely my own and has not been submitted for any other academic qualification in any other school. I confirm that the work is original and has not been plagiarized from any other source, save as acknowledged in the text. All sources utilized in the development of this project have been properly recognized and referenced in accordance with the citation style.



(17 JANUARY 2025)

AMYLIA BINTI MOHAMMAD HASLY

ACKNOWLEDGEMENT

First of all, I would like to express my sincerest appreciation to my supervisor, Madam Hamizan Binti Sharbini, for her priceless advice, support, and unwavering motivation during this project. Her knowledge in simulation, optimization algorithms and her commitment to assisting me in reaching my objectives have played a crucial role in the completion of this project.

I would also like to express my sincere thanks to the faculty and staff of the Faculty of Computer Science and Information Technology at Universiti Malaysia Sarawak (UNIMAS) for offering me the essential resources and environment to undertake this project.

I am deeply grateful to my family and friends for their endless love, support, and compassion throughout my academic journey. Their support and encouragement have been a constant source of inspiration and strength for me to overcome the challenges faced throughout this project.

Finally, I want to express my gratitude to all the participants who spend their time to fill out the survey and offer valuable insights. Their contributions have been essential in influencing the path and enhancing the quality of this project. I sincerely value the effort and insightful viewpoints they provided to me. Thanks to all who have played a part in achieving success for this project.

Table of Contents

DECLARATION.....	2
ACKNOWLEDGEMENT.....	3
LIST OF TABLES.....	VII
LIST OF FIGURES.....	VIII-X
ABSTRACT.....	XI-XIII
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction	1-2
1.2 Problem Statement.....	2
1.3 Project Scope.....	2-3
1.4 Aims and Objectives	3
1.5 Methodology.....	3-4
1.6 Significance of Project.....	4
1.7 Project Schedule.....	5-6
1.8 Expected Outcomes.....	6
1.9 Report Outline.....	6-7
CHAPTER 2: LITERATURE REVIEW.....	8
2.1 Introduction.....	8

2.2 Definition of Crowd Evacuation Simulation.....	8-9
2.3 Crowd Behaviour.....	9
2.4 Types of Simulation Tools.....	9
2.4.1 AnyLogic.....	10
2.4.2 NetLogo.....	11
2.4.3 Pathfinder.....	12
2.4.4 MATLAB.....	13
2.4.5 EXODUS.....	14
2.5 Why Choose MATLAB in this Project.....	15
2.6 Reviews on Similar Works.....	15
2.6.1 Bat Algorithm: A Novel Approach For Global Engineering Optimization.....	15-16
2.6.2 Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review.....	16
2.6.3 Genetic Algorithm Optimization Problems.....	17
2.6.4 Analysing Emergency Evacuation Strategies For Mass Gatherings using Crowd Simulation and Analysis Framework: Hajj Scenario	17-18
2.6.5 PSO-Based Evacuation Simulation Framework	18
2.7 Comparison between the Similar Works.....	19-22
2.8 Reviews on Tools and Used.....	22

2.8.1 Software used for Development.....	22
2.8.2 Hardware used for Development.....	23
2.9 Summary.....	23
CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN.....	24
3.1 Introduction.....	24
3.2 Methodology for Development.....	24-25
3.2.1 Agile Workflow Phase.....	25
3.2.1.1 Planning Phase.....	25
3.2.1.2 Development Sprints.....	26
3.2.1.3 Testing and Validation Phase.....	26
3.2.1.4 Iteration and Refinement Phase.....	27
3.2.1.5 Deployment and Evaluation Phase.....	27
3.3 Requirement Analysis.....	28
3.3.1 Functional Requirements.....	28-29
3.3.2 Non-Functional Requirements.....	29
3.4 System Design.....	30
3.4.1 System Framework.....	30
3.4.2 Flowchart.....	31-32

3.4.3 Path Selection.....	33-34
3.4.4 Obstacles Avoidance.....	34-35
3.4.5 User Interface.....	35-38
3.5 Tools and Resources.....	39
3.5.1 Software Used for Development.....	39-40
3.5.2 Hardware Used for Development.....	40
3.6 Implementation of Bat Algorithm.....	40
3.6.1 Key Components of Bat Algorithm.....	41
3.6.2 Algorithm Steps.....	41-43
3.7 User Demographics	43
3.7.1 Demographic Information.....	43-46
3.7.2 Understanding and Insights.....	47-49
3.7.3 Suggestions and Feedback.....	50-51
3.8 Summary.....	51-52
CHAPTER 4: IMPLEMENTATION AND RESULT.....	53
4.1 Introduction.....	53
4.2 Implementation.....	54
4.2.1 Evacuation Model Simulation Interface.....	54-56

4.2.2 Run Simulation Based on Fire Emergency Scenario	56-60
4.2.3 Final Simulation.....	60
4.3 Summary.....	61
CHAPTER 5: EVALUATION.....	62
5.1 Introduction.....	62
5.2 Testing Framework and Scenarios	62
5.2.1 Simulaton Testing Approach	62
5.2.2 Test Parameters.....	63
5.3 Results and Discussion.....	63
5.3.1 Evacuation Time and Path Optimization.....	63
5.3.1.1 Test Scanario 1.....	63-65
5.3.1.2 Test Scenario 2.....	65-66
5.3.1.3 Test Scenario 3.....	67-68
5.3.2 Comparison od Evacuation Time Between Different Exit Scenarios.....	69-71
5.3.3 Effect of Agent Placement and Density	72
5.4 Validation and Limitations.....	72
5.4.1 Model Validation.....	72
5.4.2 Model Limitations.....	72-73

5.5 Summary.....	73
------------------	----

CHAPTER 6: CONCLUSION AND FUTURE WORK.....	74
---	-----------

6.1 Conclusion.....	74
---------------------	----

6.2 Limitations.....	75
----------------------	----

6.3 Future Work.....	75-76
----------------------	-------

6.4 Recommendations.....	76-77
--------------------------	-------

REFERENCES.....	78-80
------------------------	--------------

LIST OF TABLES

Table 1: Comparison between the similar works.....	19-21
Table 2: Results of Evacuation Time with One Exit.....	63
Table 3: Results of Evacuation Time with Two Exits.....	65
Table 4: Results of Evacuation Time with All Exits Accessible.....	67
Table 5: Average Evacuation Time Between Different Exit Scenarios.....	71

LIST OF FIGURES

Figure 1: Gantt Chart for Planning Phase (FYP 1).....	5
Figure 2: Gantt Chart for System Requirement Analysis Phase (FYP 1).....	5
Figure 3: Gantt Chart for Implementation and Testing Phase (FYP 2).....	5
Figure 4: Gantt Chart for Evaluation and Quality Analysis Phase (FYP 2).....	6
Figure 5: Crowd Modeling and Simulation in AnyLogic.....	10
Figure 6: Example of NetLogo Simulation Environment.....	11
Figure 7: Pathfinder evacuation simulation of an educational building during fire scenarios.....	12
Figure 8: Example Simulation using MATLAB.....	13
Figure 9: Example scene using Exodus for evacuation simulation.....	14
Figure 10: Agile Methodology Phase.....	25
Figure 11: Framework of classroom evacuation simulation.....	30
Figure 12: Flowchart of classroom evacuation simulation.....	31
Figure 13: Flowchart of path selection.....	33
Figure 14: Flowchart of obstacles avoidance.....	34
Figure 15: Example Classroom Layout.....	35
Figure 16: User Interface for Crowd Evacuation Simulation using MATLAB.....	36
Figure 17: Visualization Results.....	38
Figure 18: Bat Set Up.....	41
Figure 19: Calculation Fitness.....	41
Figure 20: Update Bat Frequency, Velocities and Position.....	42
Figure 21: Create New Solutions.....	42
Figure 22: Accept New Solutions.....	42

Figure 23: Repeat Iterations.....	42
Figure 24: Choose Best Exit for Agents.....	43
Figure 25: Chart of Respondent's Gender	43
Figure 26: Chart of Respondent's Age.....	44
Figure 27: Chart of Respondent's Occupation.....	45
Figure 28: Chart of Respondent's Education Level.....	45
Figure 29: Chart of Respondent's Familiarity with BAT Optimization.....	46
Figure 30: Bar Chart of The Most Critical Factors for Ensuring a safe and Efficient Evacuation in Classroom.....	47
Figure 31: Bar Chart of Essential Features in Classroom Evacuation Simulation Model.....	48
Figure 32: Bar Chart of Real-Time Obstacles or Changes in Escape Routes.....	49
Figure 33: Chart of Asking Respondent's Opinions.....	50
Figure 34: Chart of What Evacuation Scenarios should be Included in Simulation.....	51
Figure 35: The Crowd Evacuation Simulation Interface.....	54
Figure 36: Low Density Scenario For Random Agent Placement.....	57
Figure 37: Low Density Scenario For Clustered Agent Placement.....	57
Figure 38: Moderate Density Scenario For Random Agent Placement.....	58
Figure 39: Moderate Density Scenario For Clustered Agent Placement.....	58
Figure 40: High Density Scenario For Random Agent Placement.....	59
Figure 41: High Density Scenario For Clustered Agent Placement.....	59
Figure 42: Total Evacuation Time (min) with 1 Exit.....	64

Figure 43: Total Evacuation Time (min) with 2 Exits.....	66
Figure 44: Total Evacuation Time (min) with 4 Exit.....	68
Figure 45: Comparison of Evacuation Time Between Different Exit Scenarios.....	69

ABSTRACT

The goal of this project is to improve crowd evacuation methods in classroom settings by developing a simulation model that employs the Bat Algorithm (BA), a nature-inspired optimization method. In emergencies such as fires or natural disasters, the ability to quickly and efficiently evacuate individuals from confined spaces is crucial for reducing injuries and deaths. Conventional evacuation models often lack the flexibility to adapt to changing factors such as variations in crowd behaviour, immediate obstacles, and modifications in exit routes. This research employs the Bat Algorithm, inspired by bats' echolocation skills to optimize evacuation routes dynamically and in real time, addressing these constraints. The simulation considers factors such as crowd density, exit accessibility and obstacles, utilizing agent-based modelling to replicate human movement during evacuations. The model was testing in three different scenarios. In Scenario 1, featuring a single available exit and a moderate crowd density (21–50 agents), the shortest average evacuation time was 1.01 minutes. Scenario 2, featuring two available exits and higher density with clustered agent placement (65–105 agents), recorded the longest average time of 1.25 minutes due to flow congestion. In Scenario 3, with all four exits open and high-density with clustered & random placement (80–140 agents), an average evacuation time of 1.02 minutes was recorded, indicating that several exits can efficiently accommodate larger crowds. The results indicate that the Bat Algorithm successfully adapts to varying conditions, reducing evacuation times and improving route effectiveness. This research presents a more flexible, data-driven approach for classroom evacuations that responds to real-time situations, ensuring safer and more effective evacuation procedures. The results are anticipated to enhance emergency response strategies and have wider applications for public safety management.

ABSTRAK

Projek ini bertujuan untuk meningkatkan kaedah pemindahan orang ramai di dalam bilik pelajaran dengan menghasilkan model simulasi yang menggunakan Bat Algorithm (BA), satu pendekatan pengoptimuman berasaskan alam. Semasa berlaku kecemasan seperti kebakaran atau bencana alam, keupayaan untuk memindahkan orang dengan cepat dan berkesan dari kawasan yang terhad adalah penting untuk mengurangkan kecederaan dan kematian. Walau bagaimanapun, model pemindahan konvensional kekurangan kemampuan untuk bertindak balas terhadap unsur dinamik seperti perubahan dalam tingkah laku orang ramai, halangan segera, dan perubahan laluan keluar. Bagi mengatasi kekangan ini, kajian ini menggunakan Bat Algorithm, yang diilhamkan daripada kemahiran ekolokasi kelawar, untuk mengoptimumkan laluan pemindahan secara dinamik. Simulasi ini mengambil kira beberapa unsur, seperti kepadatan orang ramai, akses ke pintu keluar, dan halangan dengan menggunakan pemodelan berasaskan ejen untuk menggambarkan pergerakan orang semasa pemindahan. Model ini diuji menggunakan tiga senario berbeza. Senario 1, dengan hanya satu pintu keluar yang boleh diakses dan kepadatan sederhana (21–50 ejen), mencatatkan purata masa pemindahan terpanjang iaitu 1.01 minit. Senario 2, dengan dua pintu keluar boleh diakses serta kepadatan tinggi dan penempatan ejen secara berkelompok (65–105 ejen), menunjukkan purata masa pemindahan paling lama iaitu 1.25 minit akibat kesesakan aliran. Senario 3, yang membolehkan akses kepada semua empat pintu keluar dengan penempatan ejen rawak dan berkelompok pada kepadatan tinggi (80–140 ejen), mencatatkan purata masa pemindahan sebanyak 1.02 minit, membuktikan bahawa kepelbagaian pintu keluar dapat membantu menguruskan bilangan ejen yang lebih besar secara berkesan. Keputusan menunjukkan Bat Algorithm berupaya menyesuaikan diri dengan pelbagai keadaan, memendekkan masa pemindahan dan meningkatkan keberkesanan laluan. Kajian ini

bertujuan untuk menawarkan pendekatan yang lebih sesuai dan berasaskan data untuk pemindahan bilik pelajaran yang dapat bertindak balas terhadap situasi masa nyata, memastikan proses pemindahan yang lebih berkesan dan selamat. Dapatan daripada kajian ini dijangka dapat meningkatkan rancangan pemindahan dalam tindak balas kecemasan, dengan potensi untuk diaplikasikan di pelbagai kawasan awam.

CHAPTER 1: INTRODUCTION

1.1 Introduction

In emergency situations, the ability to quickly and efficiently evacuate people from confined areas like classrooms is crucial for reducing injuries and fatalities. Classrooms, typically compact and often crowded, pose distinct difficulties during emergencies such as fires, earthquakes, or natural disasters. During these critical moments, every second is vital, and the safest, most effective evacuation path needs to be determined and adhered to. Traditional evacuation models typically do not possess the capability to adapt to sudden changes in real-time, like unforeseen crowd behaviour, obstacles on exit paths, or the altering positions of exits. These static models are usually constructed on rigid assumptions and are poorly suited to adapt to the ever-changing characteristics of crowds and emergencies.

The main objective of this project is to tackle these problems by employing the Bat Algorithm (BA), a nature-inspired optimization method, to dynamically enhance evacuation routes in real-time. The Bat Algorithm is inspired by the echolocation behaviour of bats that utilize sound waves to find their way and detect prey in intricate surroundings. Likewise, the Bat Algorithm can examine extensive solution areas and dynamically modifying its search direction, rendering it perfect for optimizing evacuation routes in a constantly altering environment.

This project seeks to create a flexible and adaptive model for classroom evacuation simulations by integrating the Bat Algorithm, which considers crowd behaviour, immediate obstacles, and alterations in escape paths. The model will optimize evacuation routes in real-

time, offering efficient paths for individuals to safely leave the classroom. This innovative method will improve existing evacuation models that frequently overlook these dynamic factors, providing a more precise and adaptable evacuation plan. In conclusion, this initiative seeks to boost evacuation efficiency, decrease evacuation durations, and increase safety during emergencies in confined areas such as classrooms.

1.2 Problem Statement

Current evacuation models may not account for dynamic factors such as crowd behaviour, real-time obstacles, or changes in escape routes. Examples of these models are Cellular Automata (CA), which replicates crowd movement on a grid. Another model is Network Flow Models, which aim to enhance the flow of individuals along established routes. The challenge lies in providing the best path that takes these factors into account, shortens the evacuation time, and guarantees the safety of everyone. The current methods are not flexible enough to respond to the dynamics of crowds in real time.

This project will examine how the Bat Algorithm can be used to optimize evacuation paths in a classroom setting, addressing the limitations in current evacuation models that unable to adapt to dynamic factors such as crowd behaviour, real-time obstacles, or changes in escape routes.

1.3 Project Scope

The scope of this project is focusing on optimizing the crowd evacuation simulation system in teaching and learning facilities with multiple exits at UNIMAS. By using MATLAB, which serves as an ideal tool for implementing BAT, the emergency strategic plan will be

enhanced, making it easier for everyone to review and understand the nearest emergency route during an evacuation. The scope of the project is given below:

- i. This crowd optimization system is developed for teaching and learning facilities (THEATER MULTIMEDIA (TMM) in FCSIT at UNIMAS.
- ii. This crowd optimization system only utilizes MATLAB's data analysis capabilities to analyse and interpret relevant data sets.
- iii. This crowd optimization system is only accessible to occupants to view the data and accessible to students to view the evacuation plan.

1.4 Aims and Objectives

The objectives of this project are:

- 1) To design the evacuation layout for a crowd simulation setting.
- 2) To apply Bat Optimization in the crowd evacuation simulation based on obstacle constraints for better path searching towards the exit way.
- 3) To analyse evacuation time based on different exit scenarios.

1.5 Methodology

The approach for this project includes applying a Bat Algorithm model specifically intended to simulate and enhance classroom evacuations. By utilizing agent-based simulation software like MATLAB or Python, the classroom setting will be simulated, considering variables such as crowd density, accessibility of exits, and the arrangement of obstacles. Multiple evacuation scenarios will be tested, each featuring unique crowd and room conditions, enabling the Bat Algorithm to dynamically identify the best escape routes. Information regarding evacuation durations, route effectiveness, and crowd dynamics will be gathered and examined to evaluate the model's efficacy, contrasting the Bat Algorithm's

results with conventional evacuation methods to validate enhancements in both speed and safety.

An Agile development methodology will be used in this project to ensure that the project can adapt and improve better. Agile is appropriate for this type of work since it allows us to test, modify, and make improvements as we go along, which is crucial for optimizing evacuation plans in dynamic settings. By dividing the project into smaller sprints, we will be able to concentrate on improving particular elements each time, such as modifying the model, testing evacuation algorithms, and adding new elements in response to the results of various scenarios. Agile methodology allows us to receive feedback fast and make modifications as needed.

