Martin Stigen

Applying swarm to unknown and dynamic environments in evacuation planning

Master's thesis in Master of Science in Informatics
Supervisor: Pauline Catriona Haddow
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Martin Stigen

Applying swarm to unknown and dynamic environments in evacuation planning

Master's thesis in Master of Science in Informatics
Supervisor: Pauline Catriona Haddow
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

In the event of fires and other hazards in areas where people are located, their lives could be in real danger. Evacuation planning serves the important purpose of preserving these lives by facilitating for efficient and safe evacuations away from these areas.

While many of the current approaches to evacuation planning have been able to account for both areas becoming hazardous during the evacuation and the distribution of people, these models are dependant on global knowledge of the scenario, information gathering and communication. Given the dynamic nature of evacuations, where infrastructure could fail, these dependencies could instead become fatal for the models.

To overcome this challenge, this work presents a swarm model independent from any of this knowledge. The model is based on the boid behavior rules and uses only its perceptions of the local environment when making decisions. By following a set of simple interaction rules, this model is also capable of completing evacuations in environments with dynamic hazards.
Additionally, as a result of its independence from any knowledge of the scenario, it is also capable of completing evacuations in unknown environments with dynamic hazards, which neither of the previous models have accomplished.

Sammendrag

Hvis branner eller andre farer skulle oppstå i områder hvor mennesker befinner seg, kan livene deres være i stor fare. For å bevare deres liv og sikkerhet er det derfor viktig med gode evakueringsplaner slik at de kan evakueres trygt og effektivt bort fra disse områdene.

Det finnes allerede flere evakueringsløsninger som i sanntid tar hensyn til både farer i områdene og hvor folk befinner seg. Ulempen med disse modellene er at de baserer seg på mye kunnskap om områdene, informasjonsinnhenting og kommunikasjon. Dette kan gi fatale konsekvenser for modellene hvis noe av infrastrukturen skulle bli ødelagt under evakueringsprosessen.

For å løse disse utfordringene presenterer denne avhandlingen en svermmodell som er helt uavhengig fra den kunnskapen som de andre modellene krever. Den er basert på oppførselen til boid-reglene og bruker bare sine lokale oppfatninger av området når den gjør beslutninger. Gjennom enkle interaksjonsregler klarer også denne modellen å gjennomføre evakueringer i områder hvor farer kan oppstå dynamisk.
I tillegg, som et resultat av modellens uavhengighet fra kunnskap om områdene, klarer den også å gjennomføre evakueringer i ukjente miljøer med dynamiske farer, noe ingen av de tidligere modellene har gjennomført.

Preface

This thesis constitutes the work of a master's thesis in artificial intelligence written at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway, in the period of 12.08.2018 - 22.06.2019.

I would like to express my deepest gratitude to my supervisor Professor Pauline Haddow at the NTNU Department of Computer Science for excellent supervision and guidance throughout this entire process.
I would also like to thank my fellow students Eirik Baug and Andreas Norstein for the help and support that they offered during this work.

Martin Stigen

Trondheim, June 22, 2019

Contents

List of Figures

List of Tables

List of Abbreviations

ACO Ant colony optimization

AI artificial intelligence

DNF did not finish

GUI graphical user interface

NTNU Norwegian University of Science and Technology

PSO Particle swarm optimization

WSN wireless sensor network

Chapter 1

Introduction

This chapter presents an overview of the thesis and the motivation behind it, followed by the research goal and research questions. The research process is presented and finally an overview for the rest of the thesis structure is presented.

1.1 Background and Motivation

In the event of hazards, such as fires or infrastructure failures, in buildings or other areas where people are located, lives could be at risk. Good evacuations plans are critical for safe and efficient evacuations in order to guide people away from these areas. These could be serious dangers and if the evacuation plan is inefficient lives may be lost as a consequence.

The evacuation scenario is unpredictable and hazards could emerge in the environment even during the evacuation. To accommodate for this the evacuation plan should be able to adapt to these changes in the environment and be robust to infrastructure failure in order to still be efficient. This demands a lot from the evacuation plan and it is therefore an important field of study in order to improve the current solutions that are used today.

The current approaches to evacuation planning, presented in section 2.4, are able to monitor the environment in real-time, registering the areas with hazards and the distribution of people in the evacuation scenario. Based on this information, efficient and safe evacuation routes are established and made known to the evacuees through evacuation equipment installed the environment that is able to give dynamic guiding directions. The global knowledge of the environment represented in these models and the information gathering processes are what makes them perform so well, but also the drawbacks of using such models, where each model has to be designed specifically for each scenario.

The dependencies in these models motivated this work to investigate an alternative approach to evacuation planning, independent of environmental knowledge and information gathering. A model that would be general enough to be applied to unknown environments and be able to adapt its behavior in the presence of hazards. This led to explore how a swarm approach could be applied to evacuation planning, with the goal of keeping the model as simple as possible.

1.2 Goals and Research Questions

Goal *How can swarm agents be utilized to solve the challenges of unknown environments with dynamic hazards in evacuation planning?*

The goal of this work has been to investigate how swarm agents based on simple interaction rules can be used to improve evacuations in unknown environments where areas can become hazardous during the evacuation.

Simple interaction rules implies that the model has no knowledge of the environment at any point in time and that all decisions made by the swarm agent are only based on its perceptions of the local environment and the nearby swarm agents.

The goal emphasizes on that the global swarm behavior is the important factor in this swarm approach and that the individual agents do not have to follow advanced algorithms or use knowledge of the environment for the global behavior to be beneficial for evacuations.

Research question 1 *Can some global behavior beneficial to evacuations be obtained by applying the boid model?*

The boid model is a swarm technique inspired by flocking animals and is explained in section 2.3.2. It is applied in the swarm agent model presented in this work so that the swarm agents close to each other interact locally, with the goal of obtaining some global behavior that could be beneficial to solving evacuations.

Research question 2 *How well do the swarm agent approach generalize in different environments?*

Being able to generalize between different environments is an important property in evacuation planning where each scenario could be unique. The model is able to generalize if it is able to perform consistently well between diverse scenarios given different environments.

The scenarios constitute specific building layouts that vary in size and complexity, while the environment describes whether the scenario is unknown or has guidance equipment and if there are any hazards in the scenario.

Research question 3 *How do hazards complicate the behavior of the swarm agent model in guided and unknown environments?*

Hazardous areas are a challenge in both guided and unknown environments, restricting movement in the exposed areas. This question is directed towards how the swarm agents behavior is affected by hazardous areas in these environments and how swarms can overcome such hazards.

1.3 Research Method

The research method applied in this work has been an analytic process. Through the research process, the field of evacuation planning was explored. Advantages and shortcomings of the current approaches were analyzed and inspired by swarm techniques a new swarm agent model was created. To evaluate the proposed model several experiments were performed using a simulator that was created in this work. The tests were driven by the research questions and the results were analyzed and discussed with the final goal of answering the research goal of this work.

1.4 Research Process

The research process can be divided into three phases: the initial literature search, creating a structured literature review protocol and the structured literature review. The purpose of the initial literature search was to find a theme for the masters. After finding an interesting theme, a structured literature review protocol was formed to find the relevant literature. Finally the structured literature review process was used to review the literature and find application areas where techniques from bio-inspired artificial intelligence (AI) could be applied.

Initial Literature Search

The initial literature search involved finding a topic for the masters related to bio-inspired AI. In order to find literature relevant to the state of the art for bio-inspired methods, the latest articles published at the Genetic and Evolutionary Computation Conference and the IEEE Congress on Evolutionary Computation were reviewed.

There were many interesting articles proposing different techniques and application areas. One of these articles led to exploring the possibilities of using using graphics processing unit cards and parallel programming in combination with different methods from bio-inspired AI, such as evolutionary algorithms.

At the same time other articles from the conferences were reviewed to keep the scope open for other techniques and application areas as well. This led to finding an article that applied a swarm approach to evacuation planning [23]. As an application area evacuation planning serves the very important purpose of saving peoples life by guiding them to safety. It is also a complex problem to solve because there are many uncertainties and every scenario is unique. The importance and complexity of evacuation planning motivated to further research the topic through a structured literature review.

Structured Literature Review Protocol

A structured literature review protocol was formed to find the literature relevant to evacuation planning. To guide the literature search two research questions were defined:

- What are the current approaches to evacuation planning?
- How have bio-inspired techniques been applied to evacuation planning?

The first question focuses on exploring the state of the art within evacuation planning and the second on the approaches where bio-inspired methods have been used for evacuation planning. To answer these questions some search terms were defined for the literature search. The selected search engines to find publications were: Google Scholar, IEEE Xplore and ScienceDirect. To limit the amount of literature to review, some inclusion criteria and evaluation criteria were also defined. The inclusion criteria was used to sort out the literature with too little relevance to the research questions and the evaluation criteria was used to ensure the quality of the work. The search terms and criteria are found in table 1.1.

Search Terms	Evacuation planning, optimization, dynamic environments, hazards, real-time, swarm, ant colony optimization, evolutionary algorithms
Inclusion Criteria	• Literature should focus on evacuation planning. • Articles have to appear relevant after reading only the abstract and conclusion. • Literature should discuss strengths and weaknesses of the work presented, showing a critical reflection on the work.
Evaluation Criteria	• Results presented should be reproducible by following the description of the work or model. • When evaluating an approach or a model it should be compared to existing approaches or models when possible. • The work should clearly state any assumptions made and justify the design choices.

Table 1.1: Literature search terms and selection criteria

Structured Literature Review

The final phase of the research process included using the structured literature review protocol to find literature relevant for the work herein. The first step of this process was to explore different application areas in evacuation planning where bio-inspired techniques could be applied. Three relevant application areas were found: evacuation guidance planning, building layout planning and real-time evacuation planning. The application areas and bio-inspired techniques are illustrated in figure 1.1. The remainder of the literature review was used to evaluate the application areas and decide upon one for further work.

Evacuation guidance planning is the process of evaluating the evacuation area before an evacuation in order to plan evacuation routes and place the necessary equipment to display the suggested guidance directions.

Building layout planning is the process of evaluating different building layouts when a building is being planned in order to find the layout that best facilitates for evacuations.

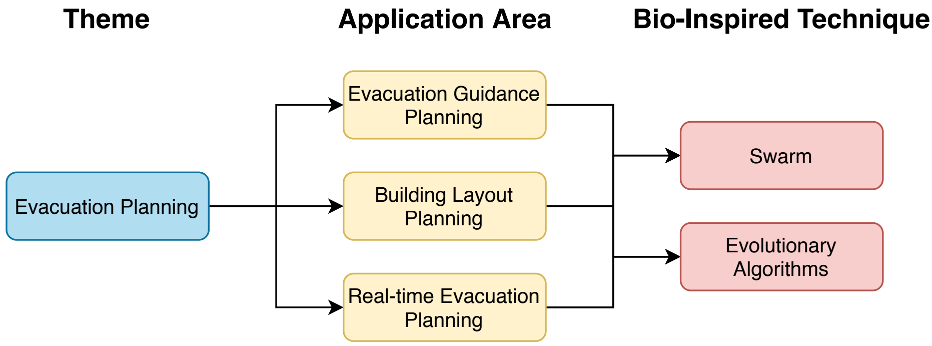


Figure 1.1: Structured literature review process

Real-time evacuation planning is the process of finding and giving evacuation directions during an evacuation.

In terms of order, the process of building layout planning happens before evacuation guidance planning, but here they are presented in the order they were explored in the literature review. From reading the literature is was evident that most of the approaches focused on optimization by reducing the total evacuation time, with the main difference being when this was done in the application areas.

Evacuation guidance planning and building layout planning are processes that are conducted before eventual evacuations. While building layout planning is used to evaluate the layout of the building itself, evacuation guidance planning attempts to find the best evacuation routes given a final layout. Both approaches require knowledge of the building layout in order to make their evaluations. For these application areas, both swarm and evolutionary techniques were found to be relevant methods that could be applied.
Real-time evacuations on the other hand are not bound by the same knowledge requirements. Instead the evacuation could be treated as a search process for finding the exits and safe areas without the need to know the building layout. This property motivated the direction of this work towards using real-time evacuation planning as the application area.

Swarm was found the most relevant technique to apply to this application area, well suited for both unknown and dynamic environments. In terms of evacuation planning it is important to be able to handle dynamic situations because areas could become hazardous during the evacuation.

1.5 Thesis Structure

The remainder of this thesis is structured as follows: Chapter two presentments the background information relevant to this work, the state of the art in evacuation planning and the motivation behind the model presented in this work. Chapter three describes the model decisions and the actual implementation of the model, followed by chapter four where the model is tested in order to answer the research questions asked in this work. Chapter five provides a discussion of the model connected to the research goal, followed by the contributions of this work and finally the further work is presented.

Chapter 2

Background Theory and Motivation

This chapter presents the required background theory on evacuation planning and the swarm model designed in this work, followed by the state of the art in evacuation planning. Finally, the motivation behind the swarm model is presented.

2.1 Evacuation Planning

In today's society buildings are becoming more complex to fit the demands of the ever-larger building requirements from a growing world population with an increasing density of settlements. However, the infrastructure for emergency evacuations has not been keeping up with this development, the most common precautions for evacuations being exit signs and emergency maps [22]. While these installations are based on expert knowledge and statistical data, they have shortcomings. Planning and installing the exit signs and emergency maps are generally performed when the building is being constructed, making it necessary to design the evacuation plans based upon the expected usage of the building. In reality the actual number of people and their distribution could deviate from these expectations. With too much difference between the expectations and the actual numbers, the planned escape routes could become inefficient and in the worst case lives could be lost.

The term evacuation can refer to several types of evacuations. There are the patient related evacuations that focuses on escorting the patient to a medical facility, such as the medical evacuation, the casualty evacuation and the casualty movement. Then there is the emergency evacuation, where there is an urgent and immediate need to guide people away from an area because of an imminent threat, an ongoing threat or a hazard to lives or property. The remainder of this thesis will be referring to the emergency evacuation when using the term evacuation.

There exists several reasons for needing an emergency evacuation away from an area. There are the natural disasters such as volcano eruptions, floods, earthquakes, tsunamis or wildfires that are caused by the natural processes of the Earth. Additionally, the other reasons for emergency evacuations include industrial accidents, traffic accidents, fires, military attacks, structural failure or viral outbreaks.

Evacuation plans are made to ensure the safest and most efficient evacuation of the evacuees away from such scenarios. Specifically they should provide some sort of guidance during the evacuation event. Evacuation planning is a complex problem involving many behavioral and environmental aspects, one of the most important factors being that the individuals do not have global knowledge of the evacuation situation. It is therefore essential that the evacuation guidance is clearly visible and easy to understand during an evacuation.

For an evacuation to be efficient it requires planning and preparation. The traditional approach can be divided into three steps:

Step 1: Gather statistical and expert knowledge of the area. This includes information about the scenario layout and the expected distribution of people in the areas.

Step 2: Plan escape routes using this knowledge so that the requirements of the expected scenarios are fulfilled, leading to efficient evacuations.

Step 3: Place guidance equipment in the environment, showing directions to the planned escape routes.

The most common equipment used in buildings to provide route guidance are exit signs and emergency maps [22]. Exit signs denote the route to the closest emergency exit and the emergency maps provide an overview of the building layout with the recommended escape routes, first-aid kits and fire extinguishers. The drawback of using this equipment is that it is unable to adapt to changes in the environment during the evacuation. It provides static guidance directions that could be inefficient given an unexpected distribution of people, or even worse, lead to blocked or hazardous areas.

2.1.1 Congestions

During an evacuation congestions occur when a path is chosen by more people than it can fit over a given time-period. The path becomes a bottleneck and the evacuees have to wait in turn to pass through. In figure 2.1 a congestion scenario is illustrated where two paths of equal size narrows down to one path that becomes a bottleneck for the evacuation flow.

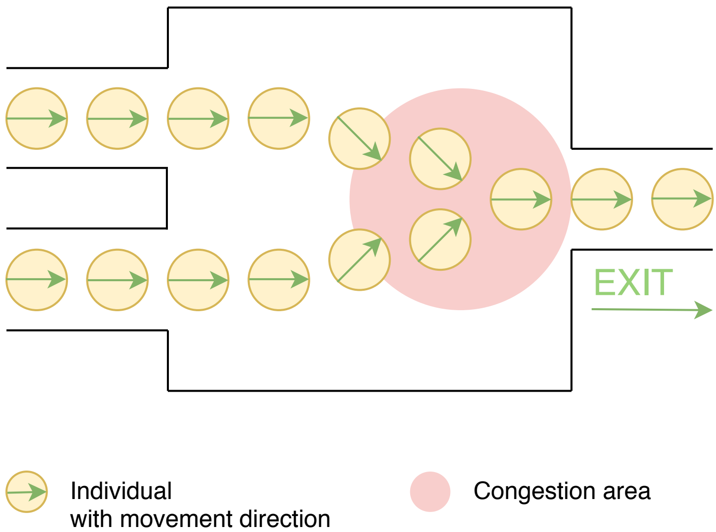


Figure 2.1: A congestion scenario where the exit path is the bottleneck

Congestions can cause serious delays in the evacuation process if people have to wait before they can get out, increasing the risk of their lives and it is therefore an important aspect to consider when planning evacuation routes.

2.1.2 Communication Networks

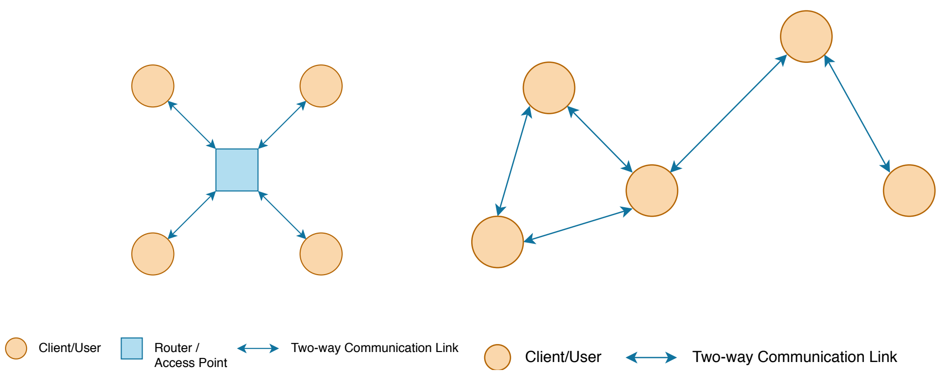
In addition to static guidance equipment, technology has made it possible to make guidance equipment that is able to give dynamic evacuation guidance. Sensors have also been made that are capable of monitoring the environment in real-time. Network structures are essential for communication and information sharing between these sensors and devices, forwarding information between them.

Ad-hoc Networks

An ad-hoc network is a type of wireless network. It it decentralized, meaning that the devices communicate directly with each other instead of going through some centralized network hardware, such as a router or access point, as shown in figure 2.2. This allows the devices to form networks where there is no pre-existing network structure. Another property of the ad-hoc network that makes it favorable for evacuation planning is the fact that is has no single point of failure since it has no infrastructure. If parts of the building infrastructure were to fail during an evacuation, the remaining parts of the ad-hoc network would still be able to operate.

As displayed in figure 2.2b not all clients in an ad-hoc network are able to directly communicate with each other because of communication reasons such as distance, walls etc. The clients are still able to communicate through multi-hop communication by forwarding the messages through the other clients in the network.

A mobile ad-hoc network is an ad-hoc network using mobile devices. It is possible to move the devices around independently of the other devices because the they are able to frequently change their connections with the other devices to stay in the network.



(a) Centralized network (b) Ad-hoc network

Figure 2.2: Network communication structures

Wireless Sensor Networks

A wireless sensor network (WSN) is a group of sensors that cooperatively monitor large physical environments and transmits the collected data wirelessly to each other or some base station [8]. A base station is a server used to receive the data.

Each sensor is able monitor and record some physical condition of the environment. Some examples are temperature, pressure, position, motion, vibration and radiation. It is also possible for a sensor to combine the sensing techniques in order to monitor multiple conditions at the same time.

The communication from the sensor to the base station can be both single-hop and multi-hop. Using single-hop each sensor transmits its collected data directly to the base station. With multi-hop the data is cooperatively transferred between sensors in order to reach the base station. The communication structure of multi-hops is similar to ad-hoc networks and it is possible to deploy a WSN using the ad-hoc structure.

2.1.3 Real-time Evacuation Planning

The traditional approach to evacuation planning has been to use static solutions and guidance equipment, implying that the evacuation routes are planned beforehand and do not change during the evacuation scenario. Real-time evacuation planning on the other hand is able to update the guiding directions based on the current evacuation scenario.

Real-time evacuation systems consist of sensors that monitor the environment and guidance equipment capable of displaying dynamic guiding directions. The system can be either centralized or distributed. Using a centralized system one node in the network will be assigned the role as the master node. All information gathered by the sensors is sent to the master node and the master node uses this knowledge to calculate the evacuation routes. Using a distributed approach, no master node is assigned. Instead the information gathered by the sensors is distributed amongst the nodes in the network so that each node calculates the evacuation routes independently.

Both system designs have strengths and weaknesses. While the centralized system is exposed to the single point of failure, which can be a critical failure in the evacuation scenario, it has the advantage of planning the evacuation routes based on the information from all the sensors and could therefore make optimal evacuation routes for all the evacuees. The distributed approach has the advantage of being robust to node failure, but being constrained to making local evacuation directions. This could be a problem if nodes in the distributed system receives information at different times and as a result makes different guidance decisions.

2.1.4 Evacuation Models

Planning evacuation routes require knowledge of the environment and the evacuation scenario. Evacuation models are used to capture this knowledge and model the evacuees movement over time in order to find good routes. The more knowledge is available, the more detailed and accurate the models can be. The models can be classified as either macroscopic or microscopic.