1.6 Significance of Project

This project provides a flexible, data-driven method for evacuation optimization, which helps to improve evacuation strategies in confined spaces. Utilizing the Bat Algorithm, the model adapts to real-time changes in crowd behaviour and spatial constraints, offering potential improvements over static evacuation models.

1.7 Project Schedule

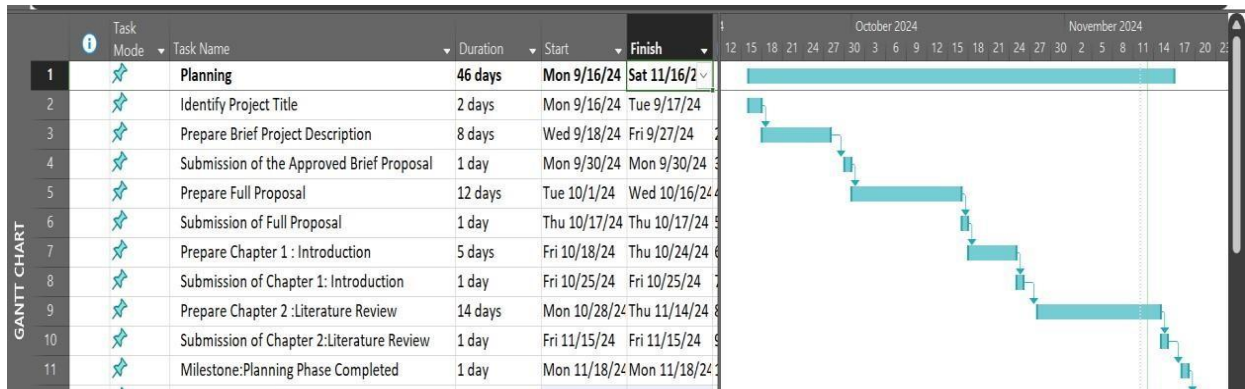


Figure 1 Gantt Chart for Planning Phase (FYP 1)

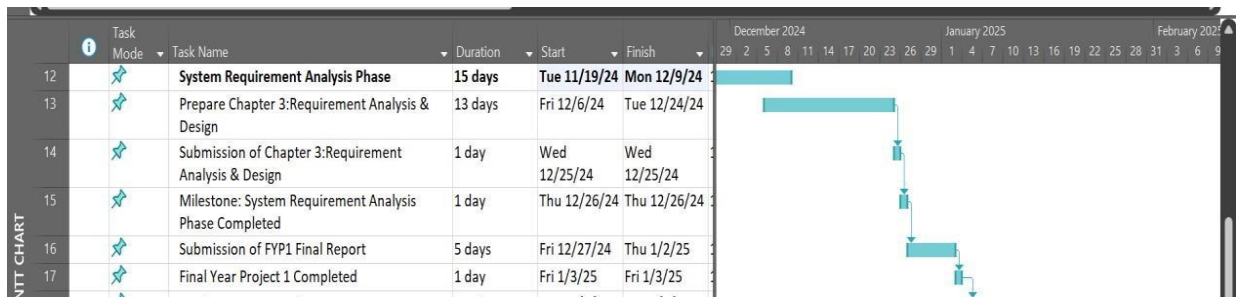


Figure 2 Gantt Chart for System Requirement Analysis Phase (FYP 1)

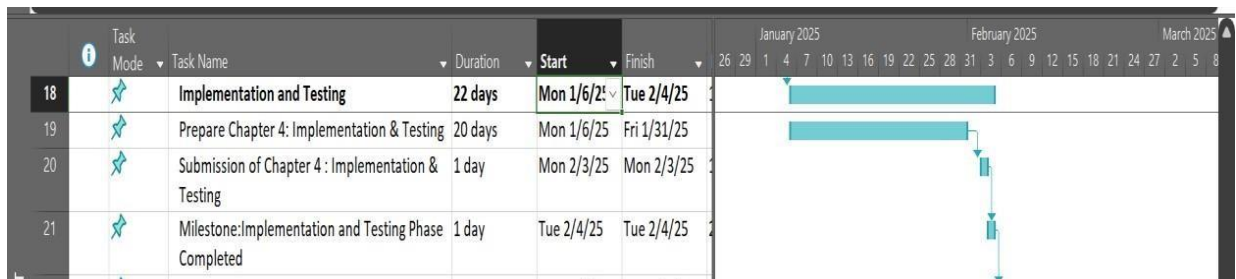


Figure 3 Gantt Chart for Implementation and Testing Phase (FYP 2)

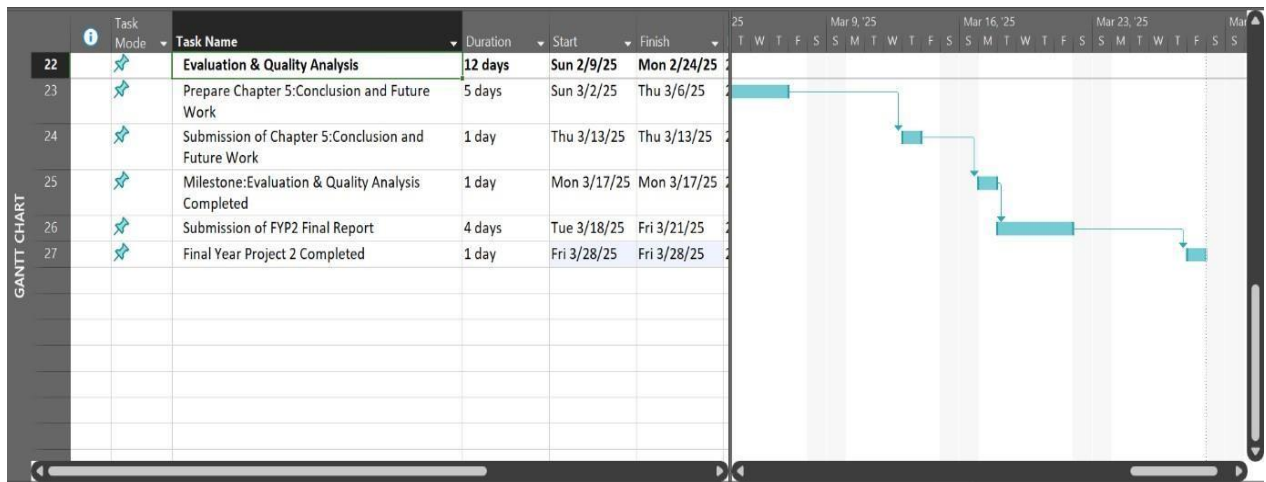


Figure 4 Gantt Chart for Evaluation and Quality Analysis Phase (FYP 2)

1.8 Expected outcomes

The outcomes are anticipated to show a significant reduction in evacuation times, illustrating the Bat Algorithm's efficacy in optimizing real-time movement. Additionally, the project will recommend best practices for classroom layout and evacuation strategies based on optimized results. This model will be a dynamic evacuation method that adapts to real-time crowd situations and exit accessibility.

1.9 Report Outline

This report is structured into six chapters that collectively provide a extensive overview of the project. Chapter 1 lays out the background and motivation for the study, elaborating on the problem statement, project scope, aims, objectives, methodology, significance, project schedule and expected results. Chapter 2 offers a comprehensive literature review detailing different crowd evacuation simulation methods, relevant optimization algorithms like the Bat Algorithm, and comparisons of current studies and simulation tools. Chapter 3 emphasizes

the requirements analysis and system design, detailing both functional and non-functional requirements, system architecture, comprehensive flowcharts, user interface design and the Agile development methodology utilized during the project. Chapter 4 outlines the execution of the simulation model, focusing on the development of the MATLAB user interface and the simulation of emergency scenarios with different crowd densities, agent behaviours and obstacle arrangements, concluded by a results analysis. Chapter 5 evaluates the simulation via systematic testing, examining evacuation times, the efficiency of path optimization, congestion resolution and the effects of agent placement and density, while also considering the validation and limitations of the model. Finally, Chapter 6 concludes the report by reviewing the project's accomplishments, addressing limitations and recommending future research directions to enhance evacuation modelling, behavioural realism and practical relevance in real-world settings.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This literature review is written to examine the current literature concerning crowd evacuation simulation, emphasizing optimization methods and their practical applications in classroom settings. The review examines important ideas including crowd behaviour, available simulation tool types, and the function of optimization algorithms such as the Bat Algorithm (BA) in enhancing evacuation plans. Moreover, this chapter examines research and initiatives that have investigated akin methodologies, contrasting their findings to offer understanding regarding the possible advantages and drawbacks of implementing BA in crowd evacuation simulation.

2.2 Definition of Crowd Evacuation Simulation

Crowd evacuation simulation involves creating models and simulations to represent the movement of people during an emergency evacuation, such as fire, natural disaster, or other dangerous situations. The simulation intends to forecast how individuals act in limited spaces (such as classrooms, theaters, and stadiums) and to evaluate the efficiency of evacuation strategies under different circumstances. Through the modelling of factors like crowd density, exit accessibility, and human decision-making, evacuation simulations assist planners and safety engineers in creating efficient escape routes, minimizing evacuation durations, and ensuring individual safety during emergencies.

Simulations of crowd evacuation play a crucial role in urban planning, architecture, transportation, and ensuring public safety. They offer valuable information on the success of

various evacuation methods and can be utilized to enhance current plans or create new ones that consider changing factors such as crowd density, obstacle arrangement, and emergency situations.

2.3 Crowd Behaviour

Crowd behaviour refers to the patterns of movement and interaction shown by a group of people in a specific area. In emergency evacuations, understanding crowd behaviour is essential for creating efficient evacuation plans. The behaviour of crowds can be shaped by numerous factors, such as individual personalities, group dynamics, panic, or cooperation, as well as the layout of the surroundings.

In evacuation simulations, crowd dynamics are generally represented through particular assumptions or guidelines that illustrate how individuals move, make decisions, and react to barriers or changes in their surroundings. Typical behaviour models consist of flocking behaviour, queueing behaviour and panic behaviour. Therefore, an accurate simulation of crowd behaviour is crucial for refining evacuation routes and guaranteeing the safety and welfare of all individuals in an emergency situation.

2.4 Types of Simulation Tools

Simulation tools are software that enable researchers to create models and simulate real-world systems. For crowd evacuation scenarios, simulation tools are created to mimic the dynamics of human movement and interaction during emergency situations.

2.4.1 AnyLogic

AnyLogic is a software platform for simulation that facilitates multi-method simulations. It combines agent-based modeling (ABM), system dynamics (SD), and discrete event simulation (DES) into one cohesive environment. It is frequently utilized for modeling intricate systems, such as logistics, healthcare, transportation, and crowd evacuation.

AnyLogic is extensively utilized for modeling crowd evacuations because of its adaptability in integrating various simulation techniques. For instance, it can simulate the personal actions of agents (pedestrians) as well as the collective dynamics of crowd movement in expansive settings. It enables researchers to model crowd behaviour based on real-time influences like obstacles, bottlenecks, or abrupt layout alterations. Using AnyLogic, individuals can simulate and enhance pedestrian movement, assess different evacuation scenarios, and analyse how various elements (such as exit location, crowd size, and emergency protocols) affect evacuation effectiveness. It also features integrated tools for visualizing and analyzing the outcomes of the simulation.

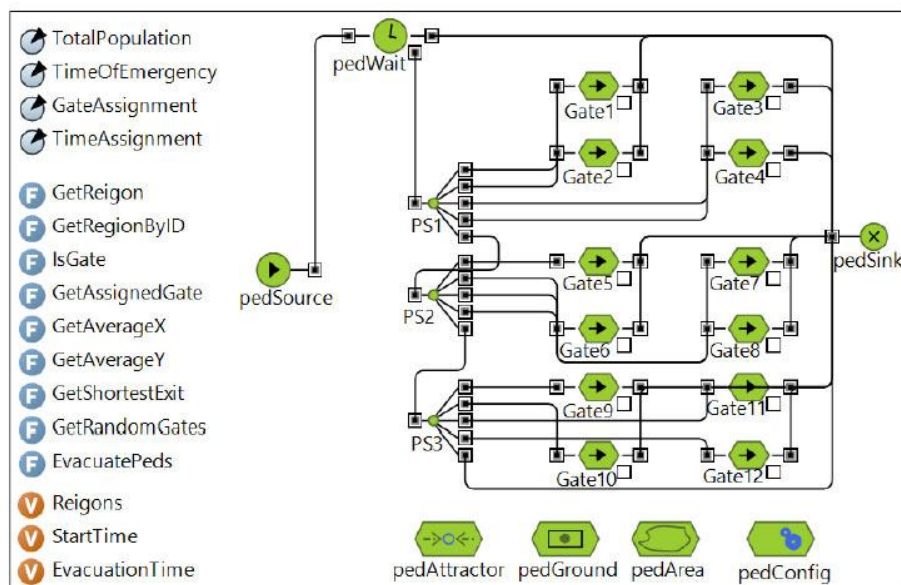


Figure 5: Crowd Modeling and Simulation in AnyLogic

2.4.2 NetLogo

NetLogo is a multi-agent programmable modeling environment that is primarily used for agent-based modeling (ABM). It enables users to model individual agents (such as pedestrians) and watch how they interact in a specific environment.

NetLogo is especially efficient for modeling pedestrian actions in crowd evacuation situations. It facilitates the modeling of numerous agents, with each agent adhering to specific behavioral rules (like heading toward an exit or steering clear of congestion). NetLogo is perfect for modeling situations where agents engage with each other and modify their actions according to nearby circumstances, such as crowd density and obstacles. This make sure it appropriate for simulating the behaviour of crowd movement in evacuations, along with evaluating the effectiveness of different crowd management techniques (including routing, exit configuration, or changes in the environment). Its user-friendly interface and robust agent-based modeling features make it a favoured tool in crowd evacuation research.

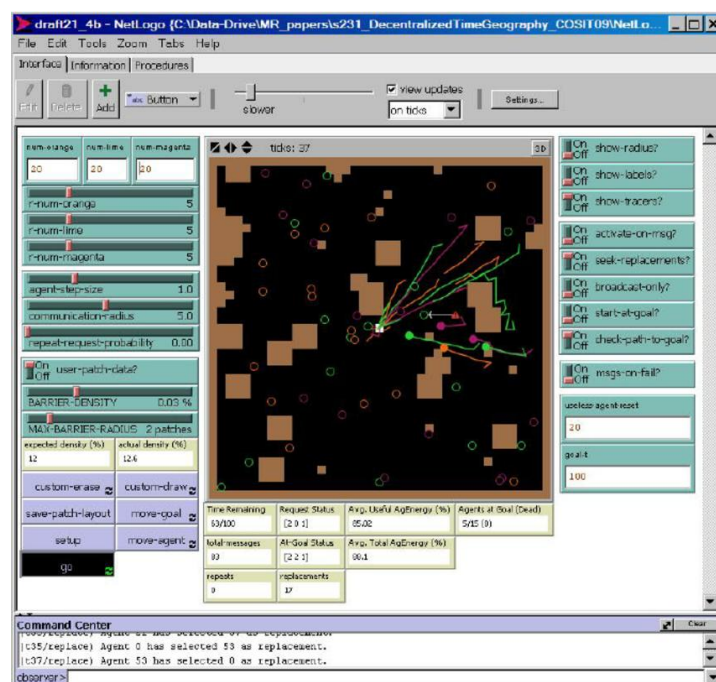


Figure 6: Example of NetLogo Simulation Environment

2.4.3 Pathfinder

Pathfinder is a software tools for crowd simulation that is tailored for modeling pedestrian movement in complex environment. It is commonly utilized in architectural design, urban planning, and emergency management. Pathfinder enables users to model the movement of crowds within buildings or other areas, considering both individual behaviours and overall crowd dynamics.

Pathfinder is perfect for simulating crowd actions during emergency evacuations, as it is tailored to simulate pedestrian movement through large buildings or densely populated areas. It can simulate how groups react to different situations, such as emergencies, changes in exit availability, or the emergence of obstacles. The program employs physics-based models to simulate the interaction between agents and their surroundings, aiding in the optimization of evacuation paths and the reduction of bottlenecks. It is especially beneficial for grasping how architectural designs (e.g., the arrangement of exits, stairs, and corridors) affect the effectiveness of evacuations and can aid in enhancing the formulation of evacuation strategies.

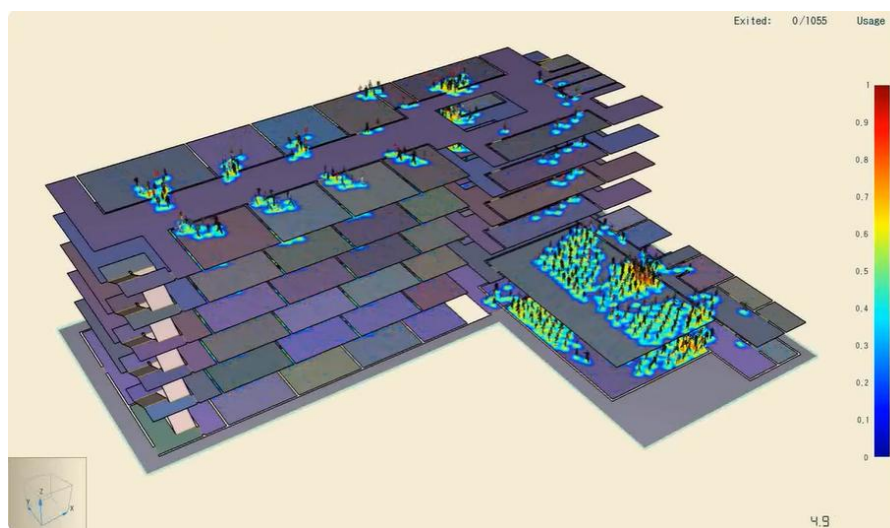


Figure 7: Pathfinder evacuation simulation of an educational building during fire scenarios

MATLAB is a high-level programming language and environment mainly utilized for numerical computing, data analysis, and developing algorithms. It encompasses integrated functions and toolboxes that enable users to carry out intricate mathematical calculations and develop simulations for diverse applications, such as engineering, physics, and social sciences.

13

2.4.5 EXODUS

EXODUS is a software tools designed for crowd simulation, utilized to model the evacuation of pedestrians in complex environment. Developed by the University of Greenwich, it relies on the idea of social force models and provides simulations at both microscopic and macroscopic simulations.

EXODUS is specifically designed to model pedestrian behaviour during emergency evacuations. It can simulate both personal pedestrian actions (such as avoiding obstacles and adjusting speed) and overall crowd behaviours (such as congestion and flow bottlenecks). EXODUS is capable of simulating different situations, ranging from typical movement to emergency evacuations, and it enables the evaluation of multiple evacuation strategies. By simulating pedestrian behaviour in reaction to environmental signals (like fire alarms or obstructed exits), EXODUS aids planners in creating safer evacuation paths, optimize building layouts, and minimizing the potential of accidents or injuries during evacuations.



Figure 9: Example scene using Exodus for evacuation simulation

2.5 Why Choose MATLAB in this project

Among the simulation tools reviewed, MATLAB was chosen as the main development environment for this project because of various significant benefits. MATLAB provides a strong and flexible environment for implementing complex optimization algorithms such as the Bat Algorithm, enhanced by its comprehensive mathematical and computational toolboxes. Its integrated environment enables seamless combination of numerical computation, data visualization and algorithm development, which is crucial for real-time simulation and analysis of crowd evacuation scenarios. In contrast to dedicated crowd simulation tools like AnyLogic or Pathfinder that prioritize agent-based modelling and graphical simulation, MATLAB offers greater control over algorithmic customization and accurate computational modelling. Additionally, MATLAB's extensive utilization in academic and engineering fields, combined with its robust capabilities for rapid prototyping and debugging, makes it an ideal choice for experimental and research-oriented projects. Although tools such as NetLogo and EXODUS provide useful frameworks for simulating crowd dynamics, MATLAB's adaptability in applying advanced optimization methods and its compatibility with other programming languages like Python offer a more thorough and scalable solution for achieving the objectives of this project.

2.6 Review on similar works

2.6.1 Bat Algorithm: A Novel Approach for Global Engineering Optimization

In this article, Yang, X. & Gandomi, A.H. (2012) introduce the Bat Algorithm (BA) as an innovative nature-inspired optimization method, particularly intended for global engineering optimization challenges. The algorithm draws inspiration from the echolocation abilities of bats, allowing them to move through their surroundings by producing high-frequency sounds. The

authors emphasize how BA merges exploration and exploitation skills to circumvent local minima and enhance the efficiency of global searches. The study additionally contrasts the Bat Algorithm with other recognized optimization methods, showcasing its efficiency in tackling intricate optimization challenges. The authors utilize BA on various test functions and engineering issues to demonstrate its flexibility and effectiveness in addressing real-world optimization problems. The conclusion highlights BA's capability in engineering fields, particularly in optimizing functions that involve many variables or intricate search landscapes.

2.6.2 Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review

Gad, A.G. (2022)'s comprehensive review of the Particle Swarm Optimization (PSO) algorithm examines its evolution, modifications, and uses in various fields. PSO, influenced by social patterns like bird flocks and schools of fish, has emerged as a popular optimization algorithm because of its ease of use and reliability. The review classifies different enhancements and hybrid variations of PSO, including its integration with techniques like genetic algorithms (GA) and differential evolution. The writer offers a comprehensive analysis of the advantages of PSO in addressing both continuous and discrete optimization challenges in areas such as robotics, control systems, machine learning, and data mining. Nevertheless, the document also addresses the limitations of PSO, including its vulnerability to premature convergence and challenges in handling large-scale or highly complex issues. The review wraps up by proposing future research avenues, especially in combining PSO with other metaheuristic methods for improved outcomes.

2.6.3 Genetic Algorithm Optimization Problems

Sivanandam, S. & Deepa, S. (2008) investigate the application of Genetic Algorithms (GA) in optimization challenges, emphasizing their utility in addressing intricate, non-linear, and multi-modal problems. The writers describe how GA operates, mimicking natural selection through selection, crossover, mutation, and fitness assessment to identify the best solution. They analyze the historical evolution and theoretical basis of GA, addressing its resilience and flexibility for various optimization issues in areas such as engineering design, scheduling, and machine learning. The paper also discusses typical issues in GA, such as the necessity of achieving an appropriate balance between exploration and exploitation, the danger of early convergence, and the tuning of parameters. The authors determine that, in spite of these difficulties, GA serves as a strong optimization tool and suggest employing hybrid methods to address the algorithm's natural constraints.

2.6.4 Analysing Emergency Evacuation Strategies for Mass Gatherings using Crowd Simulation and Analysis Framework: Hajj Scenario

Mahmood et al. (2017) present an agent-based crowd simulation and analysis framework aimed at improving emergency evacuation strategies for mass gatherings, with a focus on the Hajj pilgrimage scenario. The authors utilize the Anylogic Pedestrian Library to model large crowds in spatially explicit environments, simulating complex crowd behaviours such as evacuation during emergencies. The framework incorporates optimization techniques, including genetic algorithms, to evaluate and improve evacuation strategies by reducing evacuation times. The study highlights three evacuation approaches: Random Crowd Evacuation, Shortest Regional Distance (SRD), and Genetic Algorithm (GA), with GA demonstrating the best performance in

minimizing evacuation time. The proposed framework can be employed for public safety and security during large-scale events.

2.6.5 PSO-Based Evacuation Simulation Framework

Tsai, P.-C., Chen, C.-M., & Chen, Y.-p. (2014) present a simulation framework for evacuation based on Particle Swarm Optimization (PSO) aimed at modelling human actions and improving evacuation strategies in architectural designs. The framework streamlines the configuration of different simulation scenarios by representing individuals, obstacles, exits, and evacuation guides as elements of the PSO algorithm. The framework seeks to enhance realistic human behaviour during evacuations by integrating improvements like local search mechanisms and a global best value updating system that considers non-line-of-sight effects. Experimental findings demonstrate the framework's ability to simulate various situations, such as exit placements, obstacles, and evacuation guide indicators, aiding in the enhancement of evacuation strategies. This framework is open source, providing a flexible tool for simulating evacuations.

2.7 Comparison between the similar works

Table 1: Comparison between the similar works

Author(s)	Title	Optimization Method	Strengths	Limitations	Conclusion
Yang, X. & Gandomi, A.H. (2012)	Bat Algorithm: A Novel Approach for Global Engineering Optimization	Bat Algorithm (BA)	- Balances exploration and exploitation - Efficient in large, complex search spaces - Robust against local minima	- Can be computationally expensive for certain problems - Tuning parameters may be challenging	BA is a promising tool for global optimization in engineering, showing advantages over traditional methods.
Gad, A.G. (2022)	Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review	Particle Swarm Optimization (PSO)	- Simple and easy to implement - Works well in continuous and discrete problems - Can be hybridized with other algorithms	- Prone to premature convergence - Struggles with large-scale or complex optimization problems	PSO is versatile and efficient but requires modifications or hybridization to handle complex, high-dimensional problems.

Sivanandam, S. & Deepa, S. (2008)	Genetic Algorithm Optimization Problems	Genetic Algorithm (GA)	- Suitable for complex, nonlinear, and multi-modal problems - Can handle a wide variety of optimization problems	- Risk of premature convergence - Parameter tuning and balance between exploration and exploitation are challenging	GA is powerful for optimization, especially in complex problems, but hybrid approaches are recommended for better results.
Mahmood, I., Haris, M., & Sarjoughian, H. (2017)	Analysing Emergency Evacuation Strategies for Mass Gatherings using Crowd Simulation and Analysis Framework: Hajj Scenario	Genetic Algorithm (GA), Shortest Regional Distance (SRD), Random Crowd Evacuation	- Integrates Anylogic simulation for modelling crowds using an agent-based approach. - Allow simulation of complex crowd behaviours. - Merging with outside optimization and analysis components.	- Significant computational demands for extensive crowds. - The balance between simulation accuracy and processing efficiency. - Confined to situations where exit routes are accessible, and crowd	This study demonstrates that GA is the most effective method for reducing evacuation time in Hajj situations. Nonetheless, the approach encounters computational difficulties in extensive scenarios and might need additional

				behaviours are comparable.	adjustments for fluid or intricate environments.
Tsai, P.-C., Chen, C.-M., & Chen, Y.-p. (2014)	PSO-Based Evacuation Simulation Framework	Particle Swarm Optimization (PSO)	-Offers a basic and broad framework for simulating evacuations. Simulates human actions, challenges, solutions, and navigation signals utilizing PSO. -Enhances the realism of human behaviour through local search and updates to the global best value updating mechanisms.	- Might not completely represent complex human behaviours such as emotional or physical elements. -Environmental settings must be manually modified for various situations.	-The framework useful for enhancing evacuation strategies and building design assessments but has limitations in fully capturing complex human behaviours and may require adjustments for different scenarios.

In conclusion, the Bat Algorithm (BA), Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) possess distinct advantages in addressing optimization problems. The Bat Algorithm is notable for its effective balance between exploration and exploitation, making it very efficient at navigating large and complex solution spaces. Its inspiration from bats' echolocation allows for dynamic adaptation and robust global search ability, which is especially beneficial in real-time and changing environments such as crowd evacuation simulations. Additionally, BA shows resilience to local minima and provides flexible parameter tuning, enhancing its robustness and efficiency. These characteristics make the Bat Algorithm an excellent choice for optimizing evacuation routes in changing classroom environments, providing accurate and flexible solutions that improve safety and effectiveness.

2.8 Reviews on Tools and used

To develop this project, MATLAB is the core software for implementing the BAT optimization algorithm, additional tools for modelling, data visualization, and hardware for real-time testing or simulation is needed too.

2.8.1 Software used for development

- MATLAB - A central tool for optimizing crowd evacuation routes and putting the BAT algorithm into practice.
- Simulink - To simulate the dynamic behaviour of the classroom and the crowd flow, representing both static and moving agents.
- Python - For flexibility in optimization and simulation tasks

2.8.2 Hardware used for development

- Laptop – Used to run MATLAB, Simulink, Python etc.

2.9 Summary

Chapter 2 reviews the literature on crowd evacuation simulation, emphasizing optimization method such as the Bat Algorithm (BA) to enhance evacuation strategies, especially in classroom settings. It describes that crowd evacuation simulations depict human movement in emergencies, considering factors such as crowd behaviour, exit accessibility, and environmental layout. The chapter explores different simulation software like AnyLogic, NetLogo, Pathfinder, MATLAB, and EXODUS, which assist in modelling crowd behaviour and improving evacuation paths. Furthermore, it evaluates optimization methods such as BA, Particle Swarm Optimization (PSO), and Genetic Algorithms (GA), highlighting their advantages and drawbacks in addressing intricate issues. The chapter wraps up by underscoring the significance of integrating these techniques and instruments to create more effective and secure evacuation strategies.

CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN

3.1 Introduction

This chapter emphasizes the requirement analysis and design of an innovative evacuation simulation model that utilizes the Bat Algorithm (BA), a nature-inspired optimization technique, to address the shortcomings of traditional methods. Inspired by bats' echolocation skills, the Bat Algorithm thrives in dynamic optimization, allowing the discovery of ideal evacuation paths even in swiftly changing conditions. This initiative seeks to create a simulation model specifically designed for classroom settings, tackling immediate challenges and enhancing the effectiveness and safety of evacuation procedures.

The requirements analysis will detail the specific demands of the project, encompassing both functional and non-functional requirements, whereas the design segment will provide an in-depth architecture for the simulation model. By combining the Bat Algorithm with agent-based simulation methods, the model will replicate different evacuation scenarios, offering insights into the best evacuation strategies across varied situations.

3.2 Methodology for Development

The Agile development approach is chosen for this project because of its adaptability and repetitive structure. Agile enables ongoing enhancement of the simulation model via iterative development, testing, and feedback loops. The approach allows for immediate reactions to unexpected difficulties, improving the system's capability to accurately simulate and refine evacuation routes.

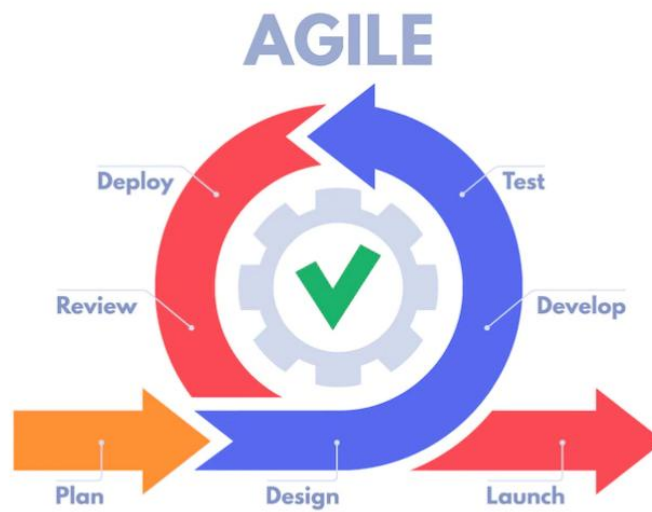


Figure 10: Agile Methodology Phase

3.2.1 Agile Workflow Phases

3.2.1.1 Planning Phase

The planning phase serves as the project's base, where the overall vision, objectives, and deliverables are distinctly outlined. In this phase, the project's scope is defined, emphasizing classroom evacuation situations and incorporating real-time elements like crowd dynamics and barriers. Stakeholders are recognized, and their needs are gathered to guarantee the model corresponds with actual requirements. Furthermore, the timeline, milestones, and deliverables are detailed, offering a straightforward roadmap for the project's advancement. This phase also includes pinpointing the essential elements, like the input module for environmental setup and crowd factors, the processing module for integrating the Bat Algorithm, and the output module for visualizing results and performance indicators.

3.2.1.2 Development Sprints

The development phase is segmented into several brief cycles known as sprints, with each focusing on particular elements of the project. These sprints are repetitive, enabling gradual advancement and ongoing improvement. During the initial sprint, a fundamental simulation framework is developed, featuring a virtual classroom design and preliminary crowd dynamics. The second sprint concentrates on applying the Bat Algorithm to enhance evacuation pathways in fixed situations. The third sprint introduces dynamic crowd behaviours, including panic and collaboration, whereas the fourth sprint integrates real-time updates regarding obstacles and exit accessibility. Every sprint enhances the last, guaranteeing a gradual improvement of the simulation model while permitting modifications according to feedback and testing outcomes.

3.2.1.3 Testing and Validation Phase

Once the essential elements are created, the project moves into the testing and validation phase. This phase guarantees that the simulation model achieves its goals and operates efficiently across different scenarios. Thorough testing is performed to replicate various classroom settings, crowd densities, and emergency scenarios. The performance of the Bat Algorithm is evaluated against conventional evacuation techniques to confirm its advantages in evacuation time and safety. Test feedback is examined to pinpoint possible enhancements, and the model is progressively improved to boost its precision, flexibility, and dependability.

3.2.1.4 Iteration and Refinement Phase

During this phase, the project concentrates on evaluating the outcomes from the testing phase and enhancing the model as needed. The feedback loop serves as an essential element, as findings from testing are integrated into the model to overcome constraints and enhance performance. Modifications are applied to settings, including crowd behaviour guidelines, obstacle identification, and Bat Algorithm configurations. This repetitive method guarantees that the model develops and enhances gradually, making it strong enough to manage various real-time evacuation situations efficiently.

3.2.1.5 Deployment and Evaluation Phase

The deployment phase signifies the completion of the project, during which the simulation model is completed and set for practical use. The model is organized into user-friendly tool, and detailed documentation is created, including setup guides, user manuals, and case studies. The model is implemented for real-world application, and its effectiveness is assessed through extensive testing in various circumstances. This stage also includes creating final reports and suggestions derived from the simulation outcomes, offering important insights for enhancing classroom evacuation methods.

By adhering to these Agile workflow phases, the project guarantees a structured, iterative, and feedback-oriented method for creating a strong, adaptable, and effective classroom evacuation simulation model. Every step aid in developing a solution that is flexible, precise, and in harmony with the actual requirements of emergency evacuation planning.

3.3 Requirement Analysis

Requirement analysis is an essential stage in every project, as it forms the basis for design, development, and implementation. In this project, the requirement analysis centers on grasping the needs of the classroom evacuation simulation and exploring how the Bat Algorithm can be incorporated to enhance evacuation routes in real-time. The assessment reveals both functional and non-functional requirements that the system needs to fulfill, as well as the data and environmental limitations that are present.

3.3.1 Functional Requirements

Functional requirements outline the essential tasks the system must execute to meet its objectives. These criteria outline what the system must accomplish, concentrating on functionalities that aid in simulating and optimizing classroom evacuations.

1. Simulation Setup

- Build a virtual classroom setting featuring adjustable layouts, exit locations, and obstacle arrangements.
- Facilitate the inclusion of dynamic crowd actions and instantaneous obstacles.

2. Bat Algorithm Implementation

- Create a module for enhancing evacuation paths utilizing the Bat Algorithm.
- Incorporate elements like sound waves, frequency variation, and exploration of the search area.

3. Modelling of Dynamic Behaviour

- Model crowd behavior, incorporating panic, collaboration, and path modification.

- Incorporate live updates regarding barriers, population density, and exit accessibility.

4. Visualization and Analysis

- Provide 2D visualizations of evacuation scenarios.
- Generate performance metrics, including evacuation times and route efficiency.

3.3.2 Non-Functional Requirements

1. Performance

- Ensure the simulation processes large datasets efficiently and produces results in real-time.

2. Scalability

- Design the model to accommodate different classroom sizes and varying crowd densities.

3. Usability

- Provide a user-friendly interface for configuring scenarios and viewing results.

4. Reliability

- Guarantee consistent and accurate results across multiple simulations.

5. Adaptability

- Allow for seamless integration of new algorithms or behavioural models.

3.4 System Design

3.4.1 System Framework

The framework of the system specifies the organization and interconnection among various elements within the simulation model. For this project, the framework aims to combine the Bat Algorithm, dynamic crowd simulation, and real-time adaptability into a unified system.