Macroscopic

Macroscopic models do not consider any individual behavior. Instead it models the evacuation from a global perspective, viewing the evacuees as groups with homogeneous characteristics such as behavior, speed and size. The models are able to find optimal evacuation times that be used to analyze how good existing building structures are, or to help plan new building layouts. The optimal times can be used as good lower-bounds for the expected evacuation time.

The macroscopic model uses a graph G = (V, E) to model the evacuation environment [4], where the vertices (V) represent the different areas where people are usually gathered and the edges (E) represent the paths between the areas. Each vertex $v \in V$ is associated with a capacity limit and the current number of individuals in the area and each edge $e \in E$ is weighted with a traveling cost such as capacity and length.

Microscopic

Microscopic models consider the characteristics of each individual and how their movement is affected by interaction with each other. The purpose of this model is to describe the evacuation process as realistically as possible. This model is therefore often used as a basis for simulators to simulate the evacuation processes and asses eventual weaknesses of the evacuation process or to evaluate any changes made to improve the process.

A microscopic model can be either discrete or continuous. The discrete model divides the environment into a grid or hexagonal patches using discrete time steps to model the flow of time. For each time step the individuals are able to move between the patches, as long at their movement follows the rules of the model. The continuous model do not divide the environment into patches. Instead the environment is continuous and movement can be viewed as a flow determined by the forces that act on the evacuees, such as their velocity.

2.2 Agents and Environments

All swarm approaches are based on some type of agents operating in a given environment. This section covers some basic agent and environment definitions as well as the relevant architecture theories used in this work.

2.2.1 Agents

In the context of AI the agent is defined as anything that perceives its environment through sensors and acts upon it through actuators [27]. The definition of the agent is quite generic, not excluding any behaviors or levels of intelligence. The relationship between the agent and the environment is illustrated in figure 2.3. The yellow question-mark box inside the agent represents the agent function. This function defines how the agent behaves given its knowledge of the environment. Technically speaking it can be seen as an input-output mapping, given a sequence of percepts the agent function maps this to an action. The specific implementation of the agent function is called an agent program [27].

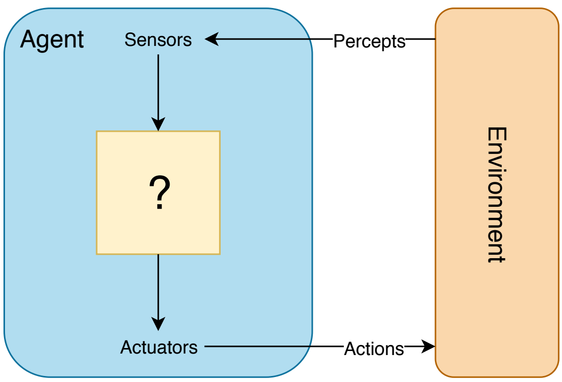


Figure 2.3: The agent-environment relationship (adapted from [27])

Local and Global Information

The agent uses its sensors to perceive the environment. How much information the agent has access to about the environment is therefore dependent on what sensors the agent has available.

Local information represents a limited view of the environment. At any given point in time the agent is able to perceive the environment, but only parts of it. With a limited view of the environment the agent must move around in order to perceive more of the environment. Limits to sensor range, obstacles blocking the sensors or closed environments, such as being inside rooms and buildings, could be reasons for the agent to have a limited view of the environment.

If the agent has access to the complete state of the environment at each point in time, it has access to the global information about the scenario. In context of evacuations this would imply that the agent at all times knows the scenario layout, the hazardous areas and the locations of other evacuees.

Control Systems

The agent program can be represented by a control system. Control systems are used to decompose the behavior functions of the agent program into modules. The advantage of this module partitioning is an increased control over the individual behaviors when the modules are combined.

In [2], Brooks presents a module combination technique called subsumption architecture, displayed in figure 2.4. This architecture uses multiple layers of control where each layer is responsible for some control behavior. The lower layers represents the basic behaviors and are unaware of the higher layers. More sophisticated behavior is implemented in the higher layers and when needed the output from higher layers are also able to suppress the output from the lower layers.

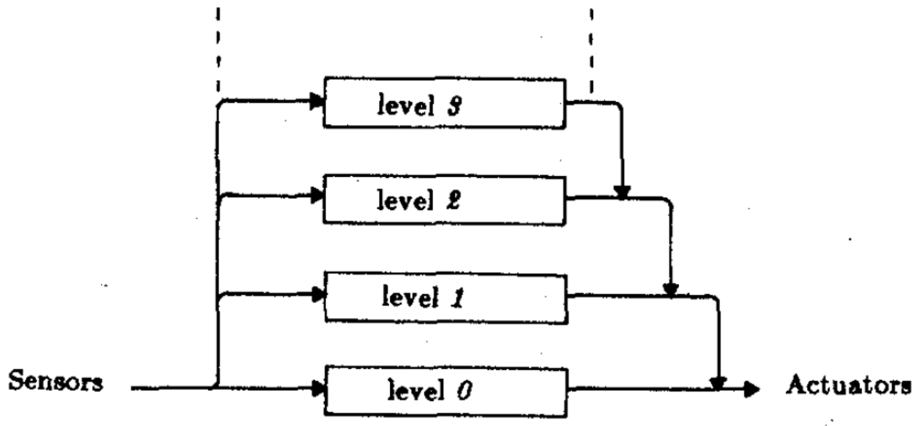


Figure 2.4: Brooks' subsumption architecture (adapted from [2])

2.2.2 Environments

During an evacuation the agent has to navigate through some scenario to reach the exit. How much knowledge the agent has about the environment, the guidance information available and how it changes over time are different attributes of the environment.

Known vs Unknown

If the environment is known, the agent has knowledge of the scenario layout, while if the environment is unknown, the agent has no knowledge of what the scenario looks like.

Guided vs Unguided

In the context of evacuation planning, guided environments have guidance equipment placed in the scenario, giving guidance directions toward the exits. In the unguided environment there is no presence of any guidance equipment.

Static vs Dynamic

In static environments the scenario do not change over time. Dynamic environments, on the other hand, can change over time independent of the agents. Examples of dynamic environments are scenarios where areas become hazardous during the evacuation or paths that become blocked because of structural collapses.

2.3 Collective Systems

Collective systems are systems where individuals are able to communicate and cooperate to achieve some goal. By working together they are able to obtain some global behavior that otherwise would not be possible. When the resulting global behavior of the individuals is said to be more than the sum of the individuals alone, it is said to be emergent [11]. This is a phenomena unique to collective systems that individuals operating in isolation alone cannot achieve. Swarm approaches are based on collective systems.

Common for swarm approaches is that the individuals operate on local information without global knowledge of the environment. Swarm approaches are therefore well suited for search processes in unknown environments.

2.3.1 Swarm Intelligence

Swarm intelligence is a special case of collective systems. The term was first introduced by Beni and Wang in [1]. In accordance with their definition it requires a system of non-intelligent robots exhibiting collectively intelligent behavior. In [11] it is further defined as large collections of simple agents that are able to solve tasks too complex for a single agent to solve, or that can display the robustness and adaptability to environmental variation displayed by biological agents.

2.3.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a swarm technique where candidate solutions to a problem are represented as positions in a search space and moved around using velocities. It is an iterative search process where each particle's velocity is determined by its locally best known position and the globally best known positions. By continuously updating the particles positions and velocities, they are collectively searching for the optimal solution to a problem.

Boids

The boid model was first introduced in [24] as an approach to simulate the flocking behavior of birds, fish and other animals. The term boid is a reference to bird-like objects, but it used as a generic term to represent the flocking behavior from other creatures as well. The boid model is considered a particle swarm approach because each individual is represented with a position and a velocity.

The collective behavior of flocks has several properties that makes it interesting. Even though a flock is made up of independent individuals, their overall motion is seemingly fluid and complex. This gives the impression of a centralized control even though it is just an aggregation of the individuals behavior based on their local perception of the world.

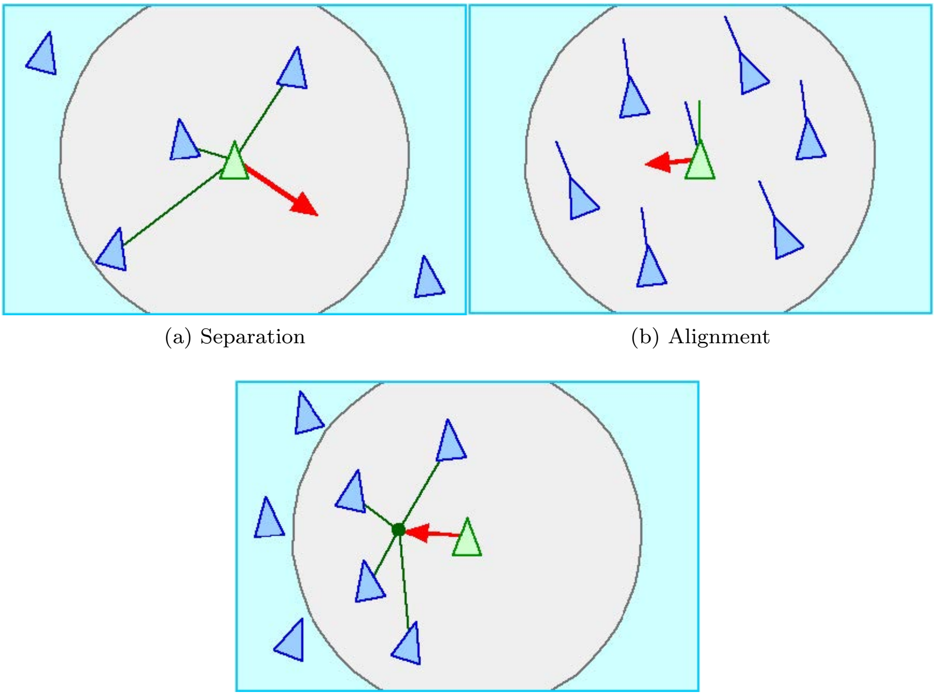
The boid model identifies three different behavior rules, that when combined can be used to simulate flock-like behavior: separation, alignment and cohesion. The rules are based upon on a limited and localized view of the world. To simulate the localized view a radius around the boid is used to limit its perceived flock mates, where the other flock mates outside this radius is disregarded when making decisions.

Separation focuses on keeping a certain distance from the rest of the flock, so that collisions with nearby flock mates are avoided. It is obtained by calculating a repulsive force from each nearby flock mate within the given radius and combining them. The repulsive force becomes greater the closer the flock mate is.

Alignment focuses on keeping the same velocity as the other flock mates. Velocity refers to the combination of speed and direction of motion. It is obtained by averaging the perceived flock mates velocity within the given radius.

Cohesion focuses on staying with the flock by moving towards the flock mates. It is obtained by finding the average location of the flock mates within the radius and moving towards this location.

As illustrated in figure 2.5 each rule produces a desired velocity, represented by the red arrows. The combination of these velocities forms the final velocity that is used to update the boid's next movement. How these rules are implemented and weighted plays a significant role in the final behavior of the boid [24].



(a) Separation

(b) Alignment

(c) Cohesion

Figure 2.5: The boid behavior rules (adapted from [25])

2.3.3 Ant Colony Optimization

Ant colony optimization (ACO) is another swarm approach, inspired by ants. It is a probabilistic technique where the agents explore the environment for a solution to a problem and leave pheromone trails on their paths. It it an iterative search process where the agent's selection of the paths are influenced by the other pheromone trails and individual exploration. By marking the paths of the locally best solutions with additional pheromones, the swarm will converge its search towards better and hopefully the global best solution(s).

2.4 State of the art

Evacuation planning is not a simple problem to solve. As elaborated in section 2.1 it is a complex problem with several factors to take into consideration, some even being unknown in advance. There exists several different approaches to solve the evacuation problem, varying from mathematical and statistical modelling to applying methods from artificial intelligence. The methods use the macroscopic model, the microscopic model or even both, see section 2.1.4.

2.4.1 Macroscopic Models

The macroscopic models used in evacuation planning are based on mathematical approaches where the scenario is represented as a graph $G = (V, E)$. In mathematics, graph theory is a well known field of study with several established algorithms known to find optimal solutions to different graph problems.

Common macroscopic models applied to evacuation planning are static networks and discrete time dynamic networks [15]. Networks are used to find the best movement flows in the graphs. This information can in turn be used to evaluate the scenario layout and to design efficient evacuation routes. While some of the approaches listed here are old, they are still relevant to the state of the art because their model achieves optimal results to the problem they are addressing in evacuation planning.

Static Networks

The static network problems are used to find exactly one route from each area to an exit. Examples of such models applied to evacuation planning are:

- **Shortest path**: Applied in [9, 16] to find the shortest paths from the areas in the scenario to the exits, considering only the total evacuation time of the paths.

- **Quickest path**: Applied in [7, 5, 19, 26] to find the quickest paths from the areas in the scenario to the exits. The quickest path problem is an extension to the shortest path problem where the amount of people is also considered when finding the paths, to avoid congestions.

Discrete Time Dynamic Networks

The discrete time dynamic network is a time expansion of the static network. By expanding the network into a predetermined set of time periods, the model is able to increase its accuracy on the estimated flow of the evacuees. This allows the model to find several routes from each area to an exit instead of just one. The downside of using these networks is the computational cost required when expanding the networks [21].

In [20] a simulator based on this network model was developed, able to to determine evacuation times and possible bottleneck locations in building scenarios. Another advantage with this approach, presented in [3, 10], is that the quickest path problem can be improved by allowing multiple paths from each area to the exits instead of just one. In [12] this model is used to maximize the number of evacuated individuals in a certain time period and [13, 18] applies it to maximize the number of evacuated individuals in each time step.

2.4.2 Microscopic Models

The microscopic models are not limited to the graph representation used in the macroscopic models. This implies that they do not require the same amount of knowledge about the environment as the creation of the graph requires.

Wireless Sensor Networks

As described in section 2.1.2 a WSN is a group of sensors able to monitor and record the physical conditions of the environment. By monitoring the environment in real-time it is possible to determine the actual distribution of people and hazardous areas and use this information when evaluating evacuation paths.

In [6] a framework based on a WSN for emergency guiding is suggested. The approach has several wireless sensors distributed in a building forming a multihop ad hoc network. The sensors are also used to display guidance directions for the people located in the area. Their framework uses a centralized algorithm where one sensor is chosen as the sink node and all the information flows towards this node as shown in figure 2.6. In turn this information is forwarded to the control host where their load balancing algorithm from emergency guiding is run. The algorithm is run on a graph $G = (V, E)$, making the approach a hybrid of the macroscopic and microscopic models. The approach is also microscopic because it simulates that the individuals move through the scenario and observe the guidance directions.

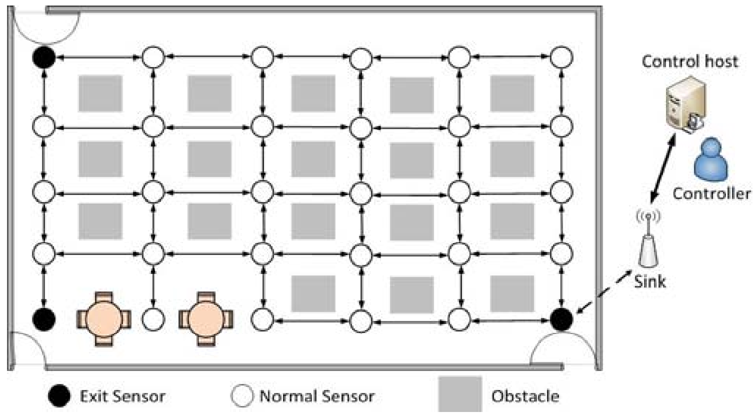


Figure 2.6: A centralized sensor network (adapted from [6])

By utilizing a WSN Chen et al. is able to account for both hazards and the distribution of people when evaluating guidance directions, resulting in safe and efficient evacuation routes updated in real-time.

There are however some shortcomings to this approach, the most critical one being the single point of failure. Should the sink node, control host or communication network at some point fail the entire system would stop working as a result. Another downside with this approach is that the sensors can only give one guidance direction at a time, making the the network-flow problem NP-hard [28].

In [28] a similar approach is suggested, but each evacuee is instead given personal guidance direction through handheld devices and it is show that the network-flow problem is no longer NP-hard. This is achieved by the sensors sending out individual guidance directions to each device in its proximity range.

Another model advantage in this work is that each sensor is able to calculate the guidance directions themselves, making the approach distributed instead of centralized. This solves the single point of failure that centralized systems have. The downside of the distributed approach is that it is hard to coordinate the guiding directions given from each sensor when they are decided locally.

Swarm Approaches

Swarm techniques have also been applied to find safe and efficient evacuation routes. In [14] an ACO approach is applied to find near optimal escape plans, considering both environments with dynamic hazards and congestion avoidance. Their results show that the approach is quick to converge towards the optimal solutions without the computational cost of using a time-expanded network. The suggested approach uses a graph to model the scenario and the capacity constraints, implying that it is a hybrid approach of the macroscopic and microscopic models.

In [23] another hybrid model is suggested. As with the other macroscopic models it presumes that the evacuation graph is available, as illustrated in figure 2.7. The approach is based on the concept of swarm intelligence, see section 2.3.1, where the swarm agents are simulated in a microscopic model using a discrete environment. The behavior of their swarm agents can be divided into two steps: information collection and route planning. These steps are repeated until the evacuation is complete. In the first step the agents have knowledge of their own position, but not the others. The whereabouts of the other agents are obtained by broadcasting their own positions and forwarding received positions in a multi-hop ad hoc network. In the second step each agents runs an algorithm that based on the current distribution of the individuals generates an evacuation plan. While this approach is able to utilize local communication to gather information about the evacuation situation it does also rely on a macroscopic model for the decisions of the swarm agent.

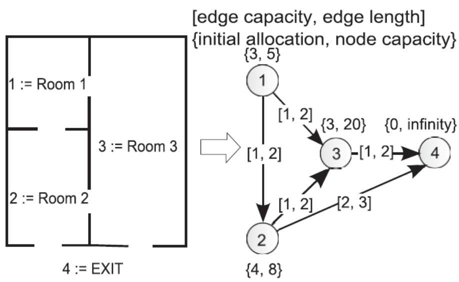


Figure 2.7: A building and its equivalent evacuation graph (adapted from Merkel et al. [23])

Finally, in [17] a PSO approach that only uses a microscopic model is presented. The approach is used to estimate evacuation times when pedestrians evacuate from one large area using one or more exits. The environment is unknown to the pedestrians and they are able to escape the area by updating their velocities with their local and global best known positions and moving accordingly. However, this approach do not account for more than one area and no hazards are involved.

2.5 Motivation

This section explains the motivation behind the chosen swarm model in the context of the research goal, the background theory and the work on the field.

As elaborated in section 2.1 the traditional approach to evacuation planning has been to gather information about the environment and use this knowledge to design the evacuation models. In the later years new technology has made it possible to monitor the environment in real-time and broadcasting this information through the environment by using communication networks, leading to the creation of models capable of dynamic evacuation planning. The advantages of these models have been multiple. As detailed in section 2.4, these models are able to give dynamic evacuation guidance and can avoid congestions to some extent by directing the evacuees in different directions. Being able to detect hazards in the environment has also made it possible to make the guidance directions safe as well.

While these models have been able to improve the evacuation problem with regards to the total evacuation time and safer routes, they are extremely dependent on gathering information about the environment, knowing the area layout and the installation of the necessary sensor and network equipment. Given the dynamic nature of evacuation environments it could be fatal for the evacuation system to depend on building infrastructure in order to operate. Thus the greatest advantage of these models, gathering and using information, is also their weakness if the infrastructure were to fail. Additionally, in the context of unknown environments, this information is not available and such models would therefore be inadequate in these environments.

Given the ability to operate in dynamic and unknown environments, existing swarm approaches applied to evacuation planning were evaluated, as elaborated in section 2.4.2. While some of them were able to complete evacuations in dynamic environments these models required global knowledge of the scenario. The PSO approach presented in [17], was able to complete evacuations in unknown environments, but the scenario did not cover more than one area or any hazards.

The swarm model proposed in this work was based on the knowledge gained from this research and the lack of a model that was able to complete evacuations in unknown environments with dynamic hazards.

Chapter 3

Model

This chapter presents the proposed swarm model along with its design choices and the simulator that was created to simulate the swarm agents in different evacuation environments.

3.1 Swarm Model

The proposed swarm model was inspired by existing swarm techniques and how they could be combined to overcome the challenges of unknown and dynamic environments for real-time evacuations. What makes this model so unique compared to the existing approaches is its simplicity. The suggested swarm model has no memory or knowledge of the environment and uses only perceptions of its local environment to navigate around, following a set of simple behavior rules. The design choices of the model were inspired by the concept of swarm intelligence, see section 2.3.1. The goal was to keep the swarm agent as simple as possible while still being able to complete the evacuation scenarios.

3.1.1 Model Knowledge

The model is completely independent from any knowledge regarding the scenario layout. Nor has it any memory of where it has been or what it has observed. The only information it has available is its current perceptions of the local environment. The perceptions represents a limited view of the scenario and the nearby swarm agents. This design choice is a major difference from the work presented in the state of the art section, where the models are filled with knowledge that they depend on.

Given a building scenario with different areas, the swarm agent is able to perceive the area it currently is located in and the connected hallways. The perceptions include if any of these areas are hazardous. This observation is illustrated in figure 3.1. While it is able to perceive the area, it has no notion of the scenario layout or where the area it observes is located in the scenario. It is also able to perceive any guidance equipment placed on the hallways connected to the area it is located in.