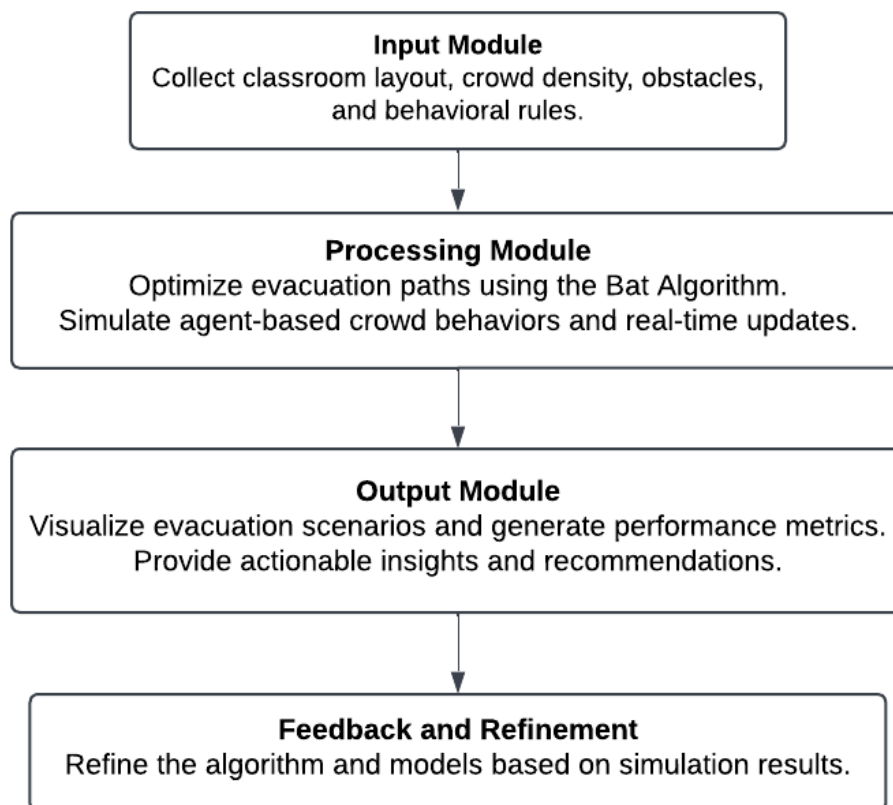


Figure 11: Framework of classroom evacuation simulation.

3.4.2 Flowchart

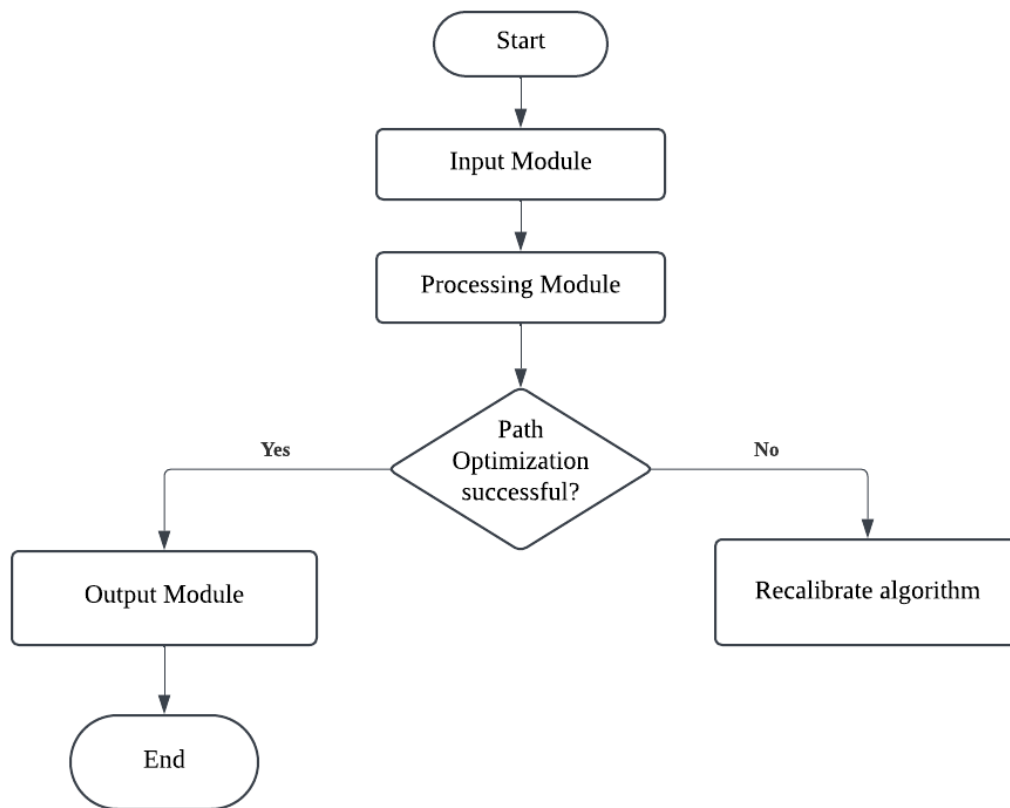


Figure 12: Flowchart of classroom evacuation simulation.

Figure 12 outlines the organized structure for the Bat Optimization applied to Crowd Evacuation Simulation in a Classroom initiative. It starts with the Start phase, during which the simulation is set up, establishing the groundwork for data gathering and setup. In the Input Module, key information including classroom structure, population density, obstacle positioning, and agent behaviour rules is collected. This information forms the basis for the simulation, guaranteeing it reflects real-life situations correctly.

Subsequently, the procedure continues to the Processing Module, where the Bat Algorithm is utilized to find the best evacuation routes. This module actively models agent-based behaviours, such as moving toward exits, avoiding crowding, and adjusting in real-time to environmental

shifts like obstructed exits or new barriers. The system subsequently assesses the efficiency of the refined paths at the Decision Point marked "Path Optimization Successful?" Should the optimization fulfil the requirements, the procedure advances to the Output Module. If that's not the case, it returns to the Recalibration Module.

In the Recalibration Module, the system enhances the algorithm and behavioural models by modifying parameters or simulation inputs. These modifications are returned to the processing phase for reassessment. After the paths are effectively optimized, the Output Module displays the results, offering evacuation metrics like overall evacuation time, congestion analysis, and exit usage. Suggestions for enhancing classroom designs or evacuation plans are produced at this point.

Lastly, the simulation concludes in the End phase, during which all outcomes and insights are gathered. The integration of the recalibration loop guarantees the system remains adaptable and iterative, enhancing its capability to manage intricate, evolving evacuation situations efficiently. This modular and decision-based method guarantees that the simulation is resilient, flexible, and in harmony with the project's goals.

3.4.3 Path Selection

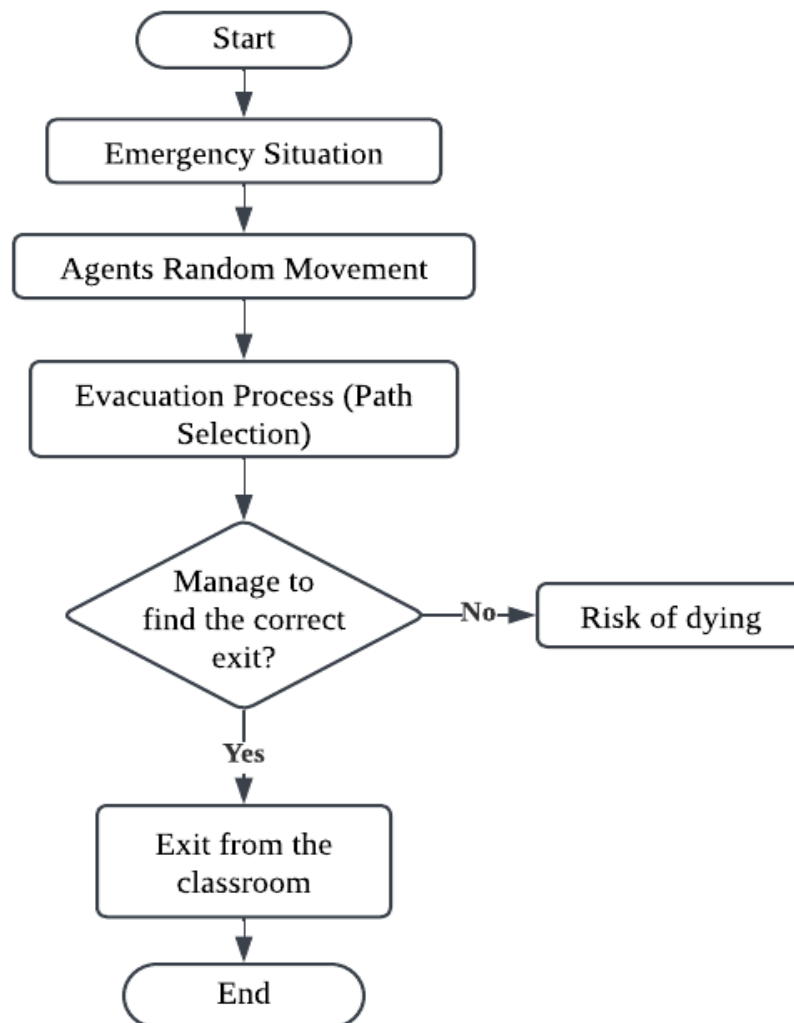


Figure 13: Flowchart of path selection

The flowchart for path selection illustrates the order of steps that agents follow during an emergency evacuation. It starts with the "Start" node, which triggers the process. When an emergency occurs, agents begin to move haphazardly around the classroom looking for a way out. The subsequent step entails the evacuation procedure, during which agents deliberately choose routes that lead to the exits. A decision point occurs, at which it is evaluated if the agent accurately recognizes the appropriate exit. If the agent successfully locates the correct exit,

they move to leave the classroom, concluding the process. Nonetheless, if the agent cannot locate the proper exit, they risk death because of the continuing emergency. In the end, the flowchart depicts the journey from the onset of the emergency to either a successful escape or a disastrous result, depending on the agent's skill in finding the exit. This method can be applied to model crowd behaviour in different emergency situations, enhancing the choice of routes and evacuation plans.

3.4.4 Obstacles Avoidance

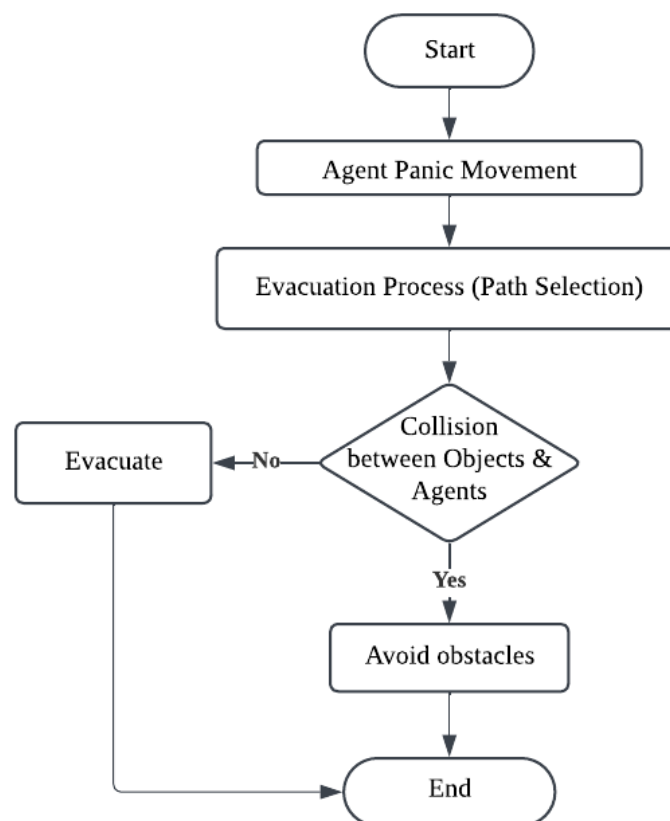


Figure 14: Flowchart of obstacles avoidance

The obstacle avoidance flowchart describes the steps agents take during an evacuation when there are obstacles present. The process begins with the agent entering a "panic movement"

mode, where the agent is responding under pressure due to the emergency situation. Subsequently, the evacuation process begins, with the agent selecting a possible exit path.

At the decision point, the flowchart check if there is a collision between the agent and the obstacles. If there is no collision, the agent proceeds with the evacuation towards the exit, and the flow concludes. However, if a collision occurs, the agent must avoid the obstacle by changing its path. Once the obstacle is avoided, the agent continues its evacuation, and the process concludes.

This flowchart simulates the behaviour of agents during emergencies, considering environmental obstacles and how agents adapt their movement to avoid collisions, thus ensuring a safer and more effective evacuation process.

3.4.5 User Interface

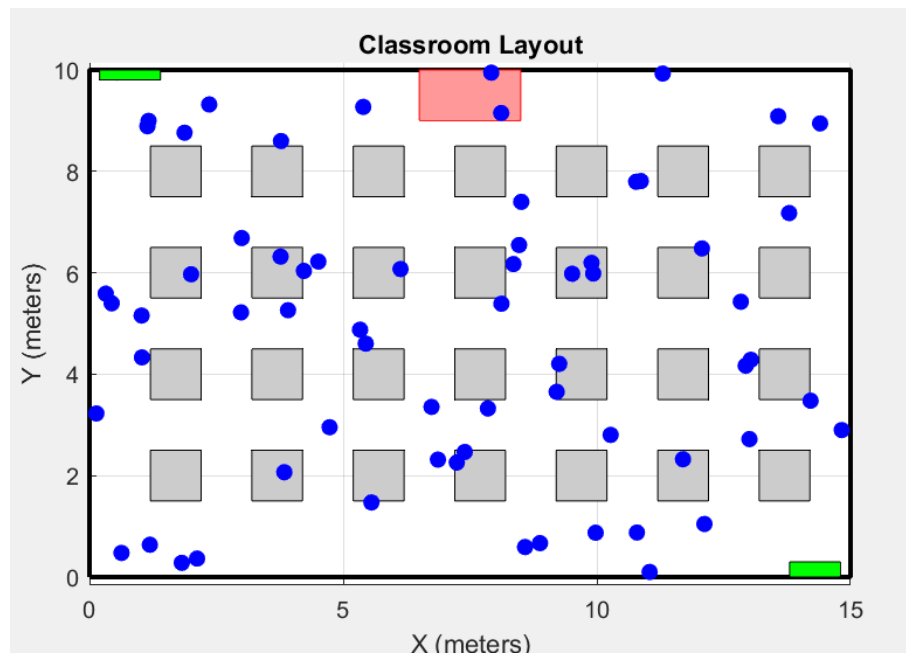


Figure 15: Example Classroom Layout

Figure 15 depicts a comprehensive classroom arrangement illustration produced by the simulation. It encompasses:

Desks: Represented as grey square organized in a grid pattern.

Students: Represented as blue dots dispersed across the classroom.

Teacher's Desk: Depicted as a red rectangle located near the upper centre of the classroom.

Exit Doors: Shown as green rectangles in the top-left and bottom-right corners of the classroom.

This graphic depicts the arrangement of desks, the location of the teacher, and exits, as well as the simulated placements of students, either distributed randomly or grouped together.

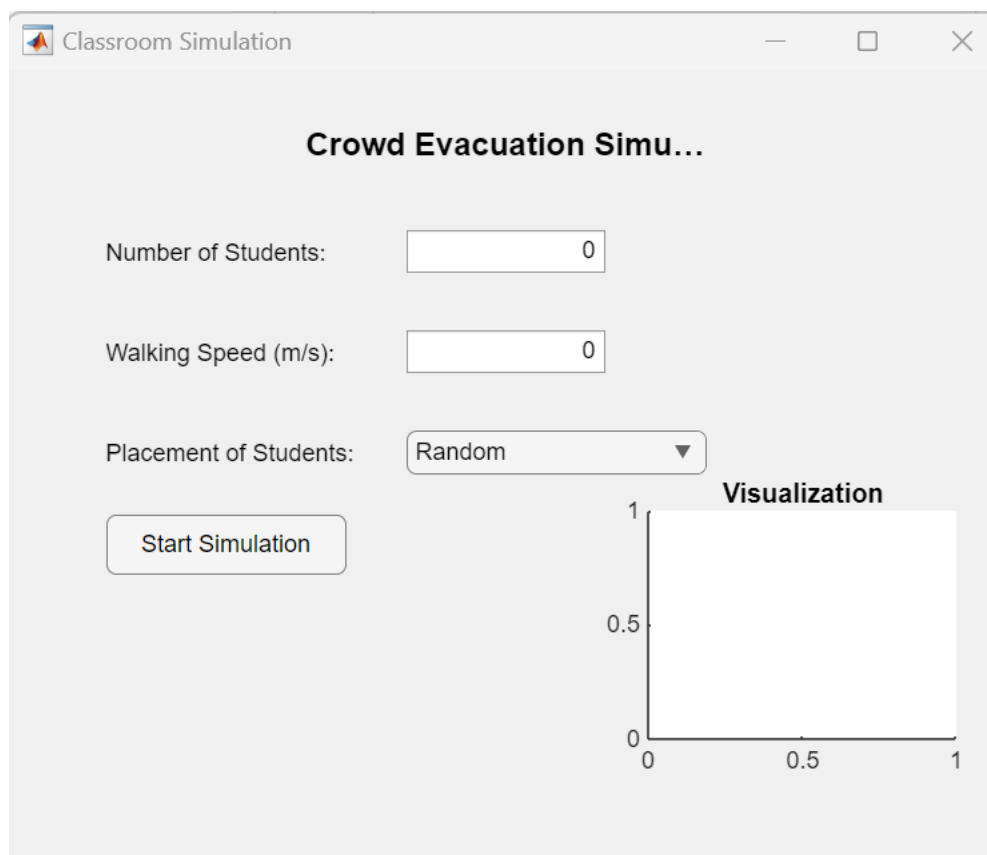


Figure 16: User Interface for Crowd Evacuation Simulation using MATLAB

Figure 16 displays a straightforward user interface developed in MATLAB to simulate a Classroom Evacuation Scenario. This interface is crafted to be simple and intuitive, enabling users to effortlessly set up the necessary parameters for the simulation. At the top, the heading, "Crowd Evacuation Simulation," distinctly shows the tool's intended use. Beneath the title, three primary input areas are present: the "Number of Students" section, where users can enter the total number of students to simulate; the "Walking Speed (m/s)" section, which enables users to define the walking speed of the students in meters per second; and a dropdown labelled "Placement of Students," providing two choices—"Random" for distributing the students randomly throughout the classroom and "Clustered" for positioning them close to specific areas.

On the left side of the interface, there is a "Start Simulation" button that initiates the simulation according to the values given by the user. On the right side, a part titled "Visualization" serves as a placeholder for showcasing the simulation results, including the spatial arrangement of students and the classroom setup. This interface showcases a structured and effective design, emphasizing simplicity while allowing users to set inputs and observe outputs effortlessly. Its user-friendly design and simplicity make it ideal for both novices and experienced users, highlighting MATLAB's flexibility in creating interactive simulation tools.

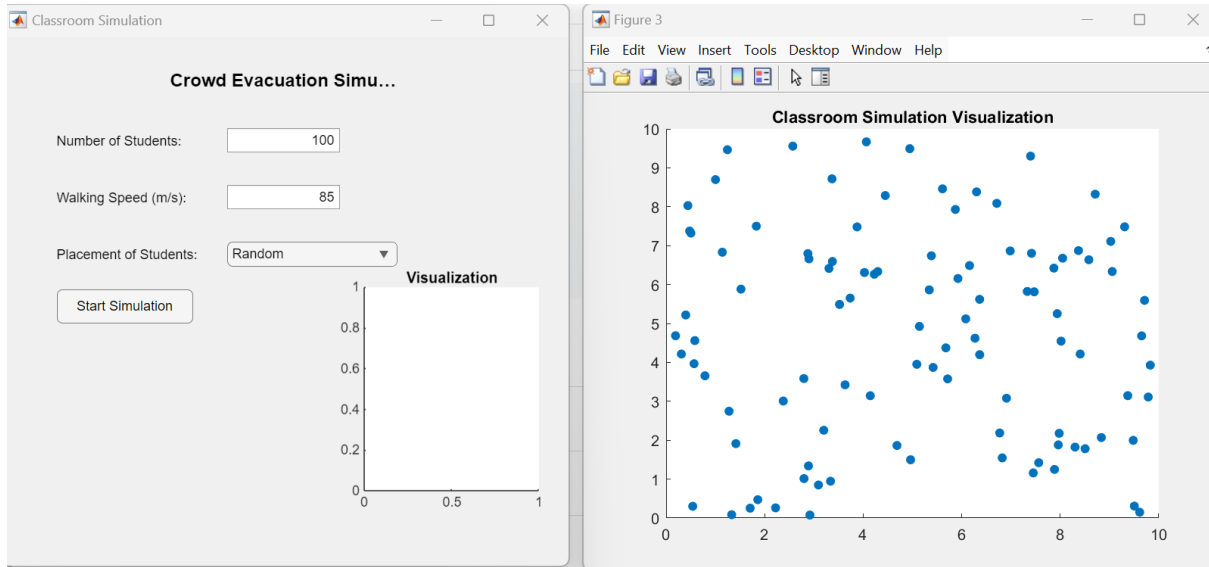


Figure 17: Visualization Results

Figure 17 demonstrates the Classroom Simulation User Interface functioning after the user has provided certain inputs and initiated the simulation. The interface is created in MATLAB and presents user-defined parameters for the simulation on the left side, while the resulting visualization appears in a separate figure window on the right side. In the interface, the user has inputted 100 for the "Number of Students," 85 m/s for the "Walking Speed," and chosen "Random" from the "Placement of Students" dropdown. Upon selecting the "Start Simulation" button, the simulation is initiated.

The figure on the right presents a scatter plot illustrating the random arrangement of 100 students throughout a classroom area. Every blue dot in the graph signifies a student, and the arbitrary positioning illustrates the choice of "Random" for the placement type. The simulation visually represents the locations of students according to the given parameters. This configuration showcases the capabilities of the MATLAB interface, where user entries are handled to produce visual feedback in real time. It underscores the user-friendly and efficient nature of the interface in replicating classroom situations and illustrating evacuation dynamics.

3.5 Tools and Resources

The creation of the classroom evacuation simulation model depends on a blend of software and hardware resources. These instruments aid in executing the Bat Algorithm, simulating dynamic crowd behaviours, and visualizing evacuation scenarios. Selecting and using the right tools and resources guarantees effective development, strong performance, and precise outcomes. The incorporation of these tools enables smooth iterations and testing throughout the Agile development process.

3.5.1 Software Used for Development

The simulation model utilizes various software tools for different components of the project, with each one fulfilling a particular role:

- **MATLAB:** The main tool utilized for executing the Bat Algorithm and carrying out numerical simulations is MATLAB. Its strong computational abilities and vast libraries render it perfect for addressing intricate optimization challenges and modeling dynamic systems. MATLAB additionally offers sophisticated data visualization, assisting in understanding evacuation outcomes.
- **Simulink:** As a supplement to MATLAB, Simulink offers a visual interface for the modeling and simulation of dynamic systems. It is especially beneficial for illustrating the interactions between students and their surroundings, including the changing behaviour of crowds in evacuation scenarios.

- **Python:** Python is utilized to improve adaptability in simulation activities and refine elements. Its diverse ecosystem of libraries, including NumPy, Matplotlib, and Pandas, facilitates data processing, visualization, and integration with various software tools.

3.5.2 Hardware Used for Development

The hardware needs for this project are fairly modest, since the simulations demand significant computation but don't necessitate specialized equipment. The main hardware resource is a development laptop that has adequate processing capability, memory, and storage to manage simulation tasks.

- **Development Laptop:** The laptop needs to feature a contemporary multi-core CPU (e.g., Intel i7 or similar), a minimum of 16 GB of RAM. These requirements guarantee the seamless operation of MATLAB and Python scripts, live updates in Simulink, and the visualization of intricate simulation scenarios.

3.6 Implementation of Bat Algorithm

The Bat Algorithm (BA) is a nature-inspired optimization method that draws on the echolocation behaviour of bats. This project utilizes BA to dynamically optimize evacuation routes within a classroom by simulating agents (students) as they search for the shortest and safest exit paths while avoiding obstacles and congestion. The BA functions by sustaining a population of virtual bats, where each bat symbolizing a candidate solution such as a possible evacuation route. Bats modify their velocity and position based on their own experiences and the best global solution discovered, integrating frequency tuning and loudness factors to maintain a balance between exploration and exploitation.

3.6.1 Key Components of Bat Algorithm

- Position ($\mathbf{x}_i$): Indicates the current evacuation route solution of the i -th bat.
- Velocity ($\mathbf{v}_i$): Controls the bat's movement within the solution space.
- Frequency ($\mathbf{f}_i$): Used to modify velocity in real-time.
- Loudness ($\mathbf{A}_i$) and Pulse Rate ($\mathbf{r}_i$): Manage the acceptance of new solutions and the balance between exploration and exploitation.

3.6.2 Algorithm Steps

Step 1: Set up bats with random positions (routes) and velocities.

```
1 bats = repmat([studentX(i), studentY(i)], numBats, 1) + randn(numBats,2) * 0.5; % random positions
2 velocities = zeros(numBats, 2); % initial velocities
3
```

Figure 18: Bat Set Up

Step 2: Calculate fitness (evacuation time or route efficiency) for every bat.

```
1 distances = sqrt((bats(:,1) - exitX(1)).^2 + (bats(:,2) - exitY(1)).^2); % distance to exit
2 [minDist, idx] = min(distances); % find best (closest) bat
3 best = bats(idx,:);
```

Figure 19: Calculate Fitness

Step 3: Update the frequencies, velocities and positions of bats according to the global best solution.

```
1 Q = Qmin + (Qmax - Qmin) * rand(numBats,1); % random frequency for each bat
2 velocities = velocities + (bats - best) .* Q; % update velocity
3 bats = bats + velocities; % update position
4
```

Figure 20: Update Bats frequency, Velocities and Position

Step 4: Create new solutions through local random walks when the pulse rate condition is met.

```
1   if rand < r && newDist < minDist
2       best = bats(b,:);
3       minDist = newDist;
4   end
5   |
```

Figure 21: Create New Solutions

Step 5: Accept new solutions if they improve fitness, modifying loudness and pulse rate as necessary.

```
1   if (fitnessNew < fitnessBest) && (rand < A) % A is loudness
2       best = bats(b,:); % Accept the new solution as the best
3       fitnessBest = fitnessNew;
4       A = alpha * A; % Reduce loudness
5       r = r0 * (1 - exp(-gamma * t)); % Increase pulse rate
6   end
```

Figure 22: Accept New Solutions

Step 6: Continue repeating iterations until convergence or the maximum iterations reached.

```
1   for t = 1:maxIter
2       % Steps 3 and 4 are performed here
3   end
4   |
```

Figure 23: Repeat Iterations

Step 7: Choose the Best Exit for Each Agent

```
1 finalDists = sqrt((best(1) - exitX).^2 + (best(2) - exitY).^2);  
2 if ~isempty(fireExitIdx)  
3     finalDists(fireExitIdx) = inf;  
4 end  
5 [~, exitChoice(i)] = min(finalDists);  
6 |
```

Figure 24: Choose Best Exit for Agents

3.7 User Demographics

A survey is conducted to collect the user suggestions and insights, around 33 responses were collected in this survey. This survey was divided into 3 parts, including Demographic Information, Understanding and Insights, and suggestions with feedback. Conducting surveys is a crucial step in gaining insightful opinions and feedback.

3.7.1 Demographic Information

Gender
33 responses

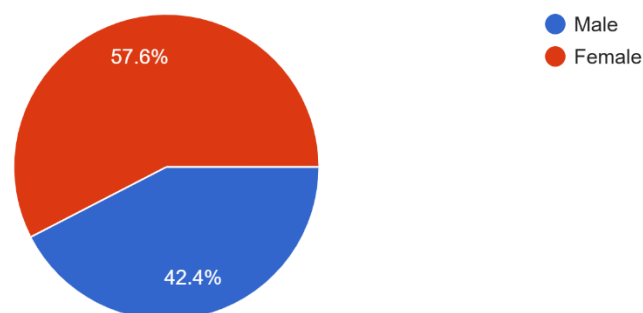


Figure 25: Chart of respondents' gender

The pie chart depicts the gender breakdown of 33 participants. It indicates that 57.6% of the participants are female, shown in red, while 42.4% are male, indicated in blue. This indicate that most of the respondents are female, with a 15.2% difference compared to male.

Age
33 responses

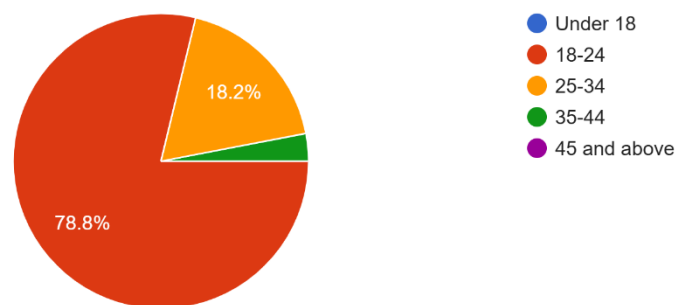


Figure 26: Charts of respondents' age

This pie chart presents the age breakdown of 33 participants. The largest section is comprised of 78.8% of individuals who belong to the 18-24 age group. The next biggest section, individuals aged 25-34, makes up 18.2%. A lesser number of respondents belong to the 35-44 and 45-and-older age categories, indicated by the green and purple section respectively. This data does not include representation for those under 18.

Occupation
33 responses

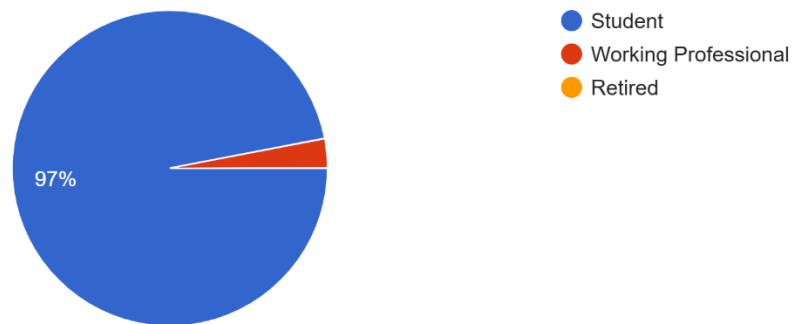


Figure 27: Chart of respondents' occupation

The pie chart illustrates how different occupations are distributed among 33 participants. Most of them, 97%, are students, shown in blue. A tiny percentage, indicated in red, consists of working professionals, whereas an even lesser section, depicted in orange, comprises retired individuals. This suggests that the survey or research mainly included students, with little involvement from other professional categories.

Education Level
33 responses

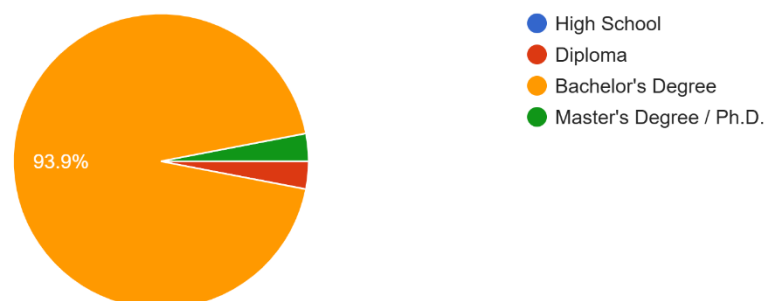


Figure 28: Chart of respondents' education level

This pie chart depicts the educational level of 33 participants. The largest portion, 93.9%, possesses a Bachelor's Degree, indicated by the orange section of the chart. The other education levels are represented by the remaining respondents: High School (blue), Diploma (red), and Master's Degree/Ph.D. (green), each making up a significantly smaller share. This information emphasizes that the majority of participants possess a high level of education, specifically at the undergraduate level.

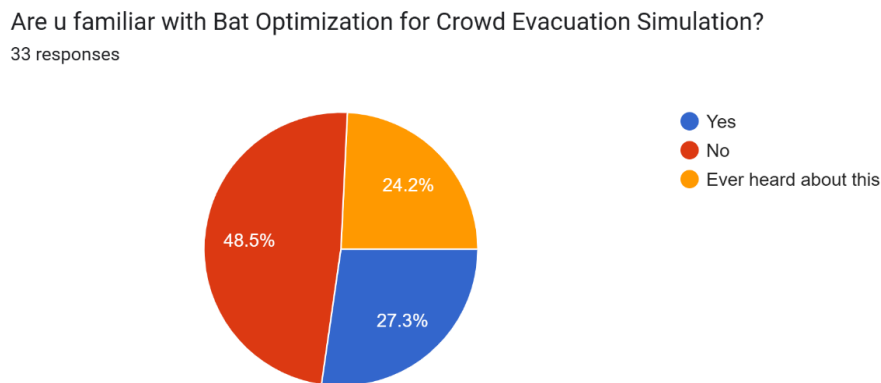


Figure 29: Chart of respondents' familiarity with BAT optimization

This pie chart illustrates the familiarity of 33 participants regarding "Bat Optimization for Crowd Evacuation Simulation." The largest group, 48.5%, expressed that they are not familiar with it (red section). At the same time, 27.3% indicated they are knowledgeable about it (blue section), while 24.2% stated they have ever heard of it (orange section). In general, the chart indicates that over fifty percent of the participants possess little or no understanding of this idea.

3.7.2 Understanding and Insights

What do you think are the most critical factors for ensuring a safe and efficient evacuation in a classroom during emergencies? (select up to 3)

33 responses

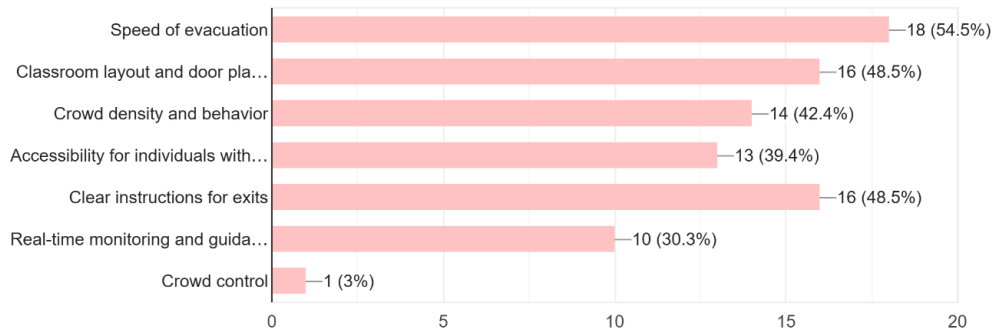


Figure 30: Bar Chart of the most critical factors for ensuring a safe and efficient evacuation in a classroom

This bar chart highlights the factors that participants view as the most important for ensuring safe and effective classroom evacuations in emergencies, allow up to three choices. The factor most frequently chosen is the evacuation speed, selected by 54.5% (18 responses). Classroom arrangement and door positioning, along with clear exit instructions, were closely followed, each chosen by 48.5% (16 responses). Crowd density and behaviour were selected by 42.4% (14 responses), whereas accessibility for disabled individuals received 39.4% (13 responses). Real-time monitoring and guidance were chosen by 30.3% (10 responses), while crowd control attracted the least interest, with merely 3% (1 response). The information highlights the significance of quick, transparent, and easily reachable evacuation procedures.