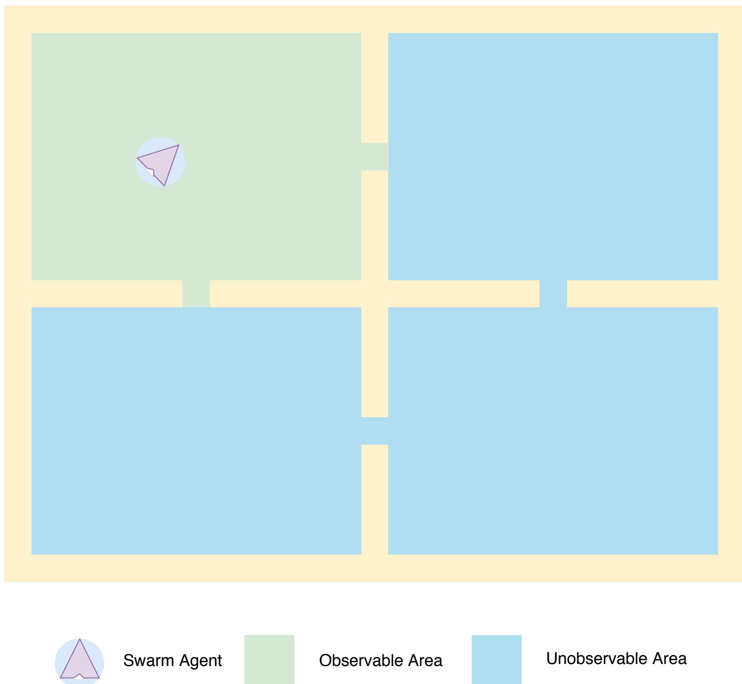


Figure 3.1: The swarm agent's perceptions of its local area

In addition to its perceptions of the scenario, the swarm agent is able to perceive other swarm agents in its *proximity range*. An example of the swarm agents proximity range is illustrated in figure 3.2. The proximity range was inspired by the boid model presented in section 2.3.2 and plays a major part of the swarm interaction introduced in section 3.1.2.

The proximity range represents a more limited view of the environment than the swarm agent's perceptions of the scenario. This was a design choice based on that the scenario area was thought to be observable when the swarm agent was inside it, but that the neighbourhood of the swarm agent needed to be more restricted for the swarm interaction.

Figure 3.2: The swarm agent's perceptions of the nearby swarm agents

3.1.2 Behavior Rules

The basic behavior of the swarm agent is to move forward using a constant velocity. For the swarm agent to be able to explore the scenario, it became evident that its basic behavior needed to include some evacuation objective to guide its movement. The evacuation objective used in this model is a unique feature of this work. It is based on a simple exploration behavior: given the perceived hallways connected to the area it is currently located in, move towards one of them. An example of the evacuation objective is given in figure 3.3. In the figure the swarm agent observes the two hallways as potential evacuation objective goals.

The evacuation objective of the swarm agent then selects one of the hallways as the objective goal. If a hallway is perceived as hazardous it is ignored from this selection. This implies that evacuation objective is passively avoiding hazardous areas by not selecting them. The passive hazard avoidance was found to work well for the model, so other alternatives were not explored. When selecting among

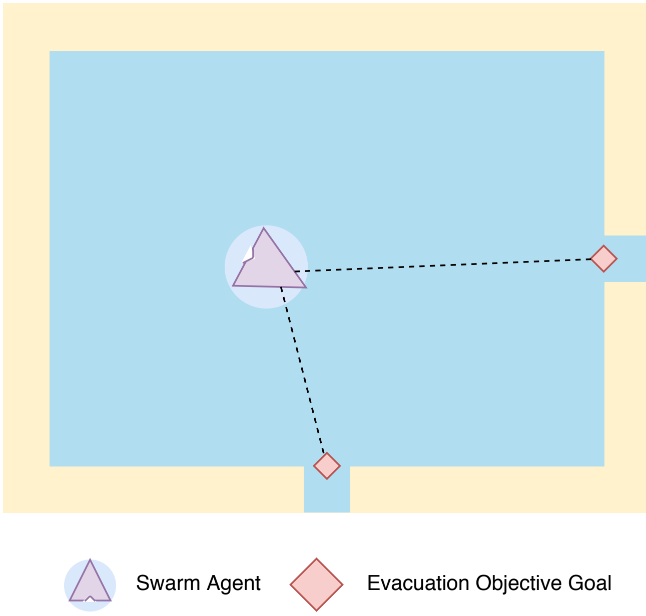


Figure 3.3: The swarm agent observing two hallways related to the evacuation objective

the non-hazardous hallways there are some rules that determines which one to choose in prioritized order:

1. **Exits**: Hallways perceived as an exit out of the scenario are always selected.

2. **Guidance Equipment**: Hallways marked with guidance equipment have a **guidance priority chance** to be selected. This rule is only used in guided environments and is ignored when the environment is unknown.

3. **Best Direction**: The hallway that requires the least change in the swarm agent's direction is selected, but could be ignored given a **ignore room chance**.

The first rule is straight forward, if a hallway leading to an exit is observed, move towards it.
The second rule is based on that guided environments represents some global knowledge of the environment that the swarm agent itself is unable to obtain. The **guidance priority chance** allows the model to adjust the chance of prioritizing the hallways marked with guidance directions over the other hallways.

The third rule keeps the movement of the swarm agent consistent, moving forward towards the next hallway.

Other rules for the hallway selection, such as random selection and closest hallway were considered, but were found to be inefficient. A random selection caused the swarm agent to constantly switch between the possible hallways, spinning in place, while selecting the closest hallway resulted in the swarm agent getting stuck by just moving back and forth through the same hallway.
The **ignore room chance** was introduced to the third rule inspired by the random selection rule, to allow more variance in the swarm agent's movement and not get stuck by always preferring the same hallways.

In addition to the evacuation objective, the boid behavior rules, **separation**, **alignment** and **cohesion** were selected for the swarm behavior, see section 2.3.2. The boid rules where selected for three reasons:

1. The rule requirements matched this model and could be implemented using only the swarm agent's perceptions of the local environment.
2. The rules are considered simple, matching the design goal of this model.
3. The flocking behavior obtained by combining the rules could be beneficial for the swarm agents in terms of evacuations.

Reason number three was based on the following assumptions:

- When multiple swarm agents attempts to go through a hallway from both sides, one of the sides will align with the other and the flow will go in one direction instead. This would reduce potential congestion problems.
- When a larger group of swarm agents have explored an area and are moving away from it, smaller groups will join the larger group instead of exploring the same area if they meet the larger group. This would reduce the total evacuation time as a result of less individual exploration in the environment.

Each boid rule is described in section 2.3.2, their implementation however is model specific. The following descriptions of the rules explain the individual implementations used in this model.

Movement

To explain the implementation of the boid rules, it is important to understand how the movement of the swarm agent is calculated.

The swarm agent's movement is represented as a vector $\vec{v} = \langle v_x, v_y \rangle$. The vector presents the agent's velocity, which is equivalent to its speed and direction of motion. The velocity vector is illustrated in figure 3.4 as the red arrow.

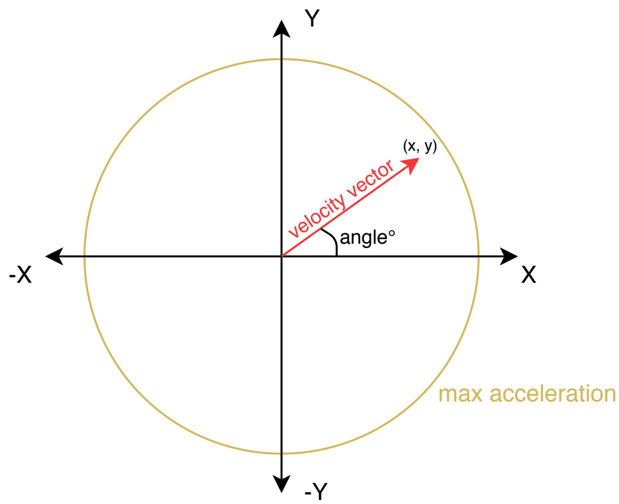


Figure 3.4: The velocity vector used to model the swarm agent's movement

To calculate the speed of the swarm agent, the vector's length, also known as magnitude, is calculated as in equation 3.1. The parameter **max acceleration** is used to limit the agent's maximum speed.

$$speed = \sqrt{{v_x}^2 + {v_y}^2} \tag{3.1}$$

Following this model, both the evacuation objective and the boid rules each produce their own velocity vector, representing the wanted changes in the swarm agent's movement.

Separation Rule

The implementation of the separation behavior rule was inspired by Reynolds' approach in [25]. The pseudo code for the implementation is illustrated in algorithm 1. The explanation for the variables used in the pseudo code of the implementations are included in table 3.1.

Algorithm 1: Pseudo code for the separation implementation

```
separationVelocity = new Vector();
for agent in agents do
    velocity = new Vector(this.position - agent.position);
    length = velocity.magnitude();
    velocity = velocity.normalize().multiply(1.0/length);
    separationVelocity.add(velocity);
end
return separationVelocity;
```

Variable	Explanation
this	The swarm agent
agents	A list of the swarm agents within its proximity radius
agentCount	Number of agents in the agents list

Table 3.1: Explanation of the variables used in the algorithm pseudo codes

Following Reynolds' approach there is an individual weighting of the nearby agents given their distance from the current agent. This means that the individual separation velocity is higher for an agent that is close than one further away. The individual velocities added together become the final separation velocity that is returned to the behavior system. During the implementation of the separation rule in this model it seemed relevant to use the same weighting because the swarm agents that are close should be weighted higher to obtain a balanced separation distance between the swarm agents.

Alignment Rule

The alignment velocity is calculated by averaging the velocities of the swarm agents within its proximity radius. The pseudo code for the implementation is shown in algorithm 2. This function was implemented following the descriptions of the alignment behavior rule in [24].

Algorithm 2: Pseudo code for the alignment implementation

```
alignmentVelocity = new Vector();
for agent in agents do
    alignmentVelocity.add(agent.velocity);
end
alignmentVelocity.divide(agentCount);
return alignmentVelocity;
```

Cohesion Rule

As with the alignment implementation, the cohesion rule was also implemented following the description in [24]. The cohesion velocity is calculated by averaging the other agents positions within its proximity radius and setting the cohesion velocity to point towards the average position.

Algorithm 3: Pseudo code for the cohesion implementation

```
averagePosition = new Position();
for agent in agents do
    averagePosition.add(agent.position);
end
averagePosition.divide(agentCount);
cohesionVelocity = new Vector(this.position - averagePosition);
return cohesionVelocity;
```

3.1.3 Collision Avoidance

In addition to the behavior rules, collision avoidance was introduced to the model in order to avoid collisions with other swarm agents whenever possible. In [24] two techniques were presented for collision avoidance: steer-to-avoid and a force-field concept. In the force-field concept all objects emit a repulsive force that increases when the boids get closer to the object, making the boid steer away from the object. Using steer-to-avoid the boid uses its perceptions to see the obstacles in front of it and correct its movement if on a collision course. Being able to use the perceptions of the swarm agent the steer-to-avoid technique was adapted for this model.

The swarm agent uses the steer-to-avoid technique to avoid moving into other agents by trying to move around them instead. After the velocity of the swarm agent is calculated, a check is made whether its next move using this velocity will cause a collision with another swarm agent. If a collision is assumed, the collision avoidance alters the velocity of the swarm agent to move around the other swarm agent instead. In figure 3.5 a swarm agent is on a collision course with another swarm agent. The collision avoidance detects this and alters its velocity.

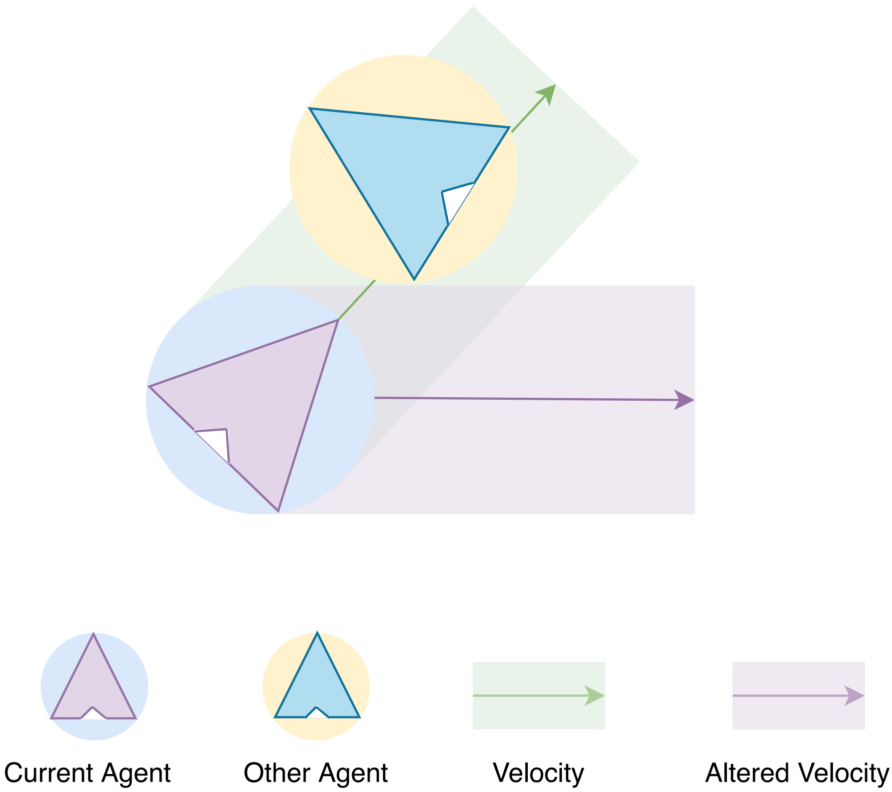


Figure 3.5: Illustration of the collision avoidance adjusting the swarm agent's velocity

While the steer-to-avoid concept was inspired by [24], Reynolds did not provide an implementation description, so the implementation presented here is therefore unique to this work.

When a collision is assumed to happen the calculation of the altered velocity can be described in three steps:

Step 1: Calculate the difference between the next position of the swarm agent and the colliding swarm agent.

Step 2: Calculate the remaining distance needed to avoid the collision.

Step 3: Alter the velocity based on the remaining distance value.

Figure 3.6 illustrates this calculation. First, the next position of the swarm agent is calculated to be inside the radius of another swarm agent, marked by the arrow. The distance from the other swarm agent's center is then calculated, marked by the green line. By subtracting this distance from the swarm agents radius, the remaining distance is be calculated, marked by the red line.
By adding the remaining distance to the swarm agent's velocity, its next position should be outside the other swarm agent's radius and the collision is avoided, marked by the second arrow. This approach does not account for the other swarm agents velocities, meaning that it does not check where the other agents are headed before calculating collisions and avoidance measures. The reasons for this design choice were that there is an uncertainty whether the other swarm agents keep their same velocities and that the boid behavior rules already reflects on the other agents velocity.

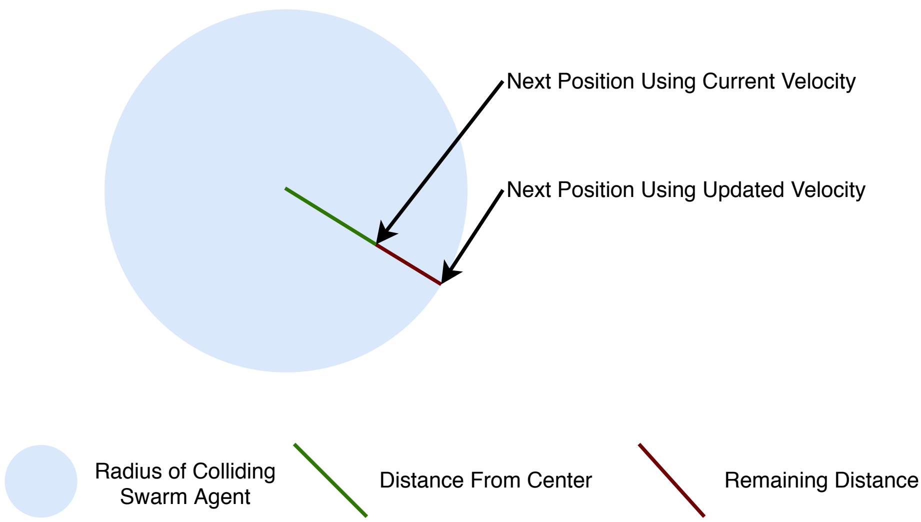


Figure 3.6: Collision avoidance calculation of the velocity adjustment

3.1.4 Combining the Behavior Rules and Collision Avoidance

So far all the behavior rules and the collision avoidance used in the model has been explained, but not how they are combined. For this purpose a behavior system was created, inspired by Brooks' subsumption architecture presented in section 2.2.1. The behavior system is responsible for combining the different behaviors and allowing some behaviors to override others. Following this architecture, dividing the behaviors into three different layers of control were found convenient, as illustrated in figure 3.7.

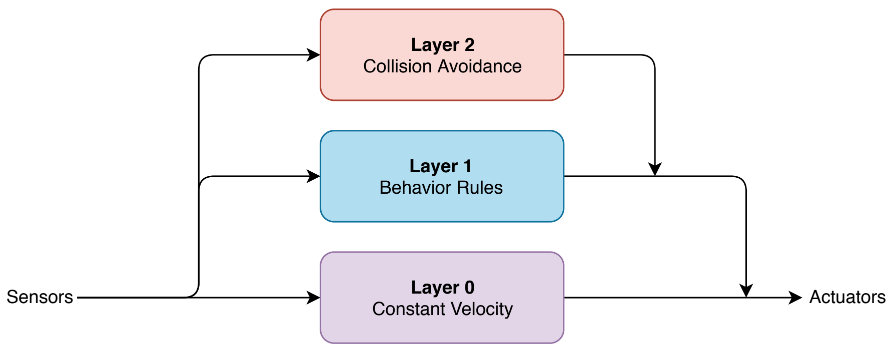


Figure 3.7: The three control layers used in the behavior system

Layer 0 corresponds to the lowest level of competence and it simply keeps the swarm agent moving in the same direction using its previous velocity. Layer 1 is more sophisticated, changing the velocity of the swarm agent by following the behavior rules. The final layer, layer 2, is responsible for collision avoidance. Using this architecture the swarm agent was able to move in the direction of the behavior rules and the collision avoidance layer would override this movement if corrections were necessary.

When combining the behavior rules in layer 1, two approaches were evaluated. In [24] a technique called prioritized acceleration allocation is used. Each behavior has a given priority and adds their velocity in turn to the final velocity. The final velocity is complete when all velocities have been added or when the velocity reaches a fixed magnitude, disregarding the remaining velocities.
The other approach presented in [25] uses a linear combination of the velocities. Each velocity is given a weight and are then combined to the final velocity.
The linear combination was selected for this model because it did not require a strict priority of the behaviors. The relationship between the velocities was

also easy to adjust by using the weights. The four weights used in this model are: **evacuation objective weight**, **separation weight**, **alignment weight** and **cohesion weight**. The weighting of the individual velocities are shown in equations 3.2, 3.3, 3.4 and 3.5.

$$evacuationVel = evacuationObjectiveVelocity \cdot evacuationObjectiveWeight \tag{3.2}$$

$$separationVel = separationVelocity \cdot separationWeight \tag{3.3}$$

$$alignmentVel = alignmentVelocity \cdot alignmentWeight \tag{3.4}$$

$$cohesionVel = cohesionVelocity \cdot cohesionWeight \tag{3.5}$$

The velocities are then combined as shown in equation 3.6. If the magnitude of the final velocity is greater than **max acceleration** it is trimmed down to match it instead.

$$finalVelocity = evacuationVel + separationVel + alignmentVel + cohesionVel \tag{3.6}$$

The final velocity is then evaluated by the collision avoidance layer to check whether this velocity will result in a collision and adjust it accordingly if this is the case.

A stagnation avoidance behavior was also applied to layer 2. It held a count on the number of times the swarm agent had attempted to update its movement according to the velocity, but collided into other swarm agents or walls instead. If this number reached a **stuck count limit**, the final velocity was changed to a random velocity in an attempt to unstuck the swarm agent from its current position. The stuck count limit was set to zero each time the swarm agent was success full in updating its movement.

3.2 Simulator

To simulate the swarm agents in different evacuation scenarios a simulator was created. The simulator can be divided into four modules: the simulator core, the graphical user interface (GUI), the scenario interface and the agent interface. The simulator was designed as a modular system using interfaces for the scenarios and agents to support multiple scenario and agent implementations. It is common to use interfaces when there can be more than one implementation of the object. As an example, the scenario interface defines the common variables and functions required for a scenario, but does not specify a specific scenario itself. An overview of the system's modules and their main interactions are illustrated in figure 3.8.

As illustrated in the figure each simulation starts at the graphical user interface. The interface allows the user to select between the simulation parameters, the scenario and the weights of the swarm agent's behavior rules. These parameters are then sent to the simulator core where the specified scenario and agent type are initialized. The scenario interface and agent interface are independent of the simulator core for the system to be flexible and facilitate for later additions to the simulator without too many dependencies.

The simulation run is then started. A simulation run has one or more iterations. Each iteration is one complete simulation of an evacuation scenario. The iterations are independent of each other, but share the same simulation parameters. Because the initial placement of the swarm agents and their initial velocities are chosen at random at the beginning of each simulation, each iteration will be different. The purpose of allowing several iterations per simulation run, instead of just one, was to allow the tests to gather more data about the parameters being tested. During each simulation iteration the simulation is displayed in the GUI, giving a visualization of the swarm agents in the given scenario.