What features do you think are essential in a classroom evacuation simulation model to ensure it is realistic and effective? (select all that apply)

33 responses

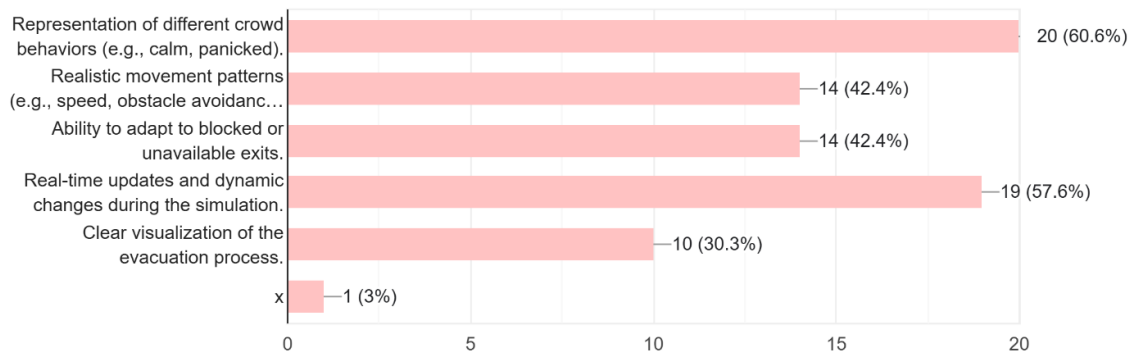


Figure 31: Bar chart of essential features in classroom evacuation simulation model

This bar chart showcases the key elements that respondents view as crucial for realistic and effective classroom evacuation simulation model, allowing multiple selections. The most selected features are the depiction of various crowd behaviours (e.g., calm, frantic), chosen by 60.6% (20 responses). Real-time modifications and dynamic alterations throughout the simulation occur at a rate of 57.6% (19 responses). Both realistic movement behaviours (e.g., speed, obstacle evasion) and the capability to adjust to blocked or inaccessible exits were chosen by 42.4% (14 responses) each. A total of 30.3% (10 responses) selected clear visualization of the evacuation process, whereas merely 3% (1 response) indicated an extra feature ("x"). The information indicates a significant inclination towards authentic crowd behaviour and flexible responsiveness in simulation models.

How should real-time obstacles or changes in escape routes be handled during evacuations in classrooms?

33 responses

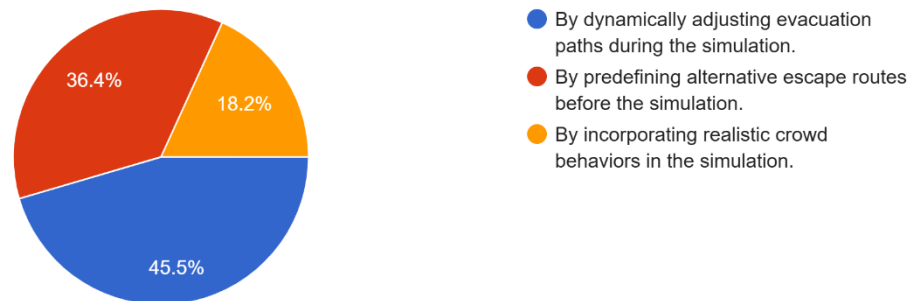


Figure 32: Chart of real-time obstacles or changes in escape routes

This pie chart depicts the preferences of respondents for handling real-time obstacles or changes in escape routes during classroom evacuations. The most favoured method, selected by 45.5% (blue section), involves dynamically modifying evacuation routes throughout the simulation. The option to predefine alternative escape routes prior to the simulation was chosen by 36.4% (red section), whereas the inclusion of realistic crowd behaviours in the simulation was selected by 18.2% (orange section). The information shows a tendency for flexible and reactive approaches to handle evacuation issues successfully.

3.7.3 Suggestions and Feedback

Do you think considering crowd behavior is important in evacuation simulations for classrooms?

Why or why not?

33 responses

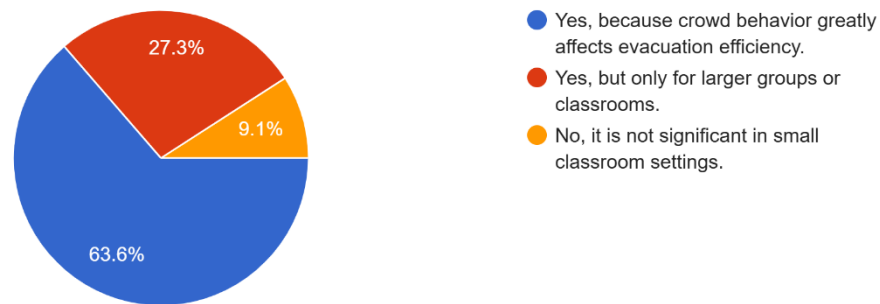


Figure 33: Chart of asking respondents' opinions

This pie chart illustrates the opinion of respondents regarding the importance of considering crowd behaviour during evacuation simulations for classrooms. Mostly, 63.6% (blue section), consider it significant since crowd dynamics significantly influence evacuation effectiveness. An additional 27.3% (red section) concur but believe it matters only for bigger groups or classrooms. A minor portion, 9.1% (orange section), does not consider it important in small classrooms. This information highlights a widespread agreement on the importance of crowd dynamics, especially in more extensive or intricate evacuation situations.

What evacuation scenarios should be included in the simulation to make the results more reliable and practical for classroom settings? (select all that apply)

33 responses

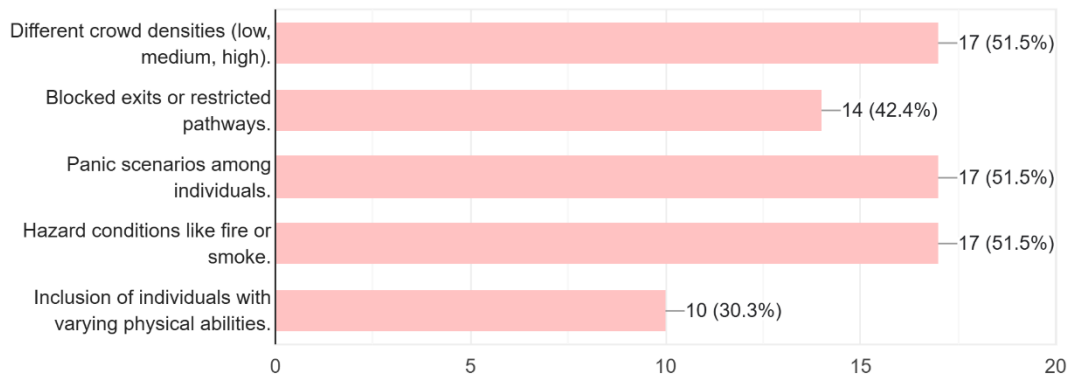


Figure 34: Chart of what evacuation scenarios should be included in simulation

This bar chart shows the evacuation situations that participants think ought to be included into a simulation to improve reliability and practicality for classroom settings. Various crowd densities (low, medium, high), panic situations among individuals, and hazard situations such as fire or smoke were chosen by 51.5% of participants (17 responses each). Blocked exits or limited pathways were selected by 42.4% (14 responses), whereas the inclusion of people with different physical abilities was chosen by 30.3% (10 responses). The findings indicate that a variety of realistic and diverse scenarios is deemed crucial for successful evacuation simulations.

3.8 Summary

Chapter 3 provides a detailed overview of the requirement analysis and design stage of the project. The Agile methodology propels the project via iterative phases, encompassing planning, development sprints, testing, refinement, and deployment, guaranteeing a systematic and adaptable strategy. The chapter outlines both functional and non-functional requirements,

encompassing simulation configuration, dynamic behaviour representation, and visualization features, facilitated by tools such as MATLAB, Simulink, and Python. A modular system structure, engaging user interface, and thorough testing guarantee accuracy, functionality, and scalability for enhancing evacuation strategies in various classroom settings.

Chapter 4: IMPLEMENTATION AND RESULT

4.1 Introduction

This chapter provides a detailed of the crowd evacuation simulation model's implementation, emphasizing emergency situations such as fires and various natural disasters within the THEATER MULTIMEDIA (TMM) at the Faculty of Computer Science and Information Technology (FCSIT), University Malaysia Sarawak (UNIMAS). The main objective of this chapter is to develop a simulation model that optimizes evacuation routes in real-time using the Bat Algorithm (BA), considering dynamic factors such as crowd behaviour, obstacles, and the availability of exits during emergencies. This simulation records the movement and decision-making behaviours of agents (individual) in the lecture hall during emergencies. This including obstacles avoidance, managing crowd congestion, and paths selection in different situations. The model simulates various emergency scenarios, such as fires and other natural disasters, with different crowd densities (low, moderate, and high) to analyse the effectiveness of evacuation strategies in different conditions. This chapter follows the requirements and design that presented in Chapter 3, with continuous testing during the implementation to ensure the simulation's accuracy and reliability. Final simulation results are shown and analysed to provide insights into potential improvements in evacuation strategies and contribute to the development of more efficient emergency response procedures.

4.2 Implementation

4.2.1 Evacuation Model Simulation Interface

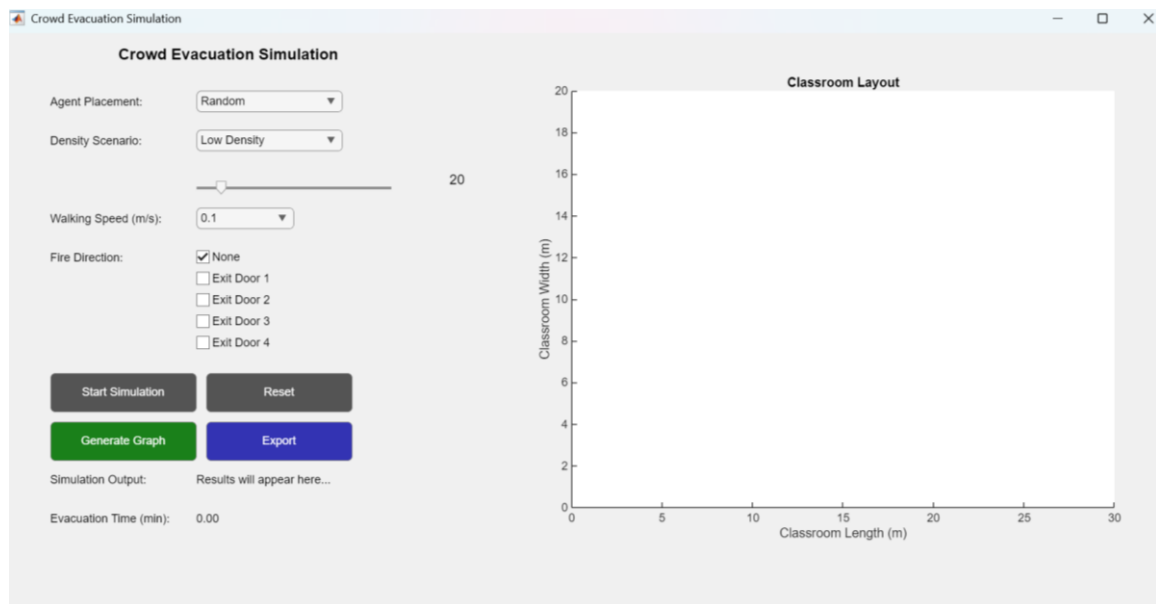


Figure 35: The Crowd Evacuation Simulation Interface

The initial phase of the implementation process involves designing an intuitive user interface (UI) in MATLAB. The interface acts as the control panel for configuring and running the evacuation simulation. Key features of the UI include:

a) Classroom Layout:

- A UI Axes element takes up the right side of the window.
- The axes are labelled in meters (Length on the x-axis: 0–30 m; Width on the y-axis: 0–20 m)
- Classroom facilities: -
 - **Long Desks:** Tiffany blue colour rectangles lined up in rows as obstacles.
 - **Student Chairs:** Represented by small square shapes in light grey arranged in rows next to the long desks.

-**Teacher Desk**: a bigger brown rectangle positioned at the front-centre serving as the podium.

-**Exit Doors**: Grey rectangles on the walls indicating the locations where students exit.

-**Stairwells**: Two large beige coloured rectangles on the left and right sides of the classroom represent the stairwells

-**Fire Visualization**: When a fire direction is selected, the blocked exit is shown with a flame visual in the classroom layout.

b) Agent Placement, Behaviour & Density Scenario:

- Agent Placement Dropdown allows selection between “random” or “clustered”.
- Agent Behaviour (Panic/Cooperation): Each agent is randomly assigned panic or cooperative behaviour, affecting their movement speed and logic.

Students are represented by green and red dots (**green** = cooperative, **red** = panic)

- Density Scenario Dropdown available in three options, (Low, Moderate, High). Internally corresponds to a slider value that determines the total count of students (e.g., 20, 50, 100 students).
- Density Slider (1 – 150 students). The numeric label updates in real-time as the slider shifts.

c) Walking Speed and Fire Direction:

- Walking Speed is dropdown option which enable user to select speeds between 0.1 to 0.5 m/s to simulate slow to fast movement.
- Fire Direction checkbox (Exit door 1, Exit Door 2, Exit Door 3, Exit Door 4)

Simulates a fire source entering from a chosen exit door. Agents will display avoidance behaviours based on this direction.

d) Simulation Control Button:

- Start Simulation – Triggers the evacuation process, the GUI disables parameter controls until the run completes.
- Reset - Clears the axes, restores all controls to their default setting (Random positioning, Low density, 0.1 m/s, no behaviours activated), and resets the evacuation time display to 0.00 min.

e) Output Displays

- Simulation Output will display status message.
- Evacuation Time (min) will dynamically display evacuation duration.

f) Graphs generate & Export Button:

- Generate Graph - This button will display a bar chart based on different exit scenarios.
- Export: Export button allows user to save the data that they had selected in user interface to Excel.

4.2.2 Run Simulation Based on Fire Emergency Scenario

The simulation's focus is on fire emergency scenarios. The system will simulate evacuations under various scenarios such as:

- 1) **Low-Density Scenario:** A scenario with a smaller number of people, enabling faster evacuation but still require path optimization for safety.

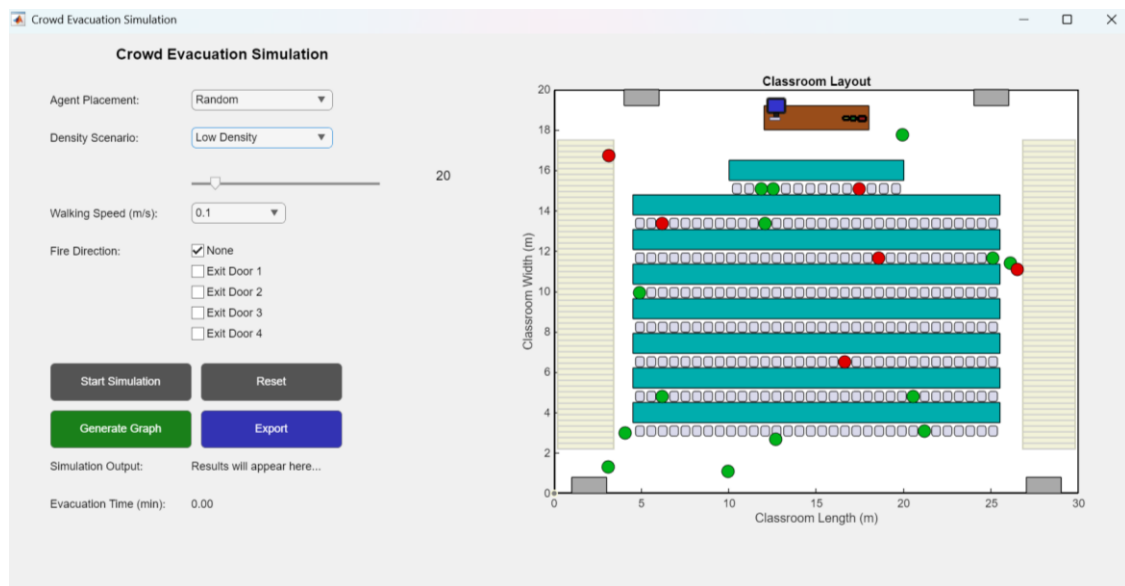


Figure 36: Low Density Scenario for Random Agent Placement

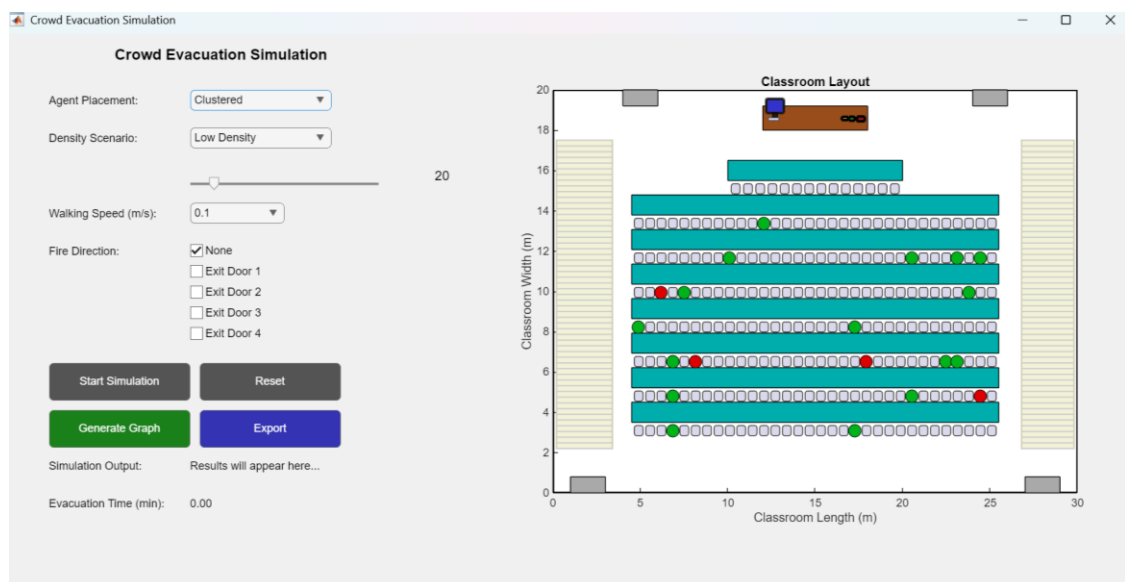


Figure 37: Low Density Scenario for Clustered Agent Placement

2) **Moderate-Density Scenario:** A scenario featuring a more manageable crowd, where the challenge lies in identifying the most suitable evacuation routes.

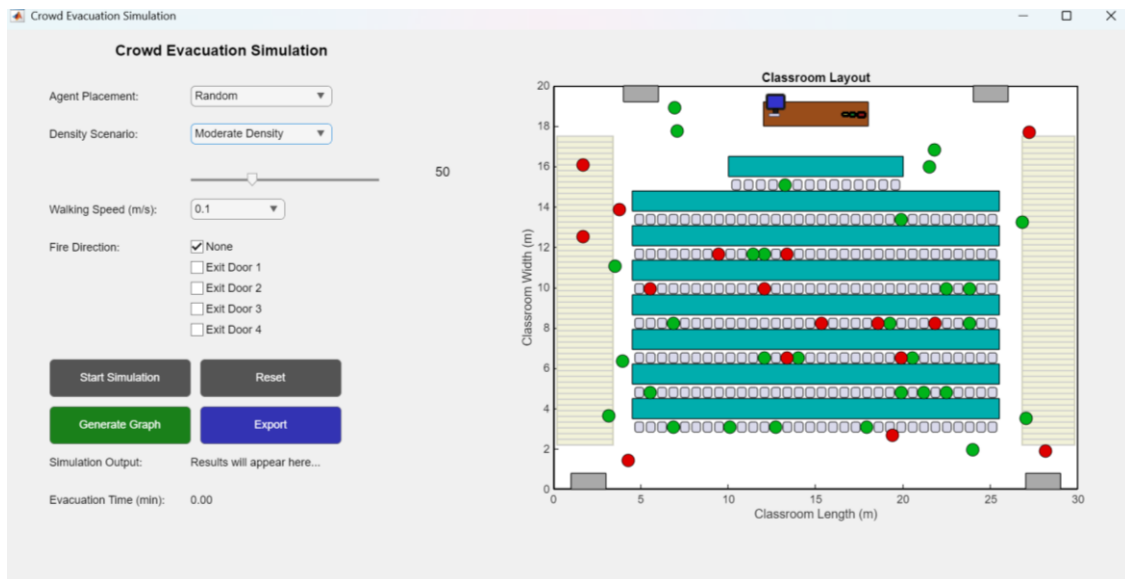


Figure 38: Moderate Density Scenario for Random Agent Placement

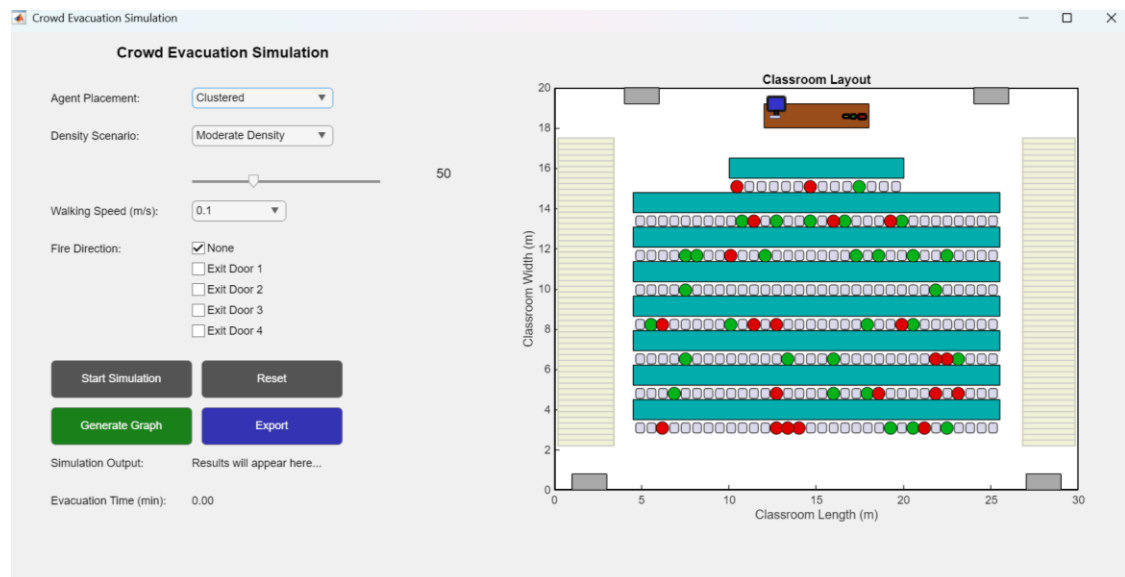


Figure 39: Moderate Density Scenario for Clustered Agent Placement

- 3) **High-Density Scenario:** A situation in which the classroom is overcrowded, increasing the difficulty of the evacuation and emphasizing the importance of optimizing evacuation paths to reduce congestion.

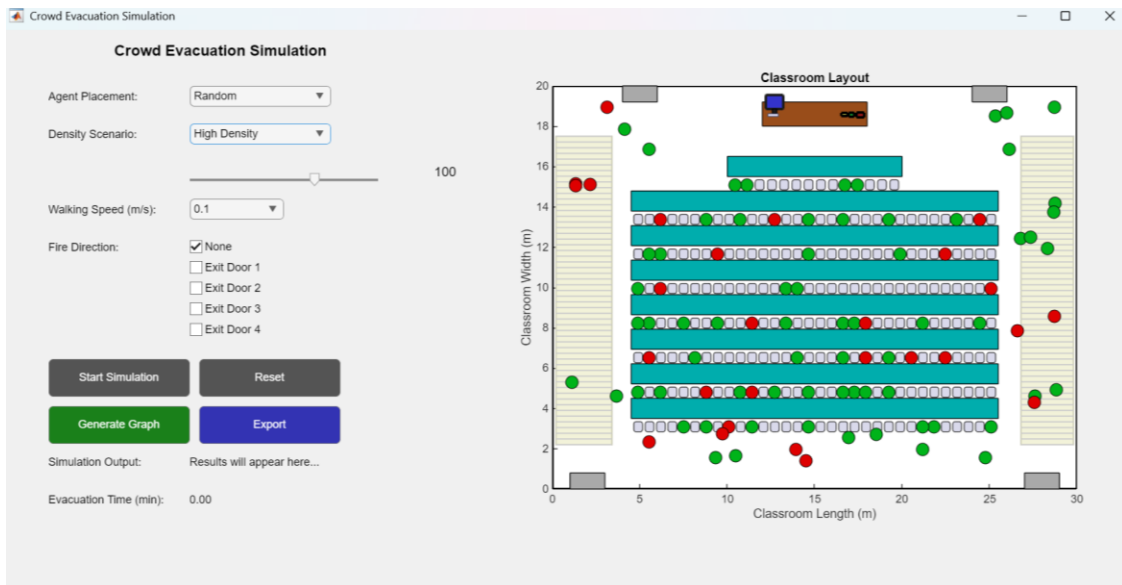


Figure 40: High Density Scenario for Random Agent Placement

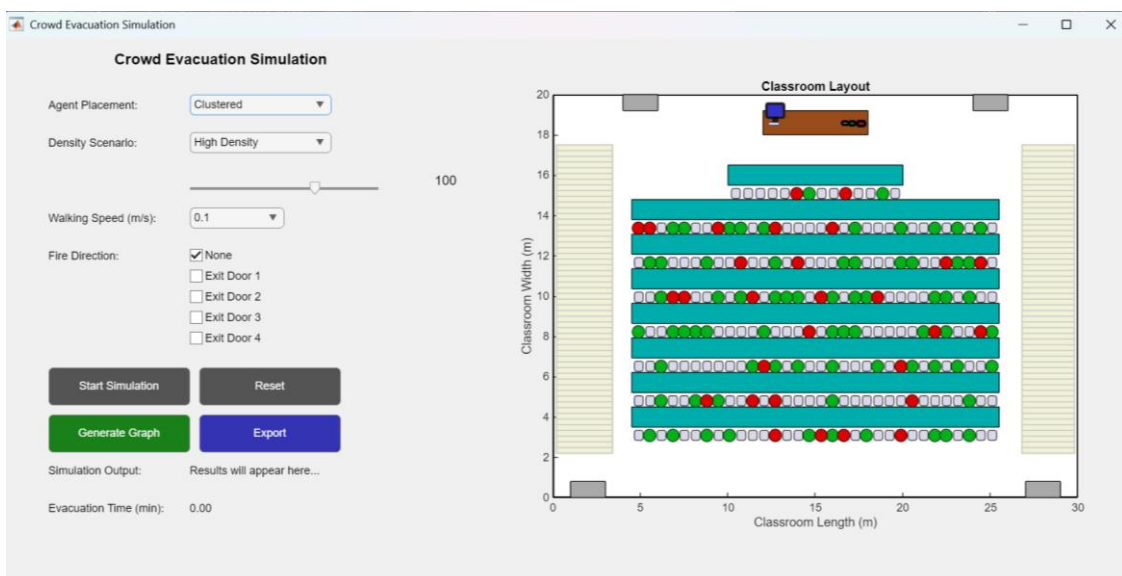


Figure 41: High Density Scenario for Clustered Agent Placement

In every fire emergency situation, challenges like fire, smoke and obstructed exits are considered. The Bat Algorithm is utilized to modify evacuation routes in real-time and considering aspects such as exit availability, obstacles and crowd behaviour. The system constantly updates the most effective evacuation path as the situation change.

4.2.3 Final Simulation

The final simulation shows the results obtained from utilizing the Bat Algorithm to optimize the evacuation process. Key outcomes from the simulation include:

- i) **Evacuation Time:** The total time taken for every student to evacuate the classroom in each scenario.
- ii) **Path Optimization:** The effectiveness of the evacuation paths, considering crowd density, obstacles and the dynamic situation of the fire emergency. The Bat Algorithm's ability to modify paths according to real-time data will be analysed.
- iii) **Crowd Behaviour Analysis:** The simulation will analyse how the crowd navigates and reacts during the evacuation, considering factors such as panic and avoidance of obstacles.

The simulation focuses on 2D evacuation scenarios, analysing evacuation time based on various exit conditions, crowd densities, and agent placements.

4.3 Summary

This chapter provided a comprehensive implementation of the crowd evacuation simulation model for the Theater Multimedia (TMM) at FCSIT, UNIMAS, focusing on emergency fire scenarios. A complete MATLAB-based interface was created, enabling users to set agent placement, density, walking speed and behavioural parameters. The simulation included classroom features like student desks, a teacher's desk, exit doors and student agents, with real-time modifications guided by the Bat Algorithm to enhance evacuation paths in dynamic conditions. Different density scenarios were examined, which is low, moderate and high density with analysis on evacuation duration, path optimization and crowd behaviour in panic and cooperation modes. The final outcomes which represented in 2D, illustrated the Bat Algorithm's enhanced flexibility and effectiveness, while providing important perspectives on emergency evacuation strategy.

CHAPTER 5: EVALUATION

5.1 Introduction

This chapter discusses the evaluation of the crowd evacuation simulation for classroom settings based on the Bat Algorithm (BA) which was developed in this project. The evaluation focuses on measuring the performance, reliability and efficacy of the simulation model across diverse scenarios such as varying crowd densities, agent placements and emergency exit arrangements. This evaluation investigates how BA utilizes real-time data to optimize routes, minimize evacuation times and reduce congestion, especially in complex and high-density environments. Quantitative outcomes (e.g., evacuation times, path efficiency) and qualitative insights (e.g., congestion resolution) are presented to provide a thorough analysis of the Bat Algorithm's impact on evacuation performance.

5.2 Testing Framework and Scenarios

5.2.1 Simulation Testing Approach

The evaluation included structured testing of the simulation model utilizing MATLAB over several predefined scenarios. Every scenario was created to reflect the realistic classroom situations in the Theater Multimedia (TMM) at FCSIT, UNIMAS, featuring obstacles, different exit path and changing student densities. The scenarios were categorized as follows:

Scenario 1: Moderate density (e.g., 21,30,50 agents), Random Placement, three exit blocked (e.g., fire at Exit 1,2,3)

Scenario 2: High Density (e.g., 65,90,105 agents), clustered placement, two exits accessible.

Scenario 3: High density (e.g., 80,110,140 agents), both clustered & random placement and all exits accessible

5.2.2 Test Parameters

For every scenario, the following parameters were assessed:

- **Evacuation Time:** Total evacuation time needed for all agents to leave the classroom.
- **Path Efficiency:** The straightforwardness and efficacy of the evacuation paths determined by the Bat Algorithm.
- **Congestion Analysis:** Identification and resolution of bottlenecks, especially close to exits.

5.3 Results and Discussion

5.3.1 Evacuation Time and Path Optimization

The Bat Algorithm consistently demonstrated shorter evacuation times, especially in high-density and dynamically changing scenarios. For example:

5.3.1.1 Test Scenario 1

Table 2: Results of Evacuation Time with One Exit

Test Scenario	Placement	Walking Speed (m/s)	Crowd Density	Blocked Exit(s)	Total Evacuation Time (min)
Test Scenario A	Random	0.1	21	Exit 1,2,3	00:27
Test Scenario B	Random	0.1	30	Exit 1,2,3	00:35
Test Scenario C	Random	0.1	50	Exit 1,2,3	00:48

Table 2 shows the total evacuation durations for three testing scenarios with increasing crowd densities (21, 30, and 50 agents). In every scenario, the placement of agents is random, the walking speed is set at 0.1 m/s, and three exits (Exit 1, Exit 2 & Exit 3) are blocked, leaving just one exit for evacuation. As the number of agents (crowd density) increases, the total evacuation time also increases, demonstrating a direct relationship between crowd density and the time needed to evacuate through a single available exit. This illustrates the increasing difficulty of handling evacuation effectively when there are limited exits.

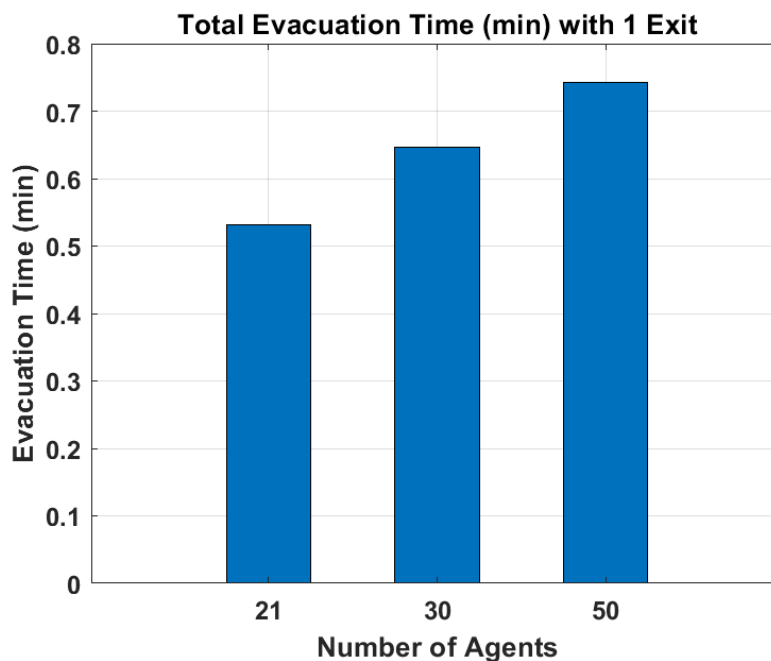


Figure 42: Total Evacuation Time (min) with 1 Exit

The bar chart illustrates the information from the table, depicting the overall evacuation time needed for varying numbers of agents (21, 30, and 50) when there is only a single exit accessible due to the blocking of Exit 1, Exit 2 & Exit 3. The x-axis indicates the number of agents (21, 30, 50), while the y-axis represents the evacuation time in minutes. The graph clearly shows that evacuation time increase with an increase in the number of agents. This emphasizes how crowd density and exit accessibility affect evacuation effectiveness. The

results highlight the need to ensure sufficient exit access, especially in crowded scenarios, to shorten evacuation times and reduce possible congestion.

5.3.1.2 Test Scenario 2

Table 3: Results of Evacuation Time with Two Exit

Test Scenario	Placement	Walking Speed (m/s)	Crowd Density	Blocked Exit(s)	Total Evacuation Time (min)
Test Scenario A	Clustered	0.1	65	Exit 3,4	1.13
Test Scenario B	Clustered	0.1	90	Exit 3,4	1.28
Test Scenario C	Clustered	0.1	105	Exit 3,4	1.35

Table 3 presents the total evacuation duration for scenarios where only two exits (Exit 1 and Exit 2) are available while Exit 3 and Exit 4 being blocked. The simulation is conducted with a constant walking speed of 0.1 m/s and agents placement in clustered. The table compare three crowd densities, 65, 90 and 105 agents. An increase in the number of students with the same exits leads to higher congestion, which slowing down the evacuation process. This effect is more pronounced at higher densities, highlighting the difficulties of controlling large crowds when exit options are limited.

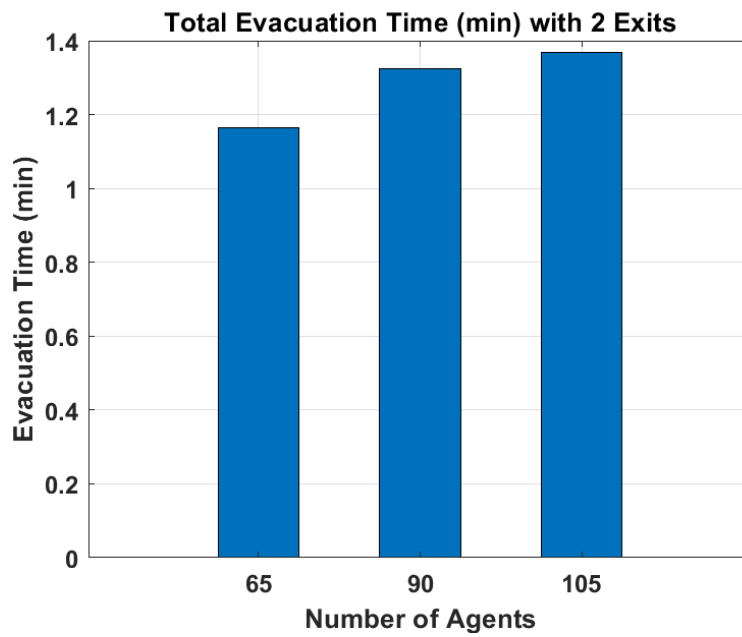


Figure 43: Total Evacuation Time (min) with 2 Exits

The bar chart visually illustrates the equivalent data from the table, demonstrating how the total evacuation time changes as the number of agents increases with only two exits accessible. The x-axis indicates the number of agents (65, 90, 105). The y-axis represents the total evacuation time in minutes. With 65 agents, evacuation was relatively fast but as the number rises to 105, the evacuation time noticeably increases.