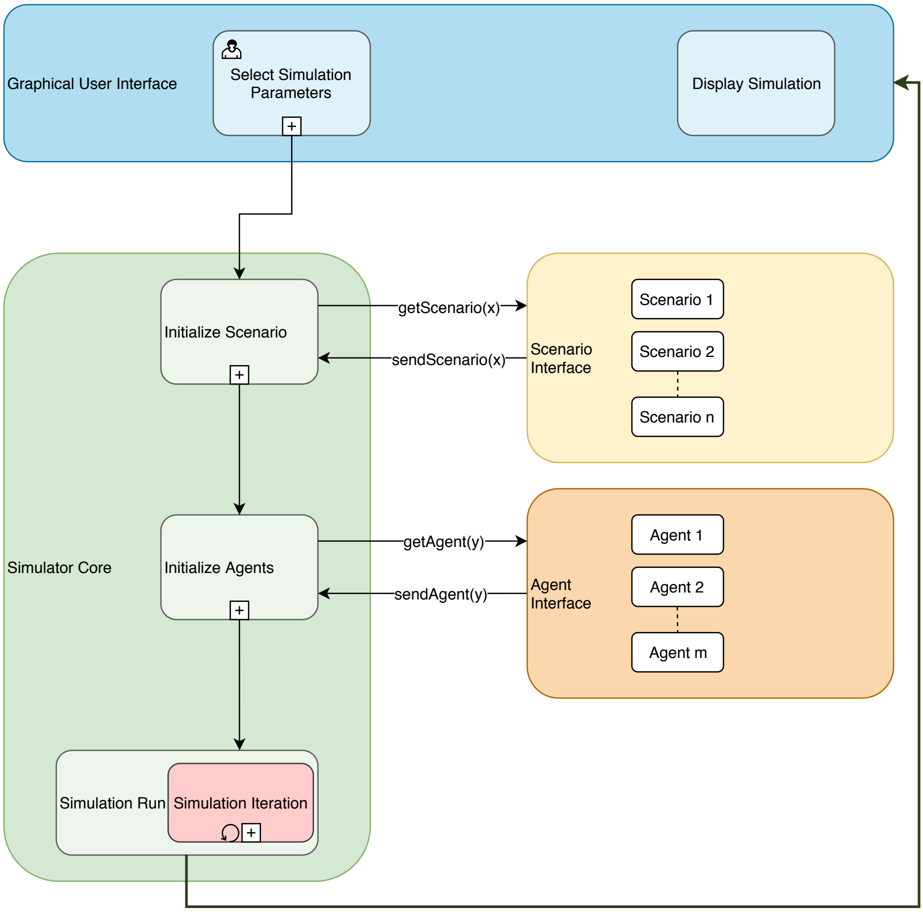


Figure 3.8: An overview of the simulator modules and their interaction

3.2.1 Graphical User Interface

The graphical user interface is the part of the program that is displayed to the user. Figure 3.9 shows a screenshot of the GUI where a simulation run has been started. The user can change the parameters on the left side and the top side of the GUI. When the *start simulation* button is pressed the simulation run is started and displayed in the GUI. The GUI has an output field for the simulator on the bottom, where information about the simulator run is shown.

Figure 3.9: A screenshot of the graphical user interface

3.2.2 Scenario Interface

The scenario interface was created so that different scenarios could be designed for the experimental tests. It defines four functions required by the simulator to simulate any scenario:

- **Scenario name**: The name of the scenario to distinguish different scenarios.
- **Areas**: The areas and hallways that the agents can move within. These also include the exits.
- **Hazard Areas**: The areas that are or will become hazard during the simulation.
- **Guidance Equipment**: The guidance equipment that should be present in the scenario.

3.2.3 Agent Interface

As with the scenario interface, the agent interface was created so that different agent implementations could be simulated. In this work the swarm agent is the only implementation of the agent interface, but the simulator was build so that other agent implementations are supported as well.

The agent interface defines all the common variables and functions required by the simulator, so that it can simulate the agent. The variables and functions are listed in table 3.2.

Agent variables	
Name	**Description**
x position	x position of the agent on the X-axis
y position	y position of the agent on the Y-axis
velocity	A vector describing the agent's velocity
findNextMove()	A function called to calculate the agents next move

Table 3.2: The variables and functions defined in the agent interface

The x and y position variables are used to represent where the agent is located in the scenario and the velocity vector represents it's speed and direction.

findNextMove() is the function that allows the agent to update its wanted velocity given its implementation. When this function is called in the swarm agent implementation, the behavior system is run to update its wanted velocity. This function is what makes each agent implementation unique.

3.2.4 Simulator Core

The simulator core is the part of the simulator that initializes the given scenario and agent type and runs the simulation. Pseudo code for the simulator core is shown in algorithm 4.

Algorithm 4: Simulator core pseudo code

```
input: scenarioNumber, agentType, agentCount, simulationIterations
scenario = initializeScenario(scenarioNumber);
agents = initializeAgents(agentType, agentCount);
timeStep = 0;
for i = 0; i < simulationIterations; i = i + 1 do
    while agentCount > 0 do
        for agent in agents do
            agent.findNextMove();
        end
        for agent in agents do
            agent.updatePosition();
            if agent.hasExited() then
                agentCount = agentCount - 1;
                agents.remove(agent);
            end
            timeStep = timeStep + 1;
        end
    end
    updateGUI(scenario, agents);
end
```

The input of the algorithm is the one given from the GUI and it is used to initialize the scenario and the agents. The outer for-loop represents the number of simulation iterations used in the simulation run. The while-loop is run as long as there are still agents left in the simulation iteration. For each time this loop is repeated, the **timeStep** variable is increased by one. This variable is used as

the simulation time unit. For each timeStep the agent is allowed to update its velocity once when **findNextMove()** is called. When all agents have updated their wanted velocity, the **updatePosition()** function is called to update the agents position as shown in equation 3.7 and 3.8. If the new x or y positions imply a collision with walls or other agents it will not update its position, but instead remain on its current position and the agent stuck count in the swarm agent implementation is increased as a result.
Note that the collision avoidance step happens when the findNextMove() function is called. If the agent is not allowed to update its movement in the updatePosition() function this is because the final velocity still resulted in a collision.

$$x = x + velocity_x \tag{3.7}$$

$$y = y + velocity_y \tag{3.8}$$

Chapter 4

Experiments and Results

This chapter describes the tests that were performed in the simulator to measure the performance of the swarm agent approach. The tests were divided into two phases: preliminary and experimental. The preliminary tests lay the basis for the experimental tests and the experimental tests aim to answer the research questions in this work. For each test the results are presented, followed by a discussion. The experimental plan presents an overview of the tests, describes the test procedures, the scenarios and the different settings used in the scenarios.

4.1 Experimental Plan

In order to find answers for the research questions in this work several tests were performed. An overview of the tests and their relationship to the research questions is presented in the test plan. Further, the experimental plan describes the test procedures used in the simulation tests and how data was gathered to produce the results. Finally, a description of the test scenarios and the different environment settings are presented.

4.1.1 Test Plan Overview

The test plan shows an overview of the tests performed in the preliminary phase and the experimental phase of this work. The overview aims to give an initial introduction to these tests.

Preliminary Tests

Table 4.1 presents the preliminary tests that were performed. The first three tests were aimed at the simulator parameters and the last two on the swarm agent parameters.

Test number	Test name
PT 1	Simulation iteration count
PT 2	Swarm population size
PT 3	Initial agent distribution
PT 4	Evacuation objective weight
PT 5	Separation, alignment and cohesion weights

Table 4.1: Preliminary test plan overview

Experimental Tests

The experimental tests are divided into two phases, as shown in table 4.2. The tests in phase 1 are directed towards research question 1 and the tests in phase 2 addresses research question 3, while both phases are relevant for answering research question 2.

Test number	Test name
Phase 1: Environments without hazards	
ET 1	Guided environments with and without swarm interaction
ET 2	Unknown environments with and without swarm interaction
Phase 2: Environments with hazards	
ET 3	Guided environments with static hazards
ET 4	Guided environments with dynamic hazards
ET 5	Unknown environments with static hazards
ET 6	Unknown environments with dynamic hazards

Table 4.2: Experimental test plan overview

4.1.2 Data Gathering and Results

During a simulation two types of data are registered for each time step: the positions of each swarm agent and the number swarm agents that still are in the scenario. This data was used to produce three types of results: charts, heat maps and total evacuation times.

Charts are used to visualize the number of swarm agents that have found an exit for each time step in the simulation. They are individual for each iteration in the simulation run and are used to illustrate trends in the evacuation patterns.

Heat Maps are colored images of the scenario, illustrating the movement patterns of the swarm agents. The colors change on a scale from blue to green to yellow to red, indicating the amount of activity, as illustrated in figure 4.1. Blue indicates no activity while red indicates the most activity. Like the charts, the heat maps are also able to illustrate individual simulation iterations. However, by aggregating the data of the simulation iterations it can also show entire simulation runs. This is useful to find the average movement patterns of the swarm agents. The heat maps are also able to visualize the activity of the swarm agents before and after an event, useful when evaluating dynamic scenarios.

Figure 4.1: Heat map color scheme

Total Evacuation Time is a simple, but very important measurement. It represents the final time step of the simulation when the last swarm agent finds an exit. The total evacuation time can be averaged over the iterations to find the average total evacuation time of the simulation run.

4.1.3 Handling Infinite Simulations

Given certain parameters and scenario types there is a possibility that the swarm agents get stuck in the scenario, not finding their way out. If this were to happen the iteration would never be able to finish. To make sure these iterations terminate at some point a hard threshold was set at 50 000 time steps. Upon reaching this threshold the iteration is classified as did not finish (DNF) and is automatically terminated. Terminating the iteration does not end the simulation run, instead the iteration is disregarded from the simulation run and added to a count of iterations that did not finish.

A terminated iteration does not necessarily mean that parameters used were bad. The other iterations could have great results, but it implies that there exists a probability that the swarm agents can become stuck using those parameters in that specific scenario. When testing combinations of many different parameters, scenarios and scenario types this is bound to happen for some iterations. Instead of the alternative to completely disregard the simulation run and say that it did not complete, this approach allows the run to continue and at the same time gather useful information about the amount of iterations that did not complete.

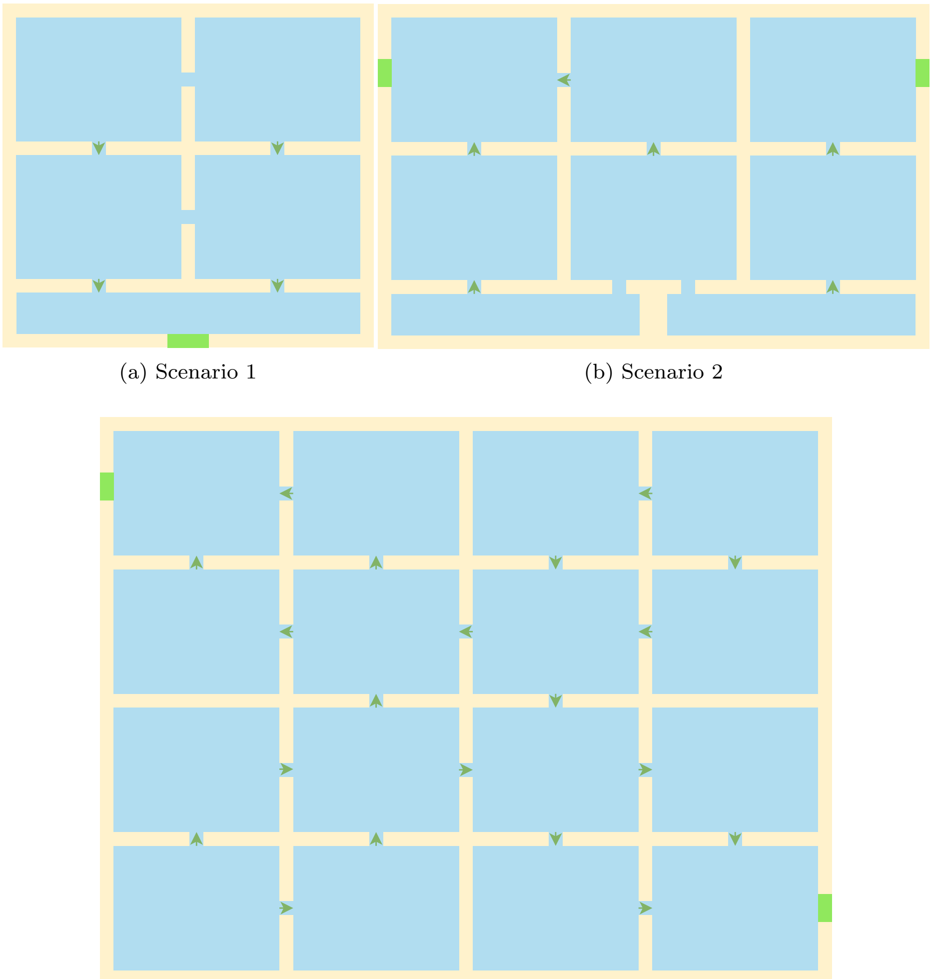
4.1.4 Scenarios

Each scenario is set to a building environment and consists of walls, areas, hallways, guidance equipment giving directions and exit(s). The swarm agent is presumed safe in the simulation when reaching an exit. The different scenario components are illustrated in figure 4.2. The guidance direction arrow indicates the guidance equipment showing the suggested evacuation direction.

Figure 4.2: Components used to describe the scenarios

An area can be either normal or hazardous. A normal area is assumed safe while the hazardous area is assumed dangerous. In a static environment the area types cannot change, while in a dynamic environment the normal areas can become hazardous during the simulation. However, it is possible to have hazardous areas in a static environment as long as they are so from the beginning. The hallways represent corridors that connect the areas and the swarm agents must use these to travel between the areas because they cannot travel through walls. If an area becomes hazardous, its adjacent hallways will also become hazardous as a result.

For the tests three different scenarios were created, as illustrated in figure 4.3. The guiding directions are placed on the hallways and are visible from the areas connected to the hallway. The guidance directions were placed in each scenario based on the shortest path to an exit. In scenario 3 there are multiple shortest paths from the same area and it does therefore have multiple guiding directions in the same area as well. The scenarios were created with different size and layout complexities for the experiments to measure how well the swarm approach generalizes in the different scenarios. Complexity implying an increased amount of loops present in the scenario, where the swarm agent could end up moving between the same areas while exploring the scenario for an exit.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 4.3: The evacuation scenarios used in the simulations

4.1.5 Swarm Interaction

Swarm interaction is the swarm agent's ability to interact with other swarm agents in its proximity radius. In the work herein this is done through indirect communication by following the three boid rules: separation, alignment and cohesion.

To simulate the swarm agent without this ability, the weights for separation, alignment and cohesion weights can be set to 0. By doing so its only behavior rule will be to follow the evacuation objective. When the tests refer to not using swarm interaction it implies that these weights were set to 0.

4.1.6 Static and Dynamic Environments

In a static environment the area type remains the same for the entire simulation, that is, areas cannot become hazardous during the simulation if they were not from the beginning. In a dynamic environment the area type can to change from normal to hazardous during the simulation. If an area is initialized as hazardous or becomes so during the simulation, it will remain so for the rest of the simulation.

In the simulation tests it is not random which areas become hazardous. Instead, they are selected and presented before the test, so that the results are comparable and specific situations can be tested.

Simulating a dynamic environment there is the question of when selected area(s) should become hazardous. This could either be at a given time step or other conditions such as: number of swarm agents left or at a certain percentage of swarm agents left. To keep the simulator runs consistent, all iterations in the same simulation run follows the same condition for when the area(s) should become hazardous.

4.1.7 Guided and Unknown Environments

The difference between guided and unguided environments is the presence of guidance directions in the scenario. As mentioned earlier, the guidance directions present the shortest path to an exit. By following these directions the swarm agents are guaranteed that they chose the shortest path. However, there is no guarantee that this path is the fastest or that is even possible to follow as a consequence of congestions and hazardous areas.

The guided directions represent some global knowledge of the environment that the swarm agent itself is unable to obtain. Still, by perceiving its local environment it is able to utilize some of this knowledge by following the guidance directions it is able to observe. The guided environment is not equal to the known environment because that would imply that the swarm agent possessed some global knowledge of the scenario.
The unknown environment on the other hand is equal to the unguided environment because there is no additional knowledge added to the environment or the swarm agent.

As described about the evacuation objective implementation in the previous chapter, the **guidance priority chance** is used to prioritize the guidance directions. Setting this chance to 0 equals not having any guidance directions present in the scenario. When simulating the unknown environments this chance was therefore set to 0.

4.2 Preliminary Tests

The preliminary tests were conducted during the development of the simulator and the implementation of the swarm agent in order to find suitable parameters to use for the experimental tests. The purpose of these tests was also to limit the scope of possible parameter combinations to test in the experimental tests. To begin the preliminary tests some initial parameters were chosen, as presented in table 4.3. The initial values for these parameters were based on observations done while developing the simulator and the swarm agent.

The preliminary tests were performed using unknown and static environments without hazards. The unknown environment was chosen because of the uncertainty it implied, where the swarm agents had to search the environment for an exit. The uncertainty was important when testing the simulation iteration count. Using a static environment without hazards was chosen because the focus of preliminary tests were on the initial parameters and not on the environment complexity.

4.2.1 Simulator Parameters

The first preliminary tests were performed to find simulator parameters that would give accurate results for the remaining tests. The most important parameter for this was the simulation iteration count. The swarm population size and initial agent distribution were also found important to test for this purpose.

Simulator Parameter	**Value**
Simulation Iteration Count	100
Swarm Population Size	500
Initial Agent Distribution	Equal
Swarm Agent Parameter	**Value**
Ignore Room Chance	0.01
Max Acceleration	1.0
Proximity Range	10.0
Stuck Count Limit	10
Guidance Priority Chance	1.0
Evacuation Objective Weight	1.0
Separation Weight	1.0
Alignment Weight	1.0
Cohesion Weight	0.1

Table 4.3: Initial model parameters

Preliminary Test 1: Simulation Iteration Count

As explained in section 3.2, each iteration in the simulator run is different. The accuracy of the simulation run is therefore strongly dependent on the number of iterations used. Too few iterations could give much variance in the results while too many could be time consuming and unnecessary.

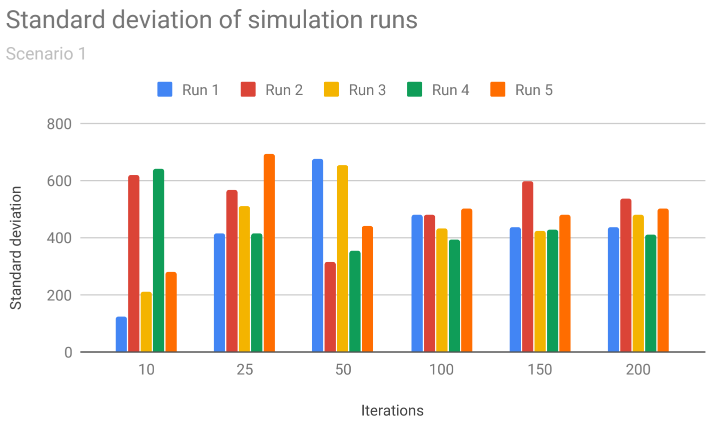
To measure how the number of iterations affected the accuracy of the simulation run, five simulator runs using different iteration counts were tested on all three scenarios. The lowest number of iterations being 10 and the highest being 200. The point of this test was to find the number of iterations required per simulation run to get accurate results without having to use more iterations than necessary.

Standard deviations of the total evacuation time were used to evaluate the accuracy of the simulator runs. The graphs presented in figure 4.4 are the standard deviations for each simulator run given the iteration counts. They represent the variation from the average evacuation time of each simulation run, given the iteration count. Five simulator runs were performed with the same iteration count to compare how stable the standard deviations were between each run, stable being similar values of the standard deviations. It is noticeable that when increasing the number of iterations the standard deviations became more similar as a result.

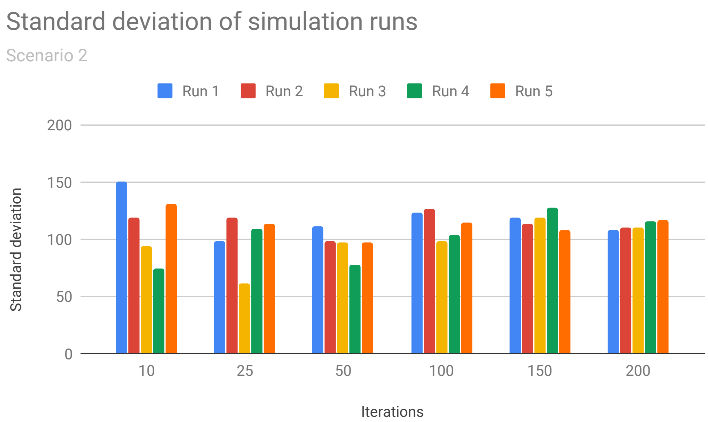
The standard deviations for each scenario are not directly comparable because they measure the deviation from the average evacuation time, which is unique to each scenario. However, by calculating the variance of the simulation runs standard deviation, one measures how much the standard deviation itself variate with respect to the iteration count. This is a good measure to evaluate the accuracy of the simulation runs because it is a number representing how similar the runs are and it is comparable between the scenarios. Figure 4.5 illustrates this variation for each scenario and the average variance. A low number implies that the variation between each run was similar, indicating a stable simulation run.

From the results the average variance decreases with the increase of iterations, which is good because this implies that the results are becoming more accurate by using several iterations. Looking at the results after 100 iterations the average variance improvement was slow, but scenario 3 achieved noticeable more stable results at 150 iterations. Based on these results the number of iterations per simulation run was changed from 100 to 150. This was based on the observation that 150 iterations gave more stable results compared to 100, especially scenario 3 and that there was little to no improvement of the average variance after this point.

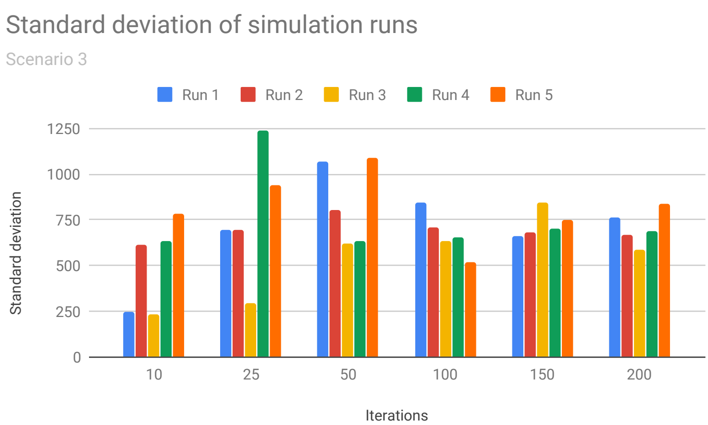
Scenario 2 stands out with its low values compared to the other scenarios. This is likely explained by its layout being the only one that does not have any loops for the swarm agents to become stuck and circle around in, while searching for an exit.



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

Figure 4.4: PT 1: Standard deviations of the simulation runs evacuation times

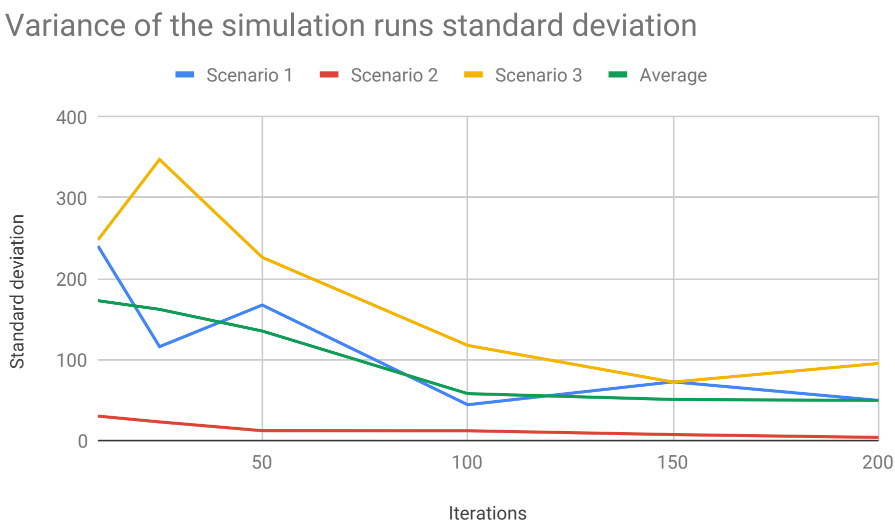


Figure 4.5: PT 1: Variance of the simulation runs standard deviation

Preliminary Test 2: Swarm Population Size

In a swarm approach, having a sufficiently large population size is important. What defines sufficient depends on the scenario. While a large scenario might require more than a smaller scenario, there should at least be enough swarm agents for swarm interaction to matter. However, in evacuation planning a large population size could also result in congestions and long evacuation times. This gave reason to believe that there exists a trade off between swarm behavior and congestions when increasing the population size.

To test this assumption each scenario was tested using different population sizes, the smallest population size being 10 and the largest being 3000. For each simulation run 150 iterations were used, for a total of 4 500 simulations for this test. The average total evacuation times were calculated and the results are illustrated in figure 4.6 and their respective standard deviations are found in figure 4.7.

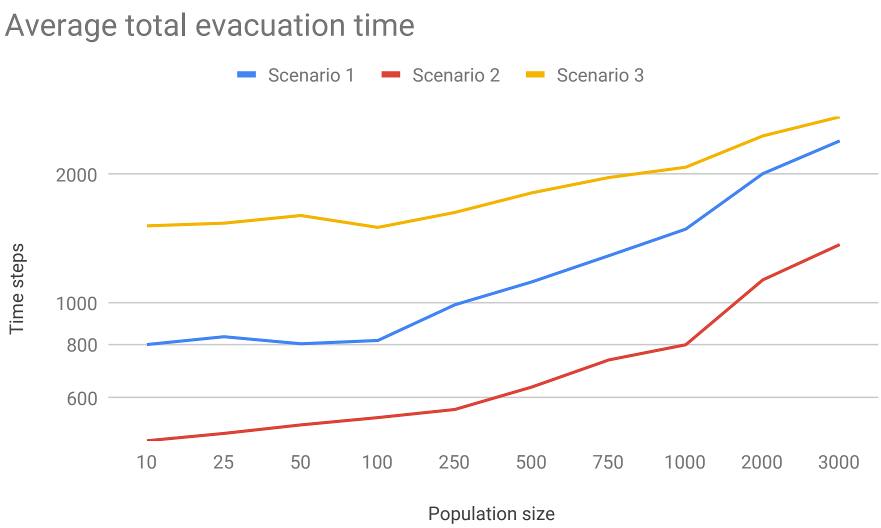


Figure 4.6: PT 2: Evacuation time results (scaled horizontal axis)

Evaluating the individual population sizes it is difficult to find any changes in the evacuation times that indicate any large congestions. The increase in evacuation time with respect to the population size indicates that there is congestion, but it is small compared to the increase in population size. Originally the largest

Standard Devation

Population Size	Scenario 1	Scenario 2	Scenario 3
10	412.51	76.07	716.54
25	464.64	74.69	739.93
50	435.74	63.20	895.62
100	436.98	64.97	740.62
250	381.95	61.66	791.47
500	373.20	116.39	608.51
750	467.44	113.52	710.85
1000	481.57	110.75	803.96
2000	444.18	98.07	758.35
3000	331.44	112.31	692.50

Figure 4.7: PT 2: Standard deviations of the evacuation times

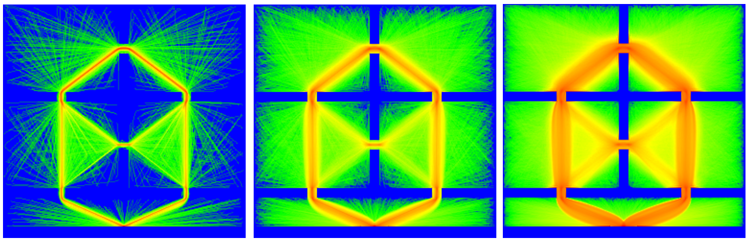
population size was set to 1000, but was later increased to 2000 and 3000 to see if these numbers would cause a larger delay in the evacuation process. Even then, the total evacuation times increased minimally compared to the amount of swarm agents. Evaluating the standard deviations, it is noticeable that the variance increases in scenario 2 after the population size becomes larger than 250, but why this happens is hard to say.

An interesting observation is that the evacuation times are almost equal in the range of 10-100 swarm agents. For scenario 1 and 3 using a population size of 100 swarm agents is actually better than using less. This could indicate that the local interaction of the swarm agents has a positive effect. The equal evacuation times in this range could also indicate that the population size compared to the size of the scenarios is to small for the agents interaction to matter and that they just individually finds an exit.

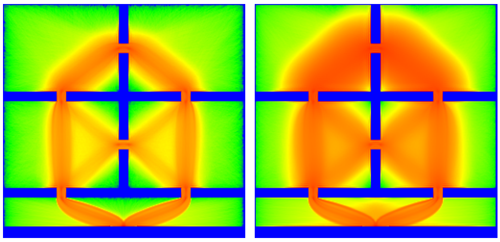
To further understand how the population size affected the swarm's behavior, the agents movement were tracked during the simulations and heat maps were generated for each population size and scenario. A selection of the heat maps are presented in figure 4.8 and the full collection can be found in appendix .1.1. The selection features a variety of population sizes for each scenario.

Each heat map is based on the 150 simulation iterations in the simulation run. Looking at the heat maps with a low population size there are many blue areas, implying that these locations were never explored by the swarm agents in any of the iterations. As a result of increasing the population size, it can be seen that almost all the locations in the scenarios are now explored. The coloring of the heat maps are normalized over the total number of agent positions registered, so a small population size has the same potential of coloring the heat map as a larger one. This is important to understand when evaluating the heat maps. When comparing any of the scenarios using a population size of 10 versus 3000, it is observable that the later one has larger red areas, indicating a wider spread in activity. This is a result of the swarm agents piling up at the hallways trying to pass through, also known as congestions. This effect is first apparent at a population size of 500 or more in the scenarios.

Assessing the evacuation times there was no obvious reason for preferring one population size over another. Keeping 500 swarm agents as the basis for the experimental tests seemed like a decent choice, both for the swarm size, but also to make room for some congestions, which is realistic in evacuation scenarios.

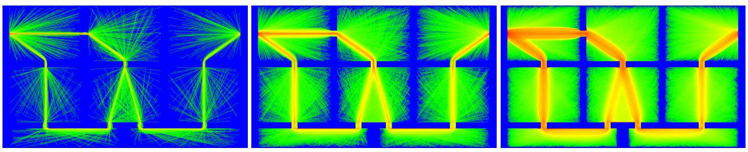


10 Swarm Agents 100 Swarm Agents 500 Swarm Agents

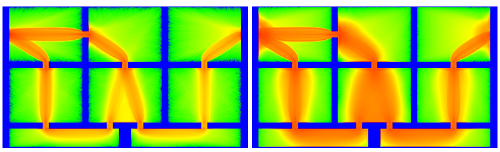


1000 Swarm Agents 3000 Swarm Agents

(a) Scenario 1

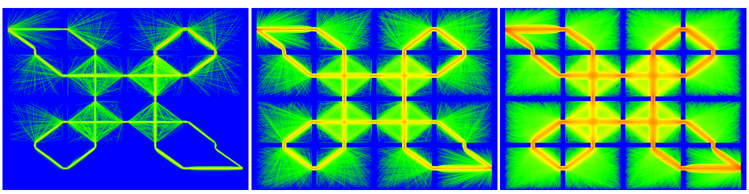


10 Swarm Agents 100 Swarm Agents 500 Swarm Agents

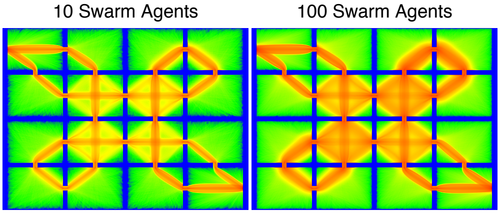


1000 Swarm Agents 3000 Swarm Agents

(b) Scenario 2



10 Swarm Agents 100 Swarm Agents 500 Swarm Agents



1000 Swarm Agents 3000 Swarm Agents

(c) Scenario 3

Figure 4.8: PT 2: Selected heat maps using different population sizes

Preliminary Test 3: Initial Agent Distribution

In evacuations the initial distribution of people is important in terms of exit paths and congestions. To measure this would affect the results in the experimental tests, two types of distributions were tested on the scenarios: random and equal.

In the random distribution the initial placement of the swarm agents were selected randomly amongst the areas. Using equal distribution the swarm agents were placed in the same areas, but they were equally distributed between the areas. Figure 4.9 illustrates the average total evacuation times of the 150 simulation iterations performed for each run and their standard deviations, using random and equal distribution.
Using the chosen population size of 500 the evacuation times were about equal on every scenario. Taking the standard deviations into account the average evacuation times are also within the same error margin. This is likely because the random distribution will converge towards the equal distribution when the population size increases and the amount of areas are constant. The equal distribution was kept for the remaining tests, seeing no reason to change the initial distribution to random.

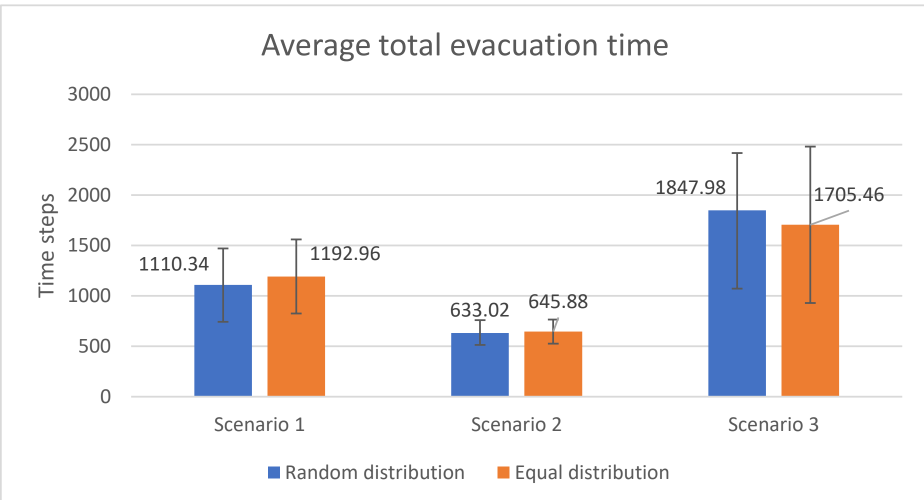


Figure 4.9: PT 3: Evacuation time results

4.2.2 Swarm Agent Parameters

In addition to max acceleration, proximity range, ignore room chance and the stuck count limit, the swarm agent has four weight parameters that forms its behavior. These are the weights for evacuation objective, separation, alignment and cohesion. Having eight adjustable parameters some limitations had to be made in order to limit the scope of the tests. The four behavior weights were found most important to test because their combination decides the swarm agent's final behavior.

As the four weights are used to adjust the relationship between the evacuation objective, separation, alignment and cohesion velocities, there is a relationship between all the weights. Finding the optimal combination of these weights given any scenario is out of the scope for this work, so approximations for good weights were made through two extensive tests instead.

Preliminary Test 4: Evacuation Objective Weight

Evacuation objective weight was the first swarm agent parameter to be tested, being the primary objective of the swarm agent this seemed reasonable to test before adjusting the weights related to the swarm behavior. Different weight ranges were tested on all three scenarios using the average total evacuation time as performance measure. The test results are presented in figure 4.10 and their respective standard deviations in figure 4.11.

As shown in the standard deviations, some of the low weights for the evacuation objective resulted in the swarm agents not being able to finish before 50 000 time steps were reached. At 0.5 in weight the swarm agents were able to complete all three scenarios and the evacuation times improved for scenario 1 and 2 until 0.6 in weight was reached and 0.7 for scenario 3. As the swarm agents were unable to complete the evacuations with too low weights for the evacuation objective and increasing the weight gave improvements for the evacuation times, the importance of the evacuation objective was confirmed.

An interesting observation is that the evacuation times of scenario 1 and 3 increases after passing 0.7 in weight, while they decrease for scenario 2. A possible theory for this is that the swarm agents cannot move between the areas in scenario 2 without finding an exit, while they can move in circles in the other two scenarios. When the weight for the evacuation objective increases, the swarm behavior will be less present as a result. This could indicate that the swarm behavior is beneficial for scenarios that have this loop layout. It could also be that

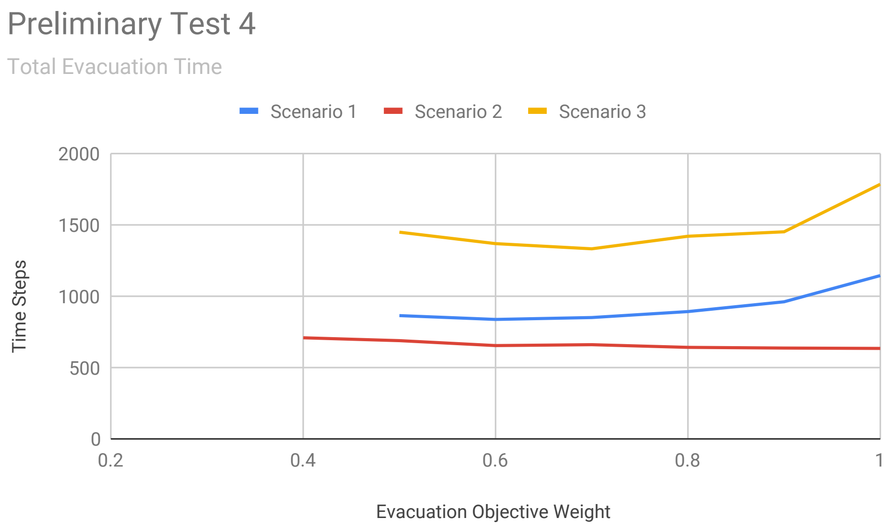


Figure 4.10: PT 4: Evacuation time results

the increased weight on the evacuation objective increases the chance of staying in a loop independent of any swarm behavior.

Scenario 3 had a noticeable improvement in its evacuation time moving from 0.6 in weight to 0.7 compared to the minor increase of scenario 1 and 2. A weight of 0.7 was therefore preferred over 0.6 for the final evacuation objective weight.

Standard Deviation

Evacuation Objective Weight	Scenario 1	Scenario 2	Scenario 3
0.1	DNF	DNF	DNF
0.2	DNF	DNF	DNF
0.3	DNF	DNF	DNF
0.4	DNF	118.96	DNF
0.5	182.41	106.92	409.47
0.6	143.72	109.89	272.48
0.7	150.93	105.32	285.71
0.8	221.62	103.68	319.49
0.9	270.41	111.26	339.63
1	378.11	113.99	734.51

Figure 4.11: PT 4: Standard deviations of the evacuation times

Preliminary Test 5: Separation, Alignment and Cohesion Weights

Having a relationship between the weights, it was thought sufficient to measure two of them against each other, while the third served as a constant value. The alignment and cohesion weight ranges were therefore tested against each other, letting the separation weight be constant at 1.0.

The average total evacuation times are presented in figure 4.12 and their standard deviations can be found in appendix .1.2. The values in the tables are colored from green to yellow to red. Green indicating the lowest evacuation time and red indicating the highest. Initially only scenario 1 was selected for this test, giving promising results for the weight combinations in the green area of figure 4.12a. However, before jumping to any conclusions the same weight parameters were tested on scenario 3 to make sure the weights were not only local best for scenario 1. Comparing the two tables it is evident that these weights were in fact only local best, because they resulted in the agents not finishing in scenario 3.

For high cohesion weights the swarm agent's behavior became too focused on moving towards the other swarm agents, not being able to finish in time. Increasing the alignment weight gave positive results for both scenarios, with the exception of some combinations in scenario 3 that resulted in the swarm agents getting stuck. These exceptions are hard to find a good explanation for, but they are related to the swarm agents not being able to move random enough around to get out of the loops in the scenario. Both the alignment and cohesion behavior rules increases the swarm agents coherence. It could be that when both these weights are increased, the flocking behavior becomes so dominant that the random behavior in each swarm agent becomes ineffective because they always prioritize following the swarm.

Given the different results from scenario 1 and 3, selecting the alignment and cohesion weights was no obvious decision. The results from scenario 3 restricted some of the best local weights for scenario 1 and the best local weight in scenario 3 were not the best in scenario 1. Still they had some common weight combinations that gave good results for both scenarios, one of them being 2.0 in alignment weight and 0.2 in cohesion weight. These weights were therefore selected as the final parameters before the experimental tests.

Alignment Weight \ Cohesion Weight	0.1	0.2	0.3	0.4	0.5	0.6
0.1	1416.15	1286.59	1500.94	6798.39	DNF	DNF
0.2	1314.49	1035.51	1155.89	2943.55	DNF	DNF
0.3	1151.32	902.41	963.27	1596.09	DNF	DNF
0.4	1162.25	867.77	841.99	1220.41	DNF	DNF
0.5	1022.43	859.30	836.17	1060.97	DNF	DNF
0.6	1052.42	803.28	768.04	1008.13	3496.67	DNF
0.7	925.07	774.31	750.31	900.19	2532.37	DNF
0.8	900.02	768.49	723.44	830.70	1923.25	DNF
0.9	866.07	761.52	717.18	772.85	1723.51	DNF
1	853.58	734.41	704.09	753.36	1347.29	DNF
1.1	833.90	728.00	691.38	728.12	DNF	DNF
1.2	829.75	745.93	700.29	726.87	1022.71	DNF
1.3	804.00	746.47	692.29	697.61	DNF	DNF
1.4	800.20	734.43	682.38	669.67	987.92	DNF
1.5	803.03	734.21	686.44	681.77	DNF	DNF
1.6	805.37	750.16	758.31	671.63	DNF	DNF
1.7	843.12	730.87	680.02	681.50	922.03	DNF
1.8	824.96	732.03	707.89	654.05	DNF	DNF
1.9	849.82	758.31	731.12	692.19	985.37	DNF
2	842.15	731.77	735.29	689.67	DNF	DNF

(a) Scenario 1

Alignment Weight \ Cohesion Weight	0.1	0.2	0.3	0.4	0.5	0.6
0.1	2433.22	2544.27	3052.73	DNF	DNF	DNF
0.2	2287.95	1786.65	1918.14	6237.21	DNF	DNF
0.3	DNF	1439.38	1655.24	3510.20	DNF	DNF
0.4	DNF	1372.04	1342.41	2439.59	DNF	DNF
0.5	DNF	1340.21	1312.05	2037.25	DNF	DNF
0.6	2503.39	DNF	1271.55	1773.62	8156.33	DNF
0.7	1597.57	DNF	1285.43	1671.18	5171.44	DNF
0.8	1420.05	DNF	1277.31	1576.51	3996.37	DNF
0.9	1339.29	1753.92	1377.01	1531.86	3566.54	DNF
1	1411.33	DNF	DNF	1612.89	3006.58	DNF
1.1	1366.37	1670.02	DNF	1462.82	2656.06	DNF
1.2	1333.65	1271.15	DNF	1474.05	2617.09	DNF
1.3	1335.21	1456.29	DNF	1611.81	2406.74	DNF
1.4	1302.69	1328.85	DNF	1987.29	2225.29	DNF
1.5	1289.87	1300.29	DNF	DNF	DNF	DNF
1.6	1289.94	1232.48	DNF	DNF	2253.93	DNF
1.7	1301.23	1239.99	DNF	DNF	2158.77	DNF
1.8	1337.24	1227.91	DNF	DNF	2174.61	DNF
1.9	1269.90	1225.25	DNF	DNF	DNF	DNF
2	1392.29	1192.57	DNF	DNF	DNF	DNF

(b) Scenario 3

Figure 4.12: PT 5: Evacuation time results

4.3 Experimental Tests

This section describes the experimental tests conducted in order to find answers for the research questions defined in the beginning this work. The updated parameters for the simulator and swarm agents can be found in table 4.4. Values marked in blue represents the parameters that were updated as a result of the preliminary tests.

Simulator	
Parameter	**Value**
Simulation Iteration Count	150
Swarm Population Size	500
Initial Agent Distribution	Equal
Swarm Agent	
Parameter	**Value**
Ignore Room Chance	0.01
Max Acceleration	1.0
Proximity Range	10.0
Stuck Count Limit	10
Guidance Priority Chance	1.0
Evacuation Objective Weight	0.7
Separation Weight	1.0
Alignment Weight	2.0
Cohesion Weight	0.2

Table 4.4: Updated model parameters

4.3.1 Phase 1: Environments Without Hazards

As mentioned in the test plan, phase 1 is directed towards research question 1: *Can some global behavior beneficial to evacuations be obtained by applying the boid model?*. The purpose of this phase was to evaluate how the boid behavior rules in the proposed swarm agent model affected the global evacuation patterns. The tests compare the swarm agents behavior in evacuation scenarios with and without using the boid behavior rules in the model.

Experimental Test 1: Guided Environments With and Without Swarm Interaction

The first test was set to guided environments, as the presence of guidance equipment is the normal case for building scenarios. In the guided environments the swarm agents have the advantage of being able to follow the guidance directions to find an exit, instead of having to search the scenario.

Six simulation runs, each with 150 simulation iterations, for a total of 900 simulations were run for this test. Two for each scenario, one with swarm interaction and one without. The average total evacuation times of the simulation runs are presented in figure 4.13 and the standard deviations are included at the top of each column.

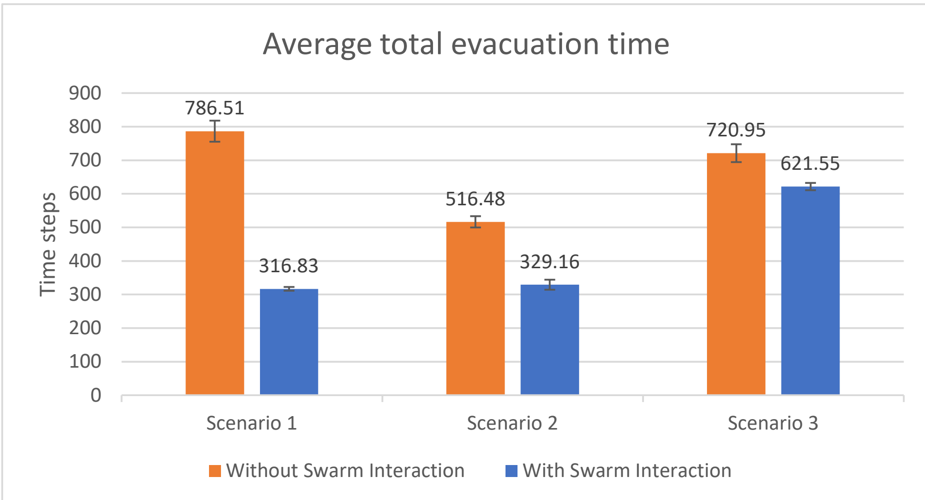


Figure 4.13: ET 1: Evacuation time results

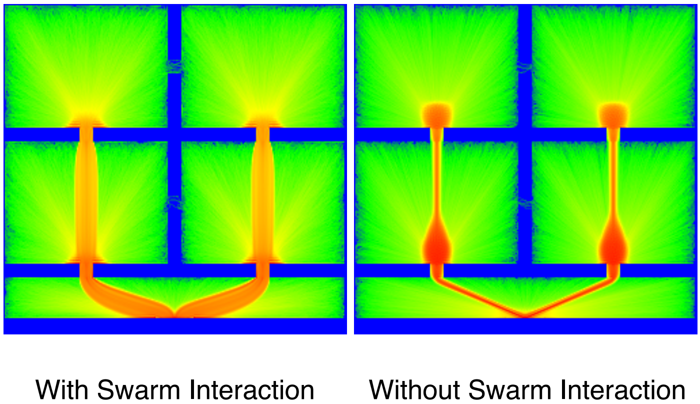
Evaluating the evacuation times, it is evident that the swarm interaction from the boid behavior rules plays an important part in the total evacuation time of the swarm agents, especially for scenario 1 where the total evacuation time is less than half of the one without this behavior. Using a guided environment, it is likely that the observed improvement in the evacuation times for the swarm agents using swarm interaction, is a result of their interaction facilitating for a better movement flow.

There is a pattern here where the evacuation times become more similar going from scenario 1 to scenario 2 and then scenario 3. This is likely related to the size of the scenarios and that using 500 swarm agents results in more agents per area in scenario 1 than in scenario 2 and 3. So when the number of swarm agents per area decreases, the improvement gained from swarm interaction is less present. Evaluating the heat maps of the scenarios in figure 4.17, this relation between the the swarm interaction and scenario size is noticeable.

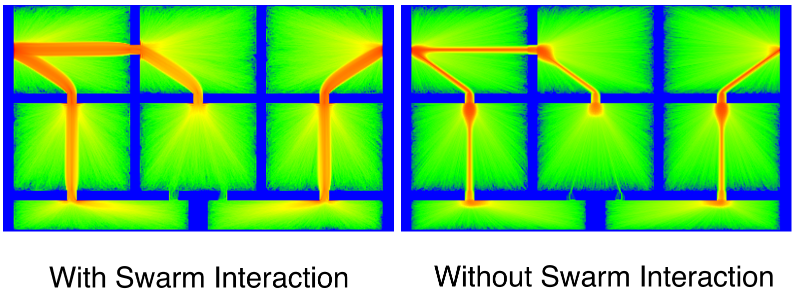
Looking at scenario 1, that achieved the most improvement using the swarm interaction, there is a distinct difference in the heat maps between using swarm interaction and not. The one with swarm interaction has wide lines of movement leading to the exit, indicating that the swarm agents spread out and moved side by side through the areas. The one without on the other hand has thin lines in comparison, where the agents preferred to walk in a line instead of spreading out. By walking in a line the resulting movement flow was much slower than the the one where the swarm agents walked side by side. This is especially noticeable at the hallways, by looking at the large red areas, where the agents without swarm interaction experienced more congestions.

Studying the heat map of scenario 2, the same patterns discussed in scenario 1 are observable, but to a smaller degree. This is probably because of the increased scenario size. Looking at the heat map of scenario 3, it is noticeable that the swarm agents still spread out compared to the agents without swarm interaction. Still, the evacuation times for scenario 3 are quite similar. Looking at the hallways of scenario 3 there are no noticeable congestion areas for both approaches, so when no congestions occur, the evacuation times are almost equal.

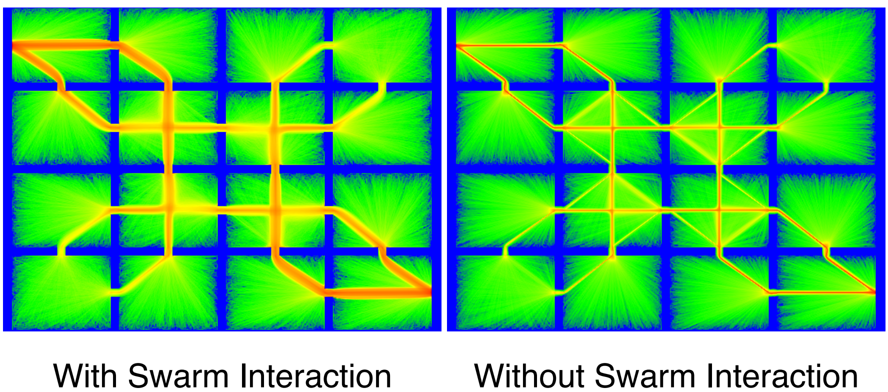
Evaluating the results from experimental test 1, it is evident that the swarm interaction has a positive effect on the movement flow of the swarm agents in the scenarios. The fact that it is more effective as the number of swarm agents increases per area is promising for the suggested swarm agent approach. Evaluating the individual boid rules it was probably the separation behavior that enabled the swarm agents to spread out so that they moved side by side in the guided environment.



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

Figure 4.14: ET 1: Heat maps of the scenarios

Experimental Test 2: Unknown Environments With and Without Swarm Interaction

In the context of this work it is important that the swarm agent is able to complete evacuations in unknown environments. The goal with this experimental test is to see how the swarm agent using swarm interaction performs compared to the agent without swarm interaction and if there are any noticeable patterns in their movement that can be found beneficial to evacuations in unknown environments.

As with experimental test 1, six simulation runs, each with 150 simulation iterations, for a total of 900 simulations were run for this test as well. Two for each scenario, one with swarm interaction and one without. The average total evacuation times of the simulation runs are presented in figure 4.15 and the standard deviations are included at the top of each column.

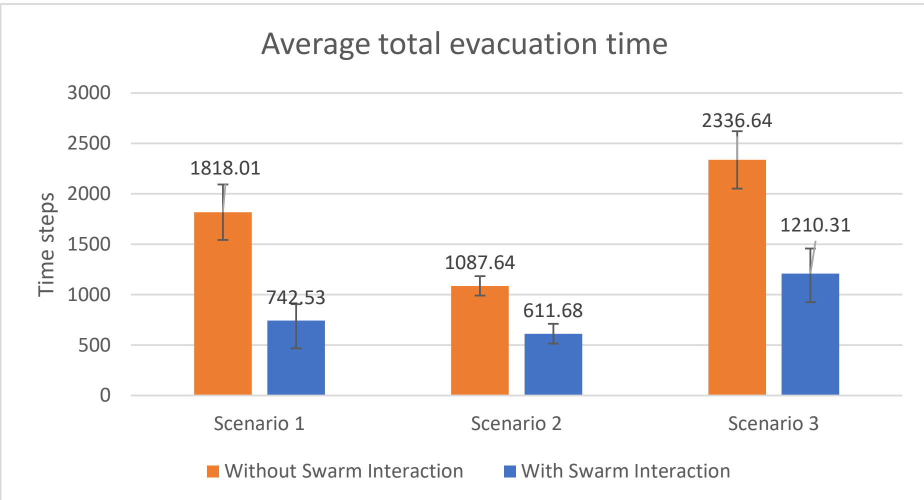


Figure 4.15: ET 2: Evacuation time results

Looking at the evacuation times there is now a new trend. The simulation runs with swarm interaction use only about half of the evacuation time compared to the simulation runs without swarm interaction on all the scenarios. This is a good indication that applying the boid technique for evacuations can indeed be beneficial for unknown environments.

Comparing the standard deviations from the previous test to the ones in this test, the variations are now higher. This is a result of going from a guided environment where the shortest paths are shown in the environment, to an unknown environment where the swarm agents have to search the environment for an exit. In some simulation iterations the swarm agents find the exit fast, while they use more time in others, causing more variance in the total evacuation times.

To help understand why the simulation runs using swarm interaction were able to perform so much better than the ones without, charts of the individual simulation iterations were created. The charts show how the swarm agents evacuated in the given simulation iteration with respect to the simulation time steps. Since each simulation run has 150 iterations, a total of 300 such charts were therefore created for each scenario. Figure 4.16 show a selection of four of the charts, two with swarm interaction and two without for scenario 1.

The trend shown in the charts for the simulation iterations with swarm interaction is that a high degree of swarm agents are able to evacuate over short amount of time steps, followed by a plateau where no swarm agents evacuate. This is a result of the swarm agents coherence, where they flock together and evacuate in groups. The plateaus are the time periods where no group is currently exiting.

In the simulation iterations without swarm interaction, the charts show another evacuation pattern with no plateaus. Instead there is a steady flow of agents evacuating, followed by the last agents spending a good amount of time searching for an exit. Notice that the charts horizontal axis are scaled to when the last agent evacuates, so that even though the flow is steady, it is still slower than the approach with swarm interaction.

The search process of the swarm agents in the unknown environment with and without swarm interaction is illustrated in the heat maps in figure 4.17. A first observation, compared to the guided environment, is that the agents movement has no longer a clear direction. In scenario 1 the middle hallways were never used in the guided environment, but when the environment is unknown these hallways are considered just as good options for the evacuation objective. The general observation here is that all hallways are now explored, instead of just the ones with guidance directions.

An observation from studying the heat maps of scenario 1 is that the hallway in the middle of the scenario is less used by the agents with swarm interaction than the ones without. Where they color the heat map with a yellow color, the agents without swarm interaction color it by a red color, implying more activity. This could be because the global movement of the swarm is moving down towards the exit and the cohesion and alignment rules of the swarm agent prioritizes following this movement instead. Stating that the global movement of the swarm is heading downwards is based on the observation of the red areas before the hallways leading in that direction, indicating congestions and that the swarm agents had to wait for some time in order to pass through.

Another observation on the scenarios without swarm interaction is that the movement of the agents appears wider on the areas not connected to an exit on the heat maps than the ones with an exit. This is because these agents are not affected by each others movement and therefore keeps their direction when heading towards a hallway. The spread in these areas is a result of the collision avoidance moving the agent out in the room to avoid colliding into the oncoming agents. When the agents enter a room with an exit their evacuation goal becomes the same and they move in the thin line instead.

Comparing the simulation runs with swarm interaction to the ones without, it is apparent that the flocking behavior is beneficial for evacuations in unknown environments. From the first observation on the heat maps it looks like the flocking behavior itself is useful for limiting the amount of unnecessary searching in the unknown environments. From the second observation it is also apparent that the agents without swarm interaction are ineffective because they block each others movement to a much higher degree.

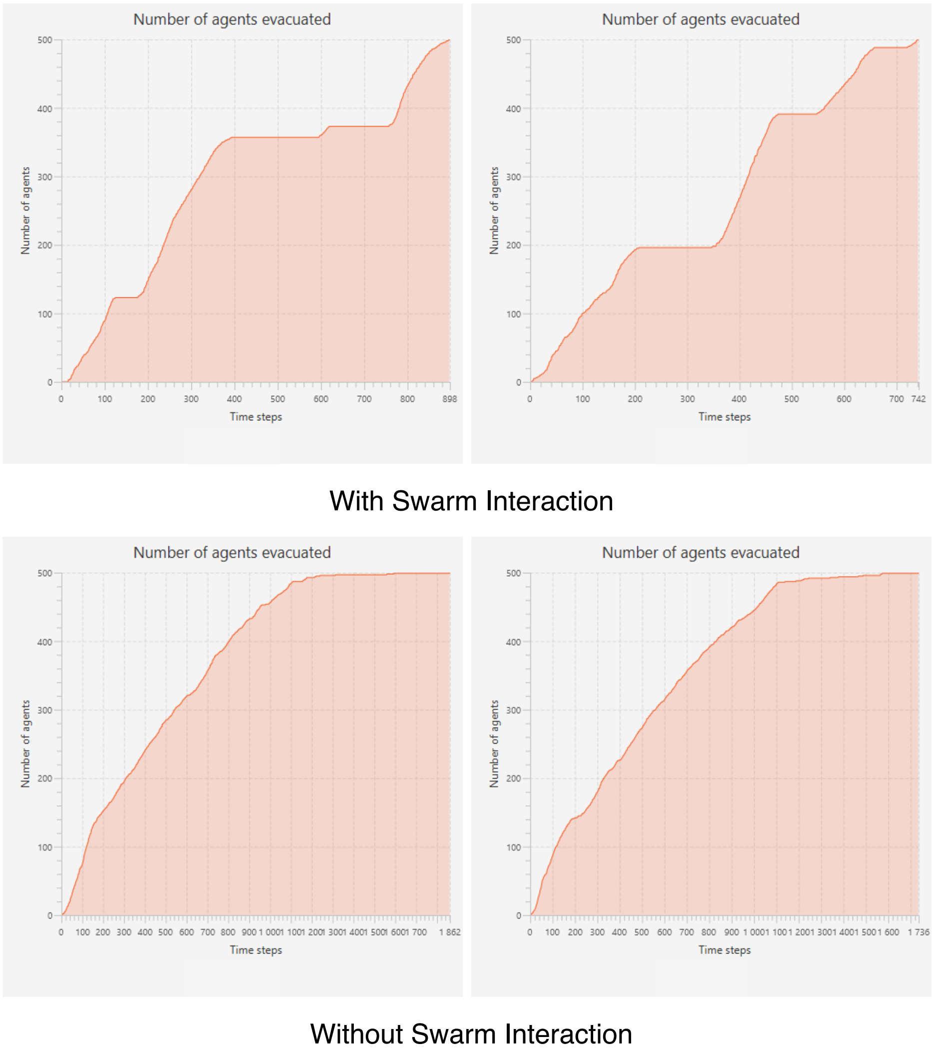
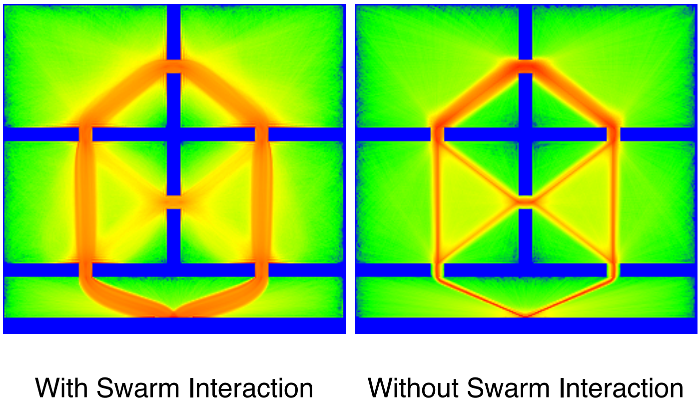
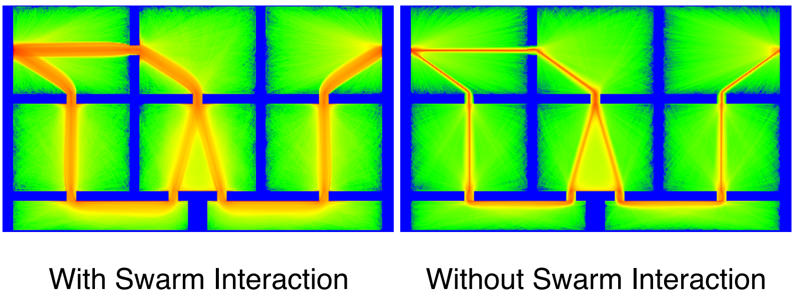


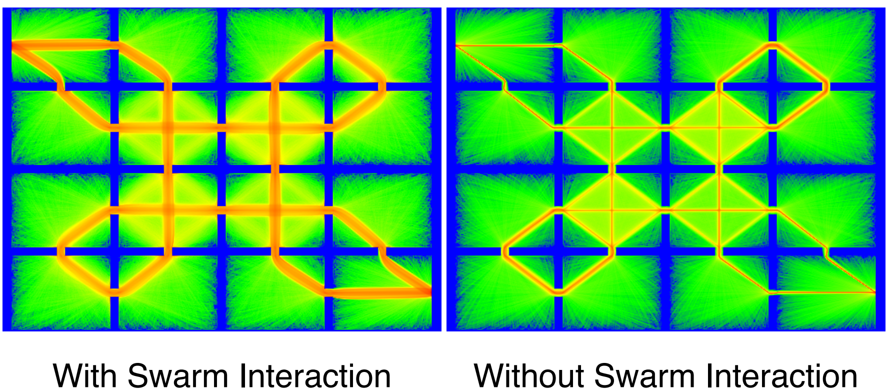
Figure 4.16: ET 2: Charts showing individual simulation iterations of scenario 1



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

Figure 4.17: ET 2: Heat maps of the scenarios

Comparing the Model in Guided Versus Unknown Environments

So far the swarm agent model has been tested and compared with and without swarm interaction in guided and unknown environments. Figure 4.18 shows a comparison of the evacuation times presented in experimental test 1 and 2.

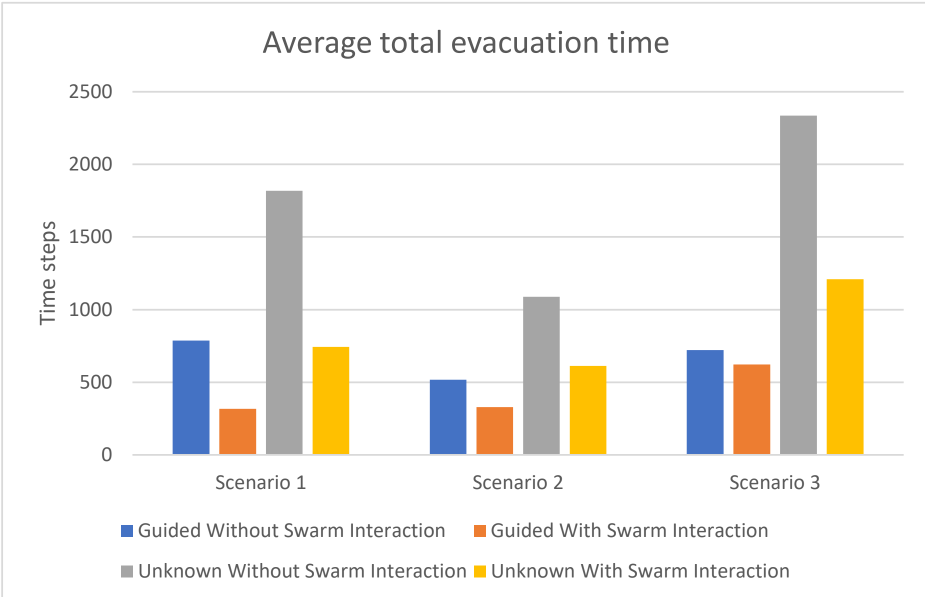


Figure 4.18: Comparison of the evacuation time results in experimental test 1 and 2

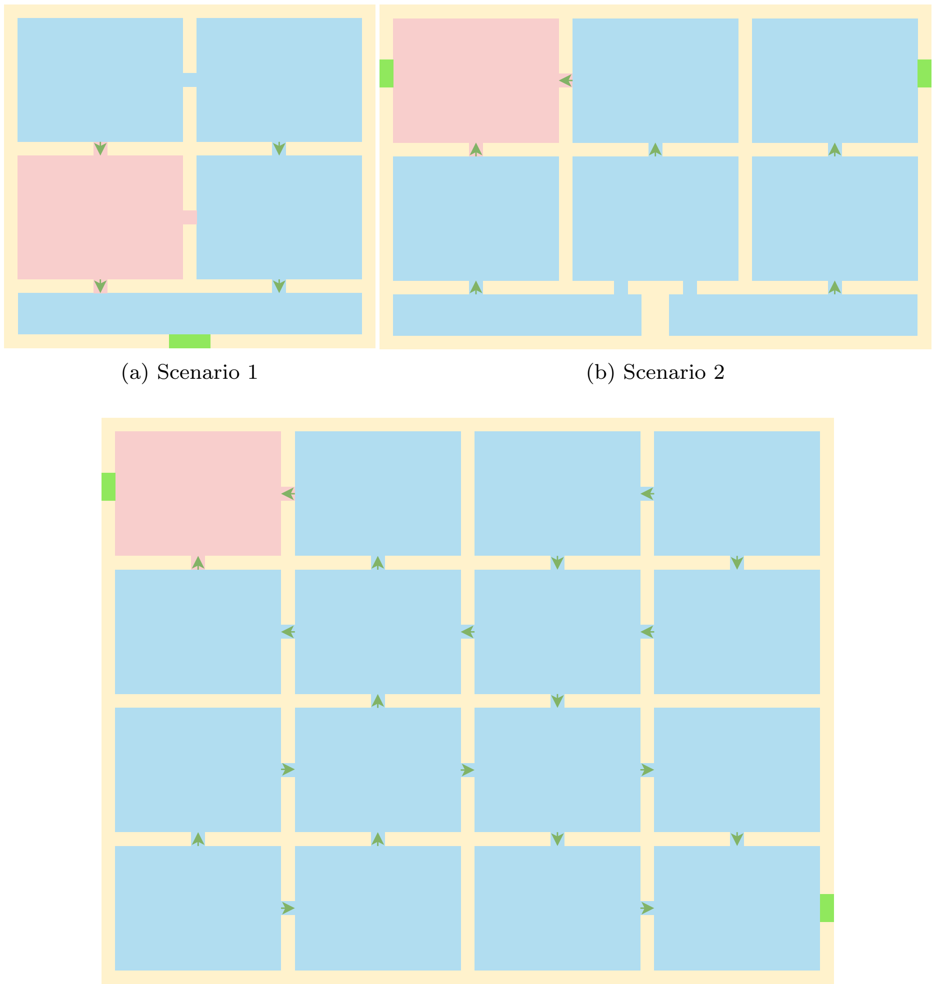
From the results it is clear that the agent with swarm interaction performs best in the guided environment for all scenarios, while the agent without swarm interaction in the unknown environment has the worst results for all three scenarios. Comparing the agents in the guided and unknown environments, both with and without swarm interaction, it is evident that that the guidance directions are beneficial for the model's evacuation times.

The agents with swarm interaction in the guided environment use around half of the time to evacuate compared to the swarm agent in the unknown environment. While the evacuation times are higher for the swarm agent in the unknown environment, the relative time difference is almost equal for each scenario even though the scenario size and layout is different. This is not the case for the agent without swarm interaction, where the evacuation time of scenario 3 in the unknown environment is over triple compared to the guided environment. Based on the results it looks like the agent with swarm interaction is able to generalize better in the different scenarios.

4.3.2 Phase 2: Environments With Hazards

The tests in phase 1 evaluated how the swarm interaction from the boid behaviors affected the model in guided and unknown environments. From the results it was apparent that this behavior was beneficial for the model. Phase 2 is directed towards research question 3: *How do hazards complicate the behavior of the swarm agent model in guided and unknown environments?*. The purpose of this phase was to evaluate how the swarm agent model performed in environments with hazards.

The areas that were selected to be hazardous for the experimental tests in phase 2 are shown in figure 4.19. The areas were selected before the tests so that the results would be comparable from using the same areas and that it would still be possible to evacuate from the scenario.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 4.19: The hazardous areas in the evacuation scenarios used in the simulations

Experimental Test 3: Guided Environments with Static Hazards

The results from phase 1 showed that the swarm agent performed best in guided environments, but this was without any hazardous areas. The purpose of this test was to see how the evacuation times of the model would be affected when hazards were introduced to the scenarios.

Three simulation runs using 150 simulation iterations were run on each scenario, for a total of 450 simulations. The average total evacuation times of the simulation runs are shown in figure 4.20 with the standard deviations on top of each column. Scenario 2 and 3 are empty because the swarm agents were not able to complete the scenarios before the 50 000 time steps threshold.

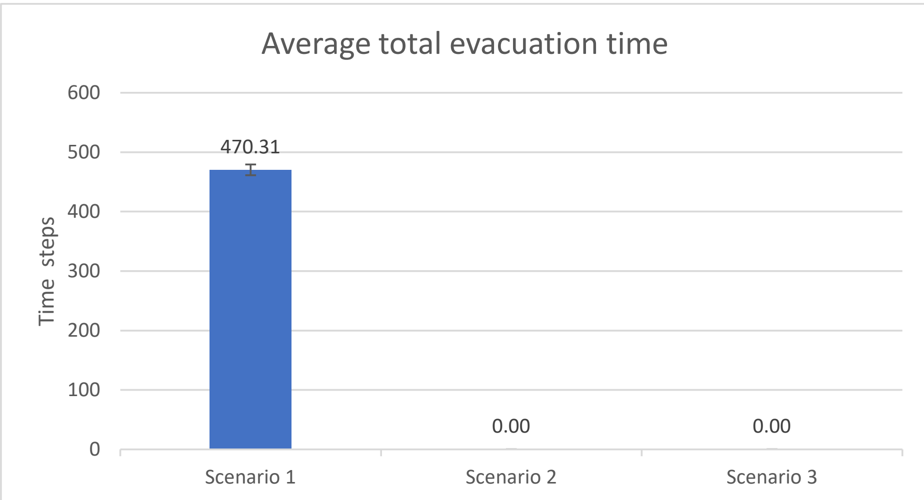
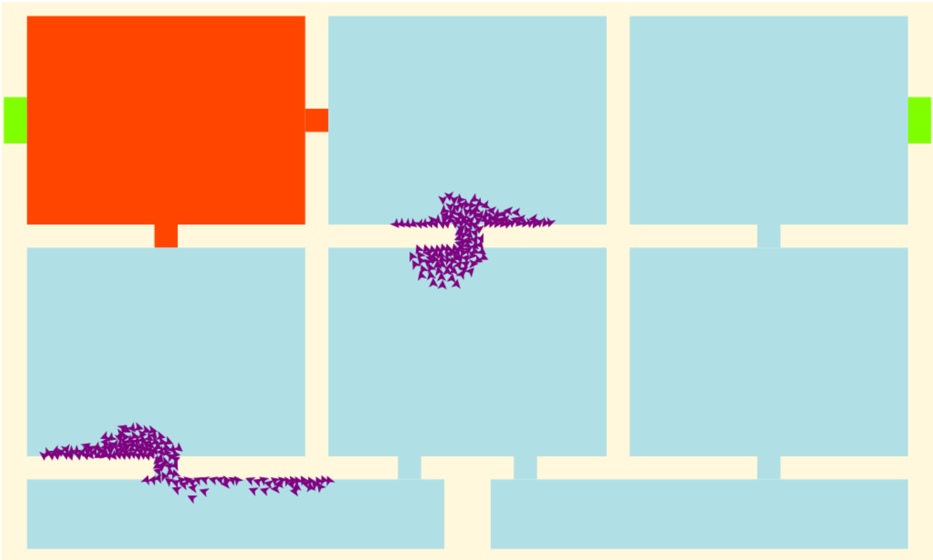
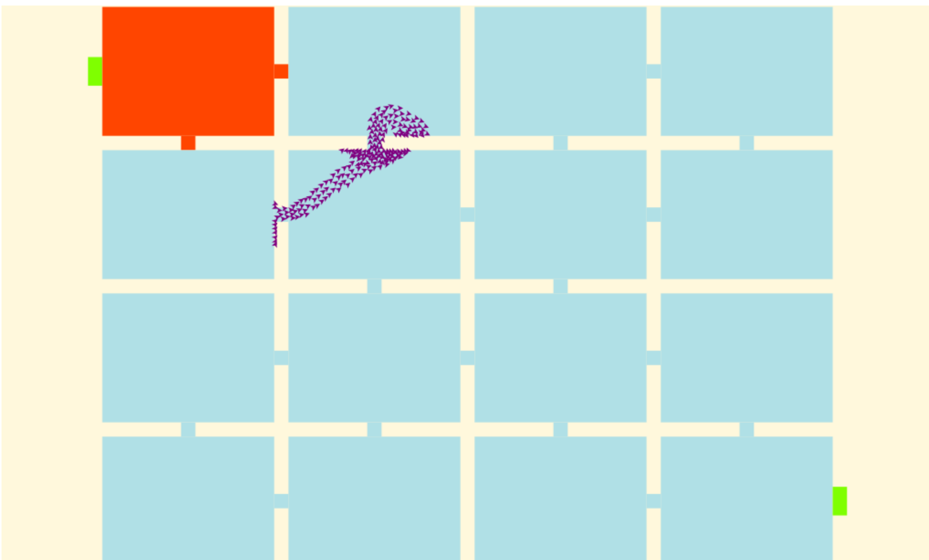


Figure 4.20: ET 3: Evacuation time results

Why the swarm agents were not able to complete the scenarios can be explained by looking at the screenshots from the simulator presented in figure 4.21. In scenario 2 the swarm agents were going through the same two hallways, while in scenario 3 they went back and forth between two hallways. This behavior was a result of the evacuation objective following the guidance directions and that the areas the guidance directions led to were hazardous. The only option for the evacuation objective was therefore to go back, only for the evacuation objective to follow the guidance direction into the same area again, leading to an infinite loop.



(a) Scenario 2



(b) Scenario 3

Figure 4.21: ET 3: Screenshots of scenario 2 and 3

The infinite loop encountered in scenario 2 and 3 was a result of the evacuation objective always prioritizing the guidance directions. In order to break this loop the evacuation objective needed to be able to select other hallways as well by reducing the **guidance priority chance**. 54 simulation runs using 150 simulation iterations, for a total of 8 100 simulations were run, testing different values for the guidance priority chance on the three scenarios. The evacuation times and the standard deviations are presented in the tables in figure 4.22. The colors range from green to yellow to red, indicating the lowest and highest values for each scenario.

Average Total Evacuation Time

Guidance Priority Chance	Scenario 1	Scenario 2	Scenario 3
0.9	470.60	DNF	DNF
0.8	470.33	DNF	DNF
0.7	471.77	DNF	DNF
0.6	472.59	DNF	DNF
0.5	473.63	DNF	DNF
0.4	479.22	15989.66	DNF
0.3	492.13	7723.36	DNF
0.2	544.69	5030.24	3727.06
0.1	641.59	3935.01	2097.81
0.09	644.86	3573.29	2104.72
0.08	649.11	3771.84	2073.37
0.07	682.64	3963.37	2109.05
0.06	676.63	3578.67	2122.57
0.05	694.36	3741.14	2138.42
0.04	704.47	3498.27	2125.20
0.03	726.71	3563.89	2126.41
0.02	727.71	4229.07	2214.88
0.01	749.27	4057.71	2225.76

(a) Average total evacuation time

Standard Deviations

Guidance Priority Chance	Scenario 1	Scenario 2	Scenario 3
0.9	8.90	DNF	DNF
0.8	9.05	DNF	DNF
0.7	9.42	DNF	DNF
0.6	8.16	DNF	DNF
0.5	12.67	DNF	DNF
0.4	27.40	7270.66	DNF
0.3	49.57	3711.48	DNF
0.2	98.14	2569.38	4457.91
0.1	125.58	2387.17	494.49
0.09	113.66	2200.12	435.27
0.08	121.35	2632.55	427.01
0.07	121.15	2490.29	406.91
0.06	112.89	1930.29	405.22
0.05	118.83	2570.49	452.18
0.04	120.43	2194.07	448.18
0.03	98.96	2048.98	524.10
0.02	115.00	3781.54	448.69
0.01	114.64	3664.50	462.45

(b) Standard deviation

Figure 4.22: ET 3: Guidance priority chance test results

Looking at the results for scenario 1 it is noticeable that the evacuation times increase with the reduction in the guidance priority chance. At 0.01 in guidance priority chance the swarm agent's behavior is almost equal to how it behaves in the unknown environment. As the agents now have to search the environment for an exit, they finish at more different times and the variance shown in the standard deviations increase as a result. Figure 4.23 presents a comparison of the swarm agents evacuation times compared to the results it had in the unknown and guided environments. While this test also included a hazardous area, it is observable that the evacuation time approaches the unknown environment's time when the guidance priority chance is reduced.

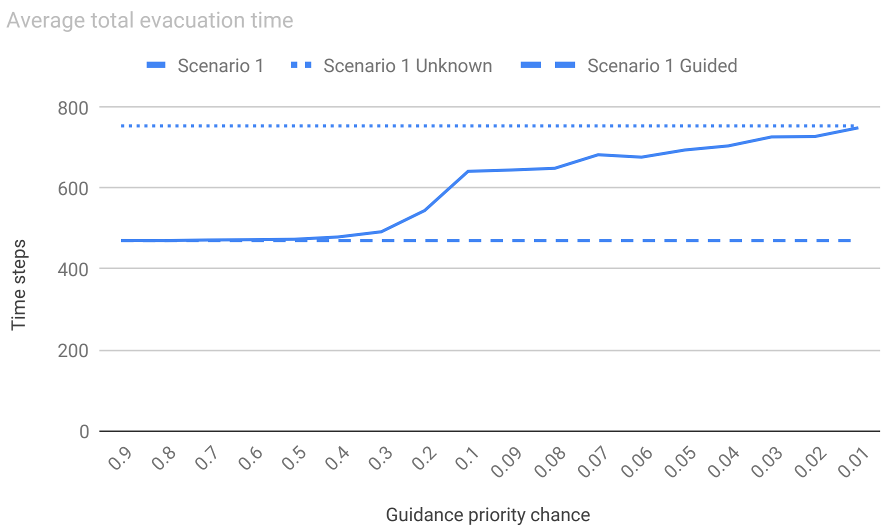


Figure 4.23: ET 3: Comparing evacuation time results of scenario 1

As the guidance priority chance was reduced for scenario 2 and 3, the swarm agents were finally able to complete the scenarios, however their evacuation times did not follow the same pattern as in scenario 1. Instead, they had their best results when the guidance priority chance was between the value when they first were able to complete the scenario and the lowest value tested. This could be explained by that a guidance priority chance too high causes the infinite loop problem in these scenarios, given the current hazardous areas, while a low priority chance is almost equal to the unknown environment. By finding the spot in between these values, the swarm agents are able to follow the guidance directions to some degree and avoid the infinite loop problem. So even though the environment had hazards, it was still beneficial to follow the guidance directions to some degree.

To visualize the movement patterns of the swarm agents when the guidance priority chance was reduced, some heat maps of the scenarios were created. Figure

4.24 shows a selection of the heat maps for scenario 2 and figure 4.25 show another selection for scenario 3. The full collection of the heat maps can be found in appendix .2.1.

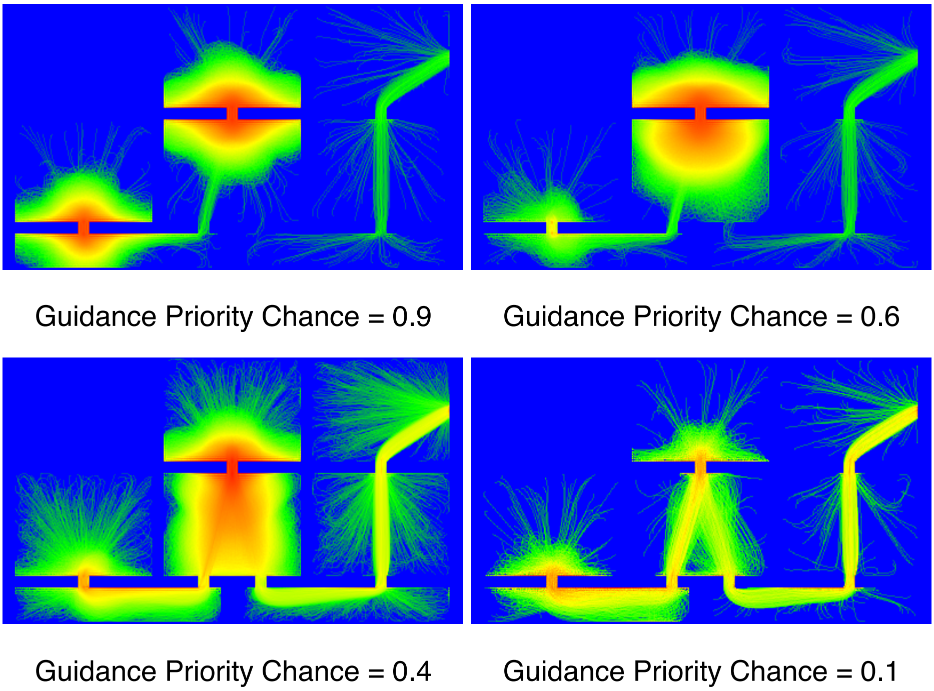


Figure 4.24: ET 3: Selected heat maps of scenario 2 from guidance selection chance test

Looking at the heat maps of scenario 2 with guidance priority chance 0.9 and 0.6, the infinite loop problem can be viewed as the red areas where the swarm agents moved back and forth through the same hallways. As the guidance priority chance is reduced these areas become bigger as a result of more variance in the swarm agents movement. At 0.4 in chance they are able to finish the scenario, however most of the activity is still near these hallways since the swarm agents still have a high chance of prioritizing them. When the chance is reduced to 0.1 most of this activity is gone because of the additional variance in the hallway selection and the total evacuation time is also reduced.

The first heat maps of scenario 3 show that the swarm agents are stuck moving between the two hallways all the way down to 0.4 in guidance priority chance. While they are still not able to evacuate with a chance of 0.4, it is noticeable

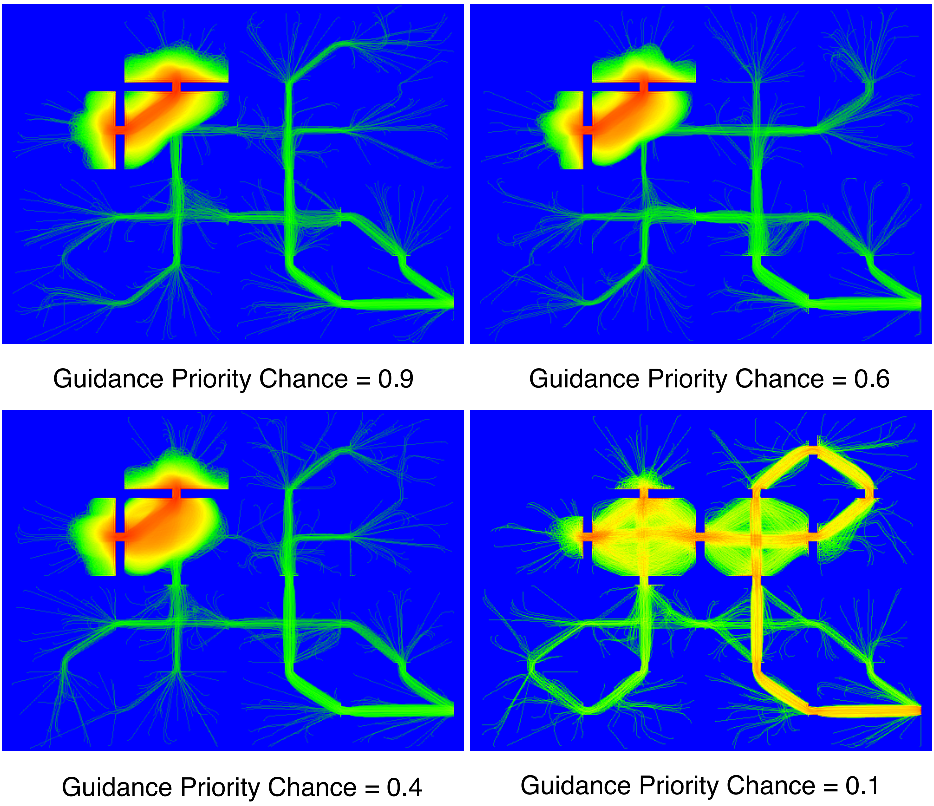


Figure 4.25: ET 3: Selected heat maps of scenario 3 from guidance selection chance test

that the movement patterns are wider because they move towards the other two hallways as well. At 0.1 in guidance priority chance their hallway selection is more random and they are able to evacuate.

The results from this test showed that the swarm agents could not follow the guidance directions blindly when there were hazardous areas present in the evacuation scenarios, as the swarm agents could get stuck in infinite movement loops between hallways. The second part of the test found that reducing the guidance priority chance of the evacuation objective allowed the swarm agents to finish the scenarios and that it was still beneficial to follow the guidance directions to some degree.

Experimental Test 4: Guided Environments With Dynamic Hazards

During an evacuation areas can become hazardous at any point in time. The purpose of experimental test 4 is to evaluate how this would change the evacuation patterns of the swarm agents. The selected areas presented at the beginning of phase 2 were set to become hazardous when 50% of the swarm agents had evacuated from the scenario. 50% was assumed to be a good condition because the swarm agents would be in the middle of the evacuation process at this point.

Experimental test 3 found that guided environments with hazards could lead to infinite movement loops between hallways if the guidance priority chance was high. In order to handle this, the guidance priority chance was set to **0.09** for experimental test 4. This value was selected because it gave good results for both scenario 2 and 3. Scenario 2 and 3 were prioritized over scenario 1 for this decision because they were the scenarios that encountered this problem.

Three simulation runs, one for each scenario, using 150 simulation iterations were run for this test, for a total of 450 simulations. The average total evacuation times and standard deviations from the simulation runs are presented in figure 4.26.

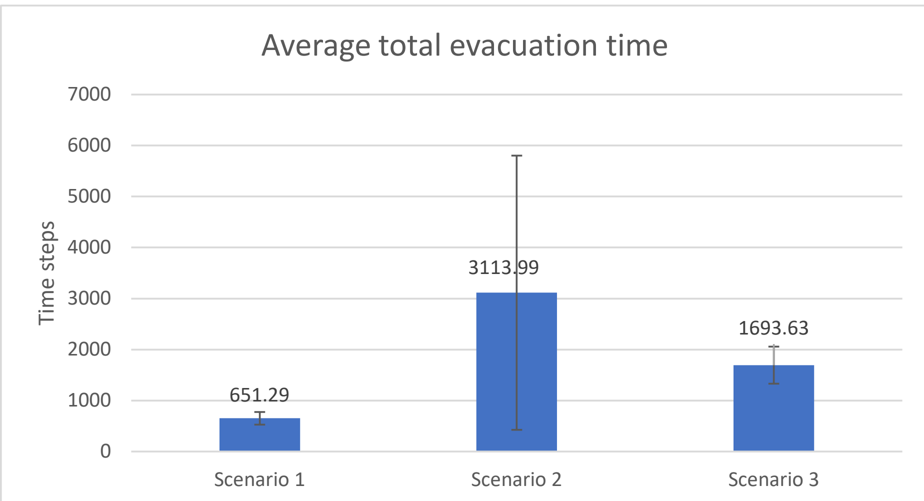
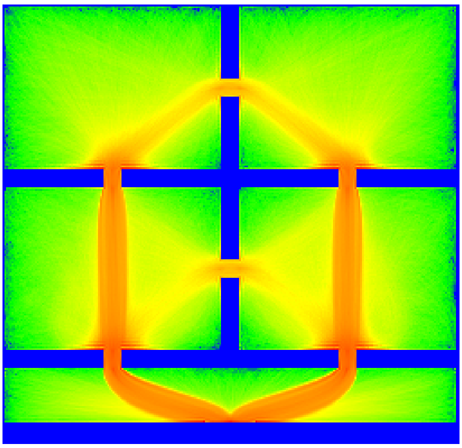


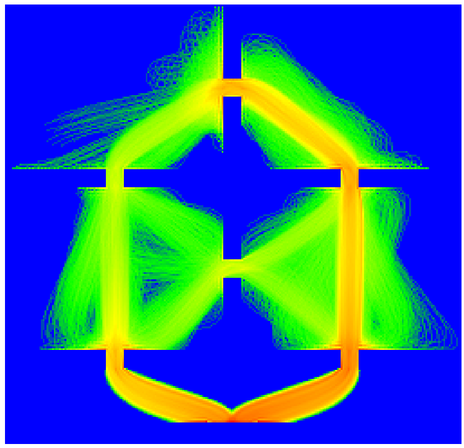
Figure 4.26: ET 4: Evacuation time results

The standard deviation of scenario 2 is very high, however by looking at the standard deviations presented in figure 4.22b from the previous test of the guidance priority chance, the standard deviations of scenario 2 were also this high. The large value of the standard deviation is a result of some of the simulation iterations having high total evacuation times. This is likely because they spent much time searching for an exit going back and forth between the same areas in the scenario.

To better understand how the swarm agents behaved in the dynamic environment heat maps were created. Instead of making heat maps based on the entire simulations, they were divided into two parts: the movement of swarm agents before the areas became hazardous and the swarm agents movement after the areas became hazardous. The heat maps of the scenarios are shown in figures 4.27, 4.29 and 4.30.



(a) Before hazard



(b) After hazard

Figure 4.27: ET 4: Heat maps of scenario 1

Looking at the heat map of scenario 1 before the hazard, the swarm movement is similar to the one found in the guided environment without hazards. The minor difference being that the middle hallways are also selected to some degree because the guidance priority chance is 0.06 instead of 1.0. After the middle left area becomes hazardous during the evacuation, the movement pattern of the swarm changes. The swarm agents in the hazardous area moves toward one of the three hallways, while the swarm agents in the other areas move in a clockwise motion around this area, as illustrated in figure 4.28. This explanation is based on the activity the heat map shows after the area in scenario 1 becomes hazardous.

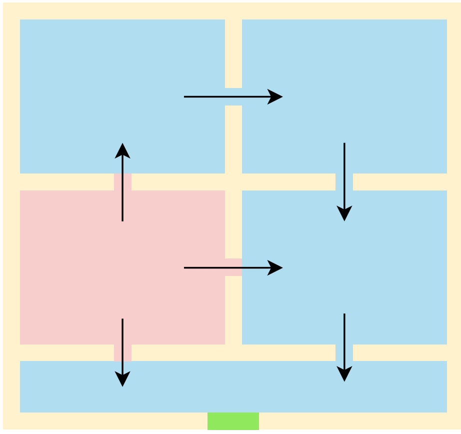
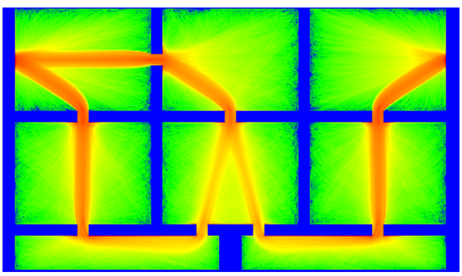
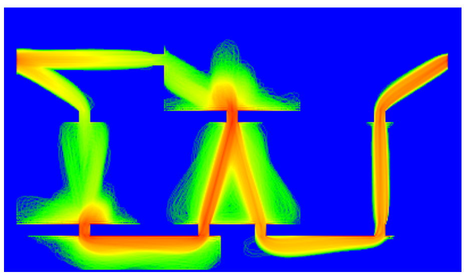


Figure 4.28: ET 4: Illustration of the swarm's movement after the area becomes hazardous in scenario 1

The heat map of scenario 2 before the hazard appears is similar the heat map of the unguided environment without hazards. After the top left area becomes hazardous, the heat map becomes similar to the ones shown in guidance priority chance test instead. Still some movement towards the exit is shown in the top left corner. This is the swarm agents that were in the area when it became hazardous. It is also apparent from the coloring of the heat map that most of the swarm activity were between the hallways with guidance directions, explaining the high total evacuation times observed in the standard deviation of scenario 2.



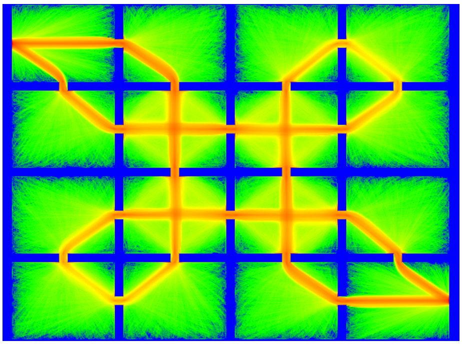
(a) Before hazard



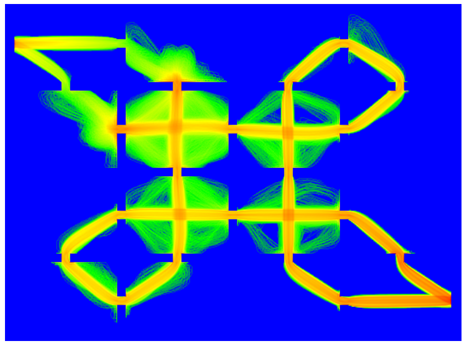
(b) After hazard

Figure 4.29: ET 4: Heat maps of scenario 2

Scenario 3 shows a similar behavior of the swarm agents before and after the top left area becomes hazardous. The only noticeable difference is that the swarm spends some time in the top left areas of the scenario, indicated by the green coloring of the area, but it is not as significant as in scenario 2.



(a) Before hazard



(b) After hazard

Figure 4.30: ET 4: Heat maps of scenario 3

Changing the guidance priority chance based on the tests performed in experimental test 3, the swarm agents were able to complete all the scenarios in experimental test 4. The heat maps from each scenario showed how the swarm changed it's movement to account for the hazard areas. As discussed in each heat map, the movement patterns changed differently depending on the scenario layout.

Experimental Test 5: Unknown Environments With Static Hazards

Experimental test 5 aims to observe how the swarm agents handle the different scenarios when the environment is unknown and there are hazardous areas. As with experimental test 3, the selected areas are hazardous from the beginning of the simulations and stay so for the entire simulation.

For this test three simulation runs using 150 simulation iterations, for a total of 450 simulations were performed, one simulation run for each scenario. The results from the simulation runs are presented in figure 4.31.

Again the standard deviation of scenario 2 is the most noticeable attribute about the graph. The chart presented in figure 4.32 shows one of the simulation iterations from scenario 2. From this chart it is clear that the total evacuation time

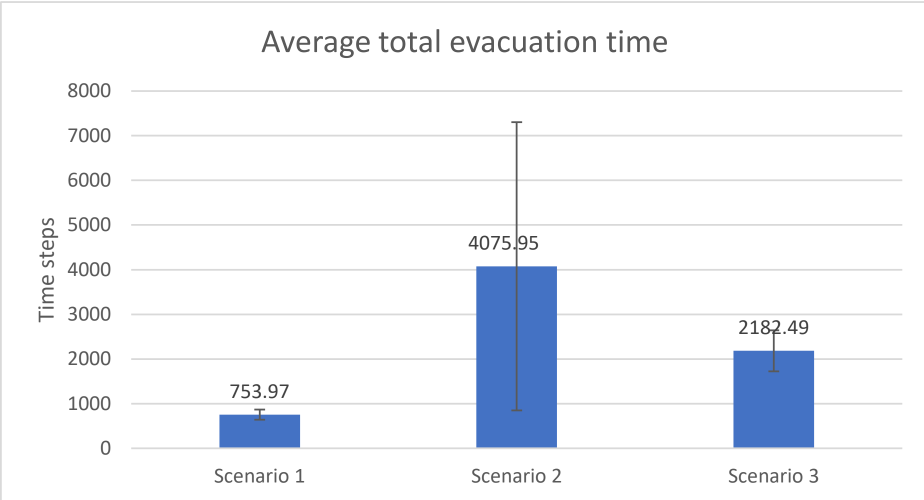


Figure 4.31: ET 5: Evacuation time results

was high because a few remaining swarm agents did not find the exit. When there are so few swarm agents left in the scenario, it is also debatable whether the swarm behavior is present.

To see how much the last swarm agents affected the total evacuation time, three new simulation runs were performed. The only difference of these simulation runs being that the simulations where stopped when 95% of the swarm agents had evacuated from the scenarios. A comparison of the initial simulation runs and these runs are presented in figure 4.33.

The comparison shows that there are only minor changes in the evacuation times for scenario 1 and 3, but a significant reduction in the evacuation time for scenario 2. Why the last swarm agents use so much extra time in scenario 2 is hard to say, but it is likely related to the implementation of the evacuation objective and how it selects the hallways to explore.

The heat maps from the first three simulation runs are shown in figure 4.34. It is noticeable that one or more swarm agents passed through the hazardous areas in scenario 1 and 2 during the evacuation by observing the small green lines in these areas. As explained in the model, the evacuation objective is passively avoiding hazard areas by not selecting them. Still some swarm agents went this way. It is possible that the swarm agents that went through were affected by the other swarm agents movement or that the collision avoidance altered their movement.

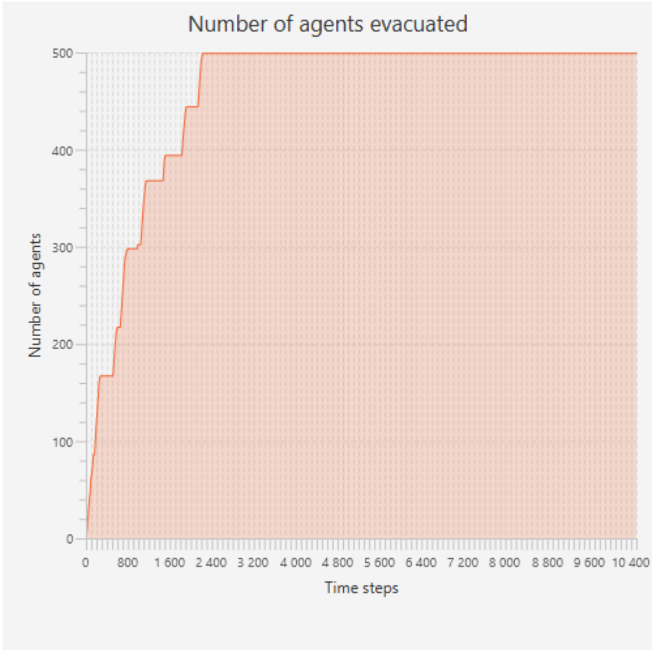


Figure 4.32: ET 5: Chart of scenario 2

Once they went through, the evacuation objective selected the hallway in the opposite end of the area because it required the shortest change in the swarm agents direction. Evaluating this behavior it would probably be better if the evacuation objective selected the closest hallway if the swarm agent went inside a hazardous area.

The heat map of scenario 1 shows that there was much activity in the top right area compared to the other areas, by looking at wide red area there. This is because the swarm agents went back and forth between this area, before finding their way down to the exit. The area in the top left corner of the heat map shows much activity around the hallway because the swarm agents that entered this area wanted to go back, as the evacuation objective only had one viable hallway to select in this area. The same movement pattern can be seen in scenario 2 and 3 in the areas connected to the hazardous area, where the swarm agents turn back around once they enter these areas. Other than the areas connected to the hazardous areas, the movement patterns of the swarm is similar to the ones presented in the unknown environments without hazards.

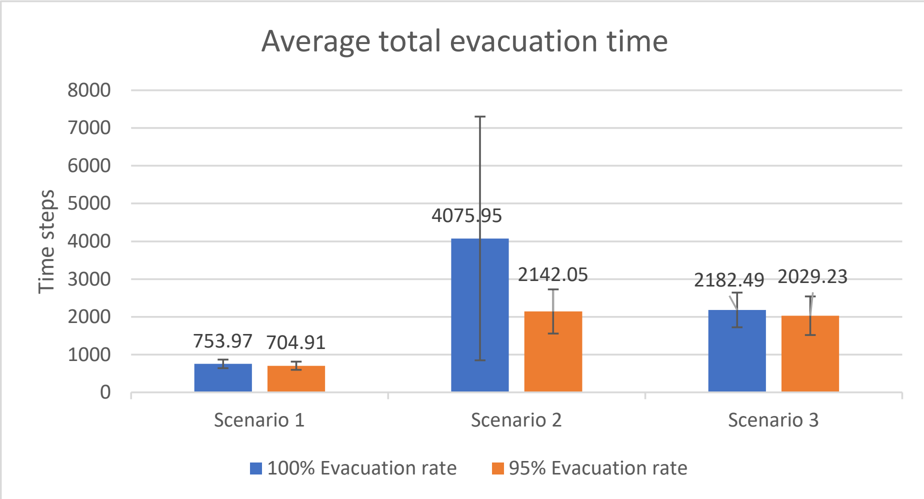
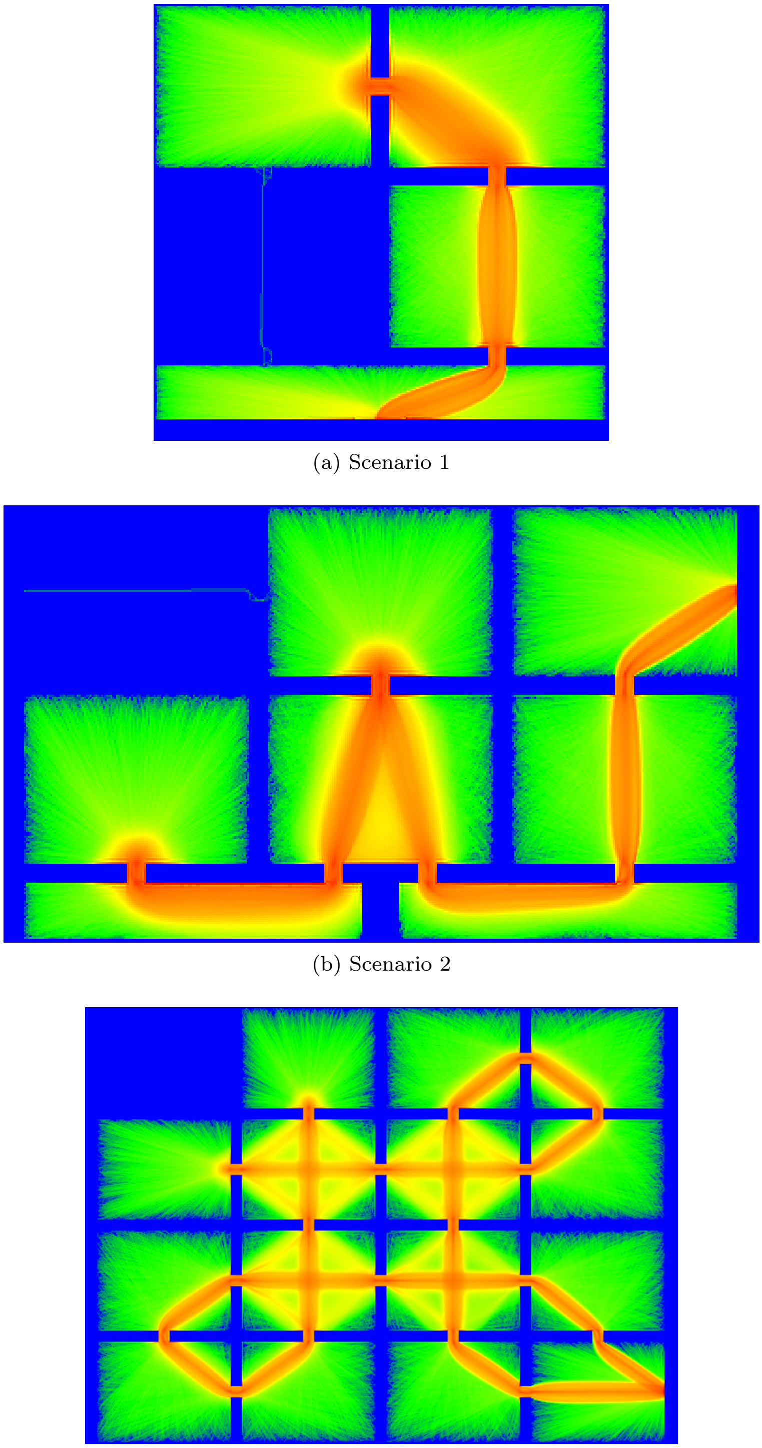


Figure 4.33: ET 5: Comparison of simulation runs using evacuation rates of 100% and 95%

From the previous experimental tests with hazards it was evident that scenario 2 used the most time to complete for the swarm agent model. In this test these results were followed up by an extra test to discover that it was only a few swarm agents that increased the total evacuation time by almost twice the amount. Given that this was only the case for scenario 2, it gave reason to believe that the hallway selection in the evacuation objective could be the problem. Looking at the heat maps it was also noticeable that some swarm agents went through the hazard areas, as a result of no active hazard avoidment in the model.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 4.34: ET 5: Heat maps of the scenarios

Experimental Test 6: Unknown Environments With Dynamic Hazards

So far it has been shown that evacuations in unknown environments are more difficult than they are in guided environments for the proposed swarm agent model when there are no hazards and that the swarm interaction from the boid model was beneficial for the evacuations. When the swarm model first was introduced to guided environments with hazards it was not able to complete two of the three scenarios, but by adjusting the guidance priority chance it was able to overcome this challenge. This was not a problem for the unknown environment with hazards in experimental test 5 because it did not follow the guidance directions in these environments from the beginning. These tests led to experimental test 6, where the environment is unknown and the hazards are dynamic, in that they appear when 50% of the swarm agents have evacuated from the scenario.

For each scenario one simulation run using 150 simulation iterations were performed, for a total of 450 simulations. The results are presented in figure 4.35 together with the results from the guided environment with dynamic hazards, from experimental test 4, for a comparison between the guided and unknown environments.

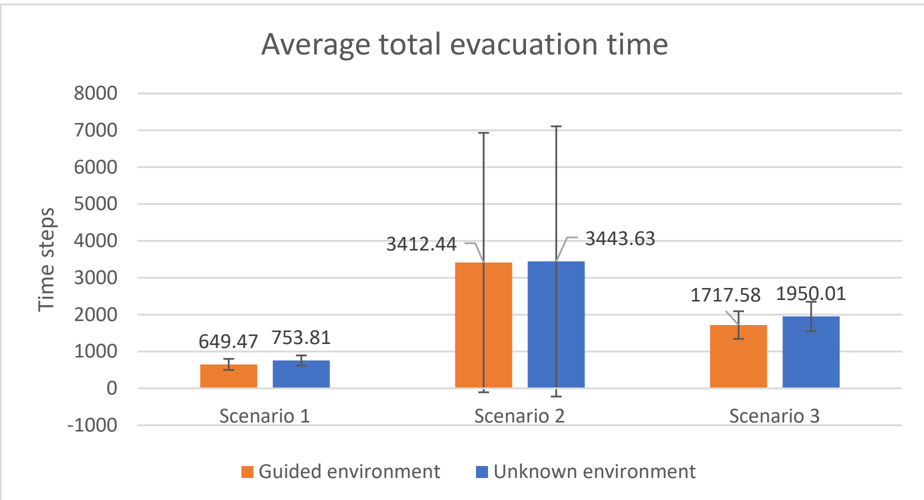