5.3.1.3 Test Scenario 3

Table 4: Results of Evacuation Time with All Exits Accessible

Test Scenario	Placement	Walking Speed (m/s)	Crowd Density	Blocked Exit(s)	Total Evacuation Time (min)
Test Scenario A	Random	0.1	80	None	0.36
Test Scenario B	Random	0.1	110	None	0.42
Test Scenario C	Clustered	0.1	140	None	1.07

This table displays the total evacuation time for various scenarios when all four exit doors are accessible and none are blocked. The scenarios are examined under different crowd densities and agent placement including 80, 110, and 140 agents with both random and clustered placement. Evacuation times are considerably shorter compared to scenarios where exits are blocked. As the number of agents increases, the evacuation time will increase too, especially when the placement is clustered (like in Scenario C). The clustered placement at the highest density (140 agents) leads to a significantly longer evacuation time compared to random placement with fewer agents.

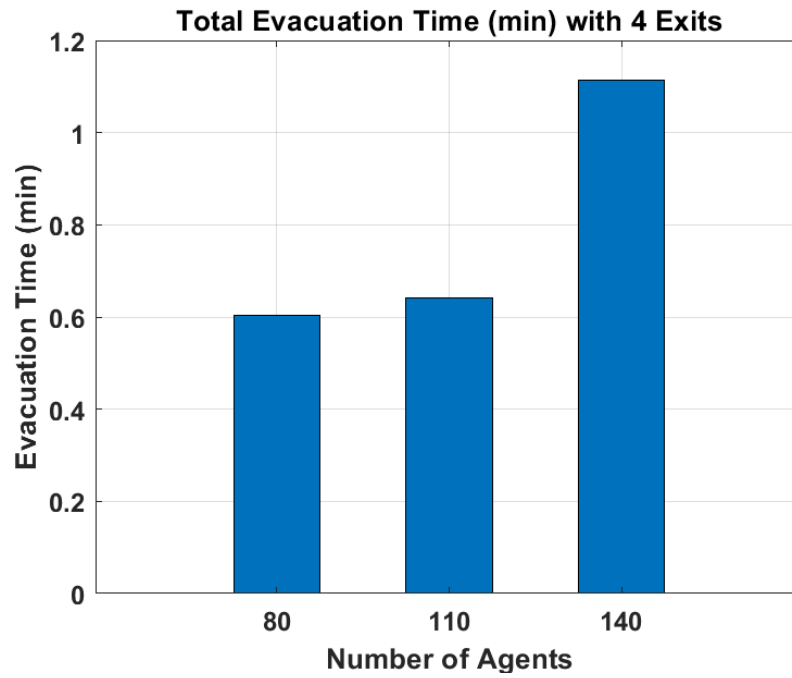


Figure 44: Total Evacuation Time (min) with 4 Exits

The bar chart illustrates the data from the table. The X-axis indicates the number of agents (80, 110, 140), while the Y-axis shows evacuation time in minutes. The chart indicates that evacuation time increases with the number of agents but stays relatively low due to all exits are accessible. The significant increase in evacuation time for 140 agents (with clustered placement) is much more pronounced, indicating that agent placement and crowding can significantly affect outcomes, even with all exits available.

5.3.2 Comparison of Evacuation Time Between Different Exit Scenarios

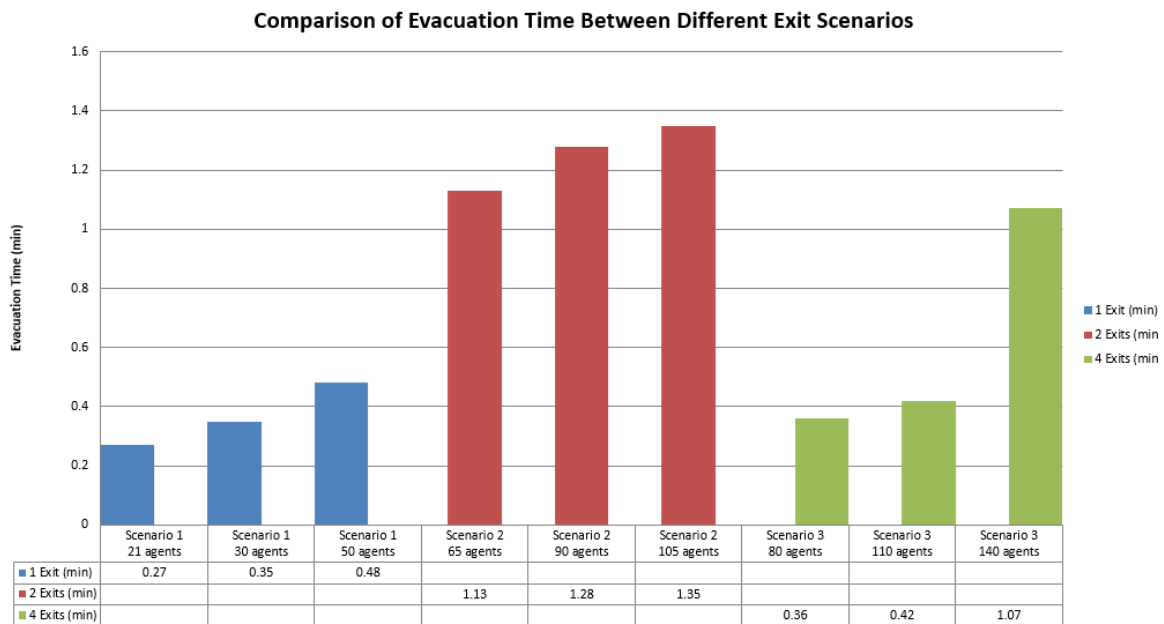


Figure 45: Comparison of Evacuation Time Between Different Exit Scenarios

The graph displays evacuation times for different scenarios with differing number of exits and agent counts, highlighting the changes in evacuation time as the number of agents increases under each exit configuration.

Explanation for Each Scenarios:

Scenario 1 (1 Exit)

- The evacuation time is the shortest since all agents move toward one exit, probably leading to a simpler and potentially more organized flow.
- Even with fewer exits, agents are not required to decide between multiple exits which could reduce confusion or delays.
- The comparatively small count of agents (21, 30, 50 agents for different densities) in this scenario also helps keep evacuation times low.

Scenario 2 (2 Exits)

- This scenario has the longest evacuation times, likely because agents are splitting between two exits, potentially resulting in congestion or ineffective distribution of agents between the exits.
- Higher number of agents (65, 90, 105 agents) are present, likely increasing the complexity of the evacuation.
- The splitting of flow and potential congestion at both exits might delay evacuation.

Scenario 3 (4 Exits)

- This situation, featuring four exits, has a slightly longer evacuation time compared to Scenario 1 but is shorter than Scenario 2.
- The increased number of agents (80, 110, 140 agents) could counterbalance the benefit of multiple exits.
- Having several exits may reduce congestion at any single exit but also requires agents to choose exits which potentially causing inefficiencies.
- Overall, the evacuation time in this scenario is similar to Scenario 1, indicating that having four exits help handle higher density but does not significantly enhance speed when compared to a properly organized single exit with fewer agents.

Conclusion:

- Scenario 1 (1 exit) has the fastest evacuation with fewer agents, probably because of its simplicity and less crowding.

- Scenario 2 (2 exits) exhibits the slowest performance, likely due to inefficient flow distribution and a moderate number of agents.
- Scenario 3 (4 exits) handles more agents but shows little reduction in evacuation time compared to Scenario 1, likely because increased density negates the benefits of extra exits.

Table 5: *Average Evacuation Time Between Different Exit Scenarios*

Test Scenario	Average Evacuation Time (min)
Scenario 1 (1 Exit)	1.01
Scenario 2 (2 Exits)	1.25
Scenario 3 (4 Exits)	1.02

Table 5 shows the average evacuation time for each exit scenario, clearly comparing the overall evacuation effectiveness of utilizing 1, 2, or 4 exits. The average evacuation times in different exit scenarios provide valuable information about the efficiency of each arrangement. Scenario 1, utilizing a single exit, demonstrates the quickest average evacuation time of 1.01 minutes, suggesting an effective movement even with limited exit options. Scenario 2, featuring two exits, unexpectedly has the longest average evacuation duration of 1.25 minutes. This could be caused by optimal crowd distribution or congestion problems that occur when agents separate between two exits. Scenario 3, utilizing four exits, reaches an average evacuation time of 1.02 minutes, very close to Scenario 1 but can support a larger number of agents. This indicates that multiple exits help manage higher crowd densities efficiently, preventing bottlenecks and ensuring a smooth evacuation process. In general, the data indicates that simply adding more exits does not ensure faster evacuation, factors such as crowd density, exit accessibility, and flow management play a crucial role in evacuation performance.

5.3.3 Effect of Agent Placement and Density

- **Random Agent Placement:** Resulted in a more scattered exit usage, slightly reducing the chance of significant bottlenecks but occasionally increasing total evacuation time due to longer initial routes for certain agents.
- **Clustered Agent Placement:** Caused higher initial congestion but enabled faster group movements when cooperative behaviour was present.
- **High-Density Scenarios:** Showed the essential significance of optimized path planning which are lack of real-time path modification, evacuation time increased significantly and highlighting the advantages of the Bat Algorithm's adaptive features.

5.4 Validation and Limitations

5.4.1 Model Validation

The outcomes are aligned with previous research, confirming that adaptive, metaheuristic optimization methods such as BA outperform conventional static models in real-time evacuation situations. Nonetheless, the validation was constrained due to the limited availability of real-world evacuation data from actual fire drills in educational environments.

5.4.2 Model Limitations

- **Simplified Agent Behaviour:** The simulation assumes consistent agent decision-making with minimal modelling of psychological and physiological influences when under pressure.
- **Fixed Layout:** This model is solely derived from the TMM classroom and might not apply to more complex or multi-level structures.

- **Lack of Real-World Calibration:** There was no calibration carried out using actual evacuation drills or sensor data.
- **Simulation of Hazards:** Fire and hazard behaviours are represented as static or moving obstacles instead of realistic and spreading phenomena.

5.5 Summary

In summary, the classroom evacuation simulation using the Bat Algorithm successfully showcased dynamic and optimized evacuation path planning in classroom scenarios. The Bat Algorithm's ability to consistently update agent routes in response to new obstacles, varying crowd densities and blocked exits enabled effective real-time optimization. Nonetheless, future validation using real-world data and extended behavioural modelling would enhance the model's realism and practical usefulness.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

In conclusion, this project has effectively developed and implemented a Bat Algorithm based simulation model aimed at optimizing crowd evacuation in classroom settings. By addressing the limitations of static evacuation models, the study incorporated real-time flexibility, behavioural variability and complex environmental factors to more accurately depict actual emergency scenarios. The simulation allowed agents to modify their paths dynamically based on varying crowd densities, blocked exits and unexpected obstacles, resulting in a more realistic and effective evacuation procedure.

The evaluation results indicated that the Bat Algorithm consistently shortened evacuation times and handled congestion effectively. The ability to simulate various behavioural modes such as panic and cooperation, also provided valuable insights into how social dynamics affect evacuation outcomes. These results emphasize the significant potential of nature-inspired optimization techniques for improving safety strategies in educational environments and similar confined spaces.

Although the model presents certain limitations, such as simplified agent behaviours and the lack of real-world drill data, it establishes a strong foundation for future research in smart evacuation planning. The insights and methodologies developed in this research aid in the progress of adaptive and data-driven strategies for emergency management, supporting continuous initiatives to build safer and more resilient educational environments.

6.2 Limitations

Despite its promising results, this research acknowledges several limitations:

- **Simplification of Agent Behaviour:** Human decision-making during emergencies were only partially modelled, such as panic, leadership, and group interaction were simplified.
- **Static Environmental Assumptions:** Only one classroom layout was simulated, actual structures tend to be more complex.
- **Hazard Modelling:** Fire and hazard dynamics were represented as static or moving obstacles, rather than as spreading or interacting phenomena.
- **Limited Validation:** The lack of real-world evacuation data or live drills restricted the ability to thoroughly validate simulation results.

6.3 Future Work

Based on the results and limitations, several suggestions for future research and development are proposed:

- 1. Improved Behavioural Simulation:** Integrate more advanced psychological and physiological models to simulate realistic human behaviours such as panic spread, group decision-making and support for vulnerable individuals.
- 2. Expansion to Multi-Room and Multi-Story Layouts:** Extend the simulation to represent larger and more complex environments such as multi-story educational facilities or large lecture halls.

3. Incorporating Real-Time Hazard Dynamics: Develop advanced hazard models that includes spreading fires and smoke, which directly affect agent decisions and the available routes.

4. Validation with Real-World Data: Collaborate with safety experts and carry out actual evacuation drills for model calibration and validation.

5. Machine Learning Implementation: Explore the application of machine learning or reinforcement learning to forecast agent behaviours and enhance evacuation plans across different emergency scenarios.

6. User Interface Improvements: Design interactive simulation tools for users to customize scenarios, analyse data, and evaluate emergency response strategies in real-time.

6.4 Recommendations

For practical adoption, it is recommended that:

- Safety planners and facility managers consider incorporating adaptive optimization algorithms such as the Bat Algorithm into their emergency response planning tools.
- Further partnerships with emergency services, architectural designers and behavioural scientists should be established to guarantee that the simulation model meets practical requirements and real-world challenges.
- Future simulation platforms support real-time data input from sensors or IoT devices to dynamically modify evacuation plans as situations change.
- Explore Different Algorithms: Examine the effectiveness of other metaheuristic and artificial intelligence algorithms such as Genetic Algorithms, Ant Colony

Optimization, or reinforcement learning for comparison and potential hybridization to achieve even better evacuation results.

- Enhanced Risk Simulation: Integrate dynamic hazard modelling, including fire and smoke spread to better depict real emergency scenarios and further challenge the optimization algorithm's adaptability.

REFERENCES

- Yang, X., & Hossein Gandomi, A. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5), 464–483. <https://doi.org/10.1108/02644401211235834>
- Sivanandam, S. N., & Deepa, S. N. (n.d.). Genetic Algorithm Optimization Problems. *Introduction to Genetic Algorithms*, 165–209. https://doi.org/10.1007/978-3-540-73190-0_7
- Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Archives of Computational Methods in Engineering*, 29(5), 2531–2561. <https://doi.org/10.1007/s11831-021-09694-4>
- Kar, A. K. (2016). Bio inspired computing – A review of algorithms and scope of applications. *Expert Systems with Applications*, 59, 20–32. <https://doi.org/10.1016/j.eswa.2016.04.018>
- baeldung. (2023, March 12). *Baeldung*. Baeldung on Computer Science. <https://www.baeldung.com/cs/the-bat-algorithm>
- Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 65–74. https://doi.org/10.1007/978-3-642-12538-6_6
- Huang, J., & Ma, Y. (2020). Bat Algorithm Based on an Integration Strategy and Gaussian Distribution. *Mathematical Problems in Engineering*, 2020, 1–22. <https://doi.org/10.1155/2020/9495281>

The Standard Bat Algorithm (BA) - File Exchange - MATLAB Central File Exchange - MATLAB Central. (2020, April 3). Mathworks.com.

<https://ch.mathworks.com/matlabcentral/fileexchange/74768-the-standard-bat-algorithm-ba>

matlabsolutions. (2016). Matlabsolutions.com. <https://www.matlabsolutions.com/genetic-algorithm/bat-algorithm-in-matlab.php>

Tong, Y., & Bode, N. W. F. (2023). Simulation investigation on crowd evacuation strategies for helping vulnerable pedestrians at different stages of egress. *International Journal of Disaster Risk Reduction*, 84, 103479. <https://doi.org/10.1016/j.ijdr.2022.103479>

Bakar, N. A. A., Majid, M. A., & Ismail, K. A. (2017). An Overview of Crowd Evacuation Simulation. *Advanced Science Letters*, 23(11), 11428–11431. <https://doi.org/10.1166/asl.2017.10298>

Ismail, F. S., Mohamad, S., Isa, K., Shah, S. M., Rafizah Mohd Hanifa, Mohd Shahrizal Sunar, & Nasir, F. M. (2018). *Simulation of Crowd Evacuation in Laboratory using Social Force Model*. 4173, 1–6. <https://doi.org/10.1109/isamsr.2018.8540545>

Hamizan, S., Roselina, S., Habibollah, H., Yusliza, Y., & Lizawati, M. Y. (2021). Crowd Emergency Evacuation Simulation Time Analysis via Obstacle Optimization Strategy. *Journal of Physics: Conference Series*, 2129(1), 012045. <https://doi.org/10.1088/1742-6596/2129/1/012045>

Liu, J., Jia, Y., Mao, T., & Wang, Z. (2022). Modeling and simulation analysis of crowd evacuation behavior under terrorist attack. *Physica A: Statistical Mechanics and Its Applications*, 604, 127891. <https://doi.org/10.1016/j.physa.2022.127891>

Tsai, P.-C., Chen, C.-M., & Chen, Y.-P. (n.d.). *PSO-Based Evacuation Simulation Framework*.

Retrieved February 4, 2025, from

<http://www.cmap.polytechnique.fr/~nikolaus.hansen/proceedings/2014/WCCI/CEC-2014/PROGRAM/E-14758.pdf>

Mahmood, I., Haris, M., & Sarjoughian, H. (2017). Analyzing Emergency Evacuation Strategies for Mass Gatherings using Crowd Simulation And Analysis framework. *Proceedings of the 2017 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*.
<https://doi.org/10.1145/3064911.3064924>

StayWise. (2024). *Hazard Detective Game* | StayWise. Staywise.co.uk.
<https://staywise.co.uk/public/resource/hazard-house-game>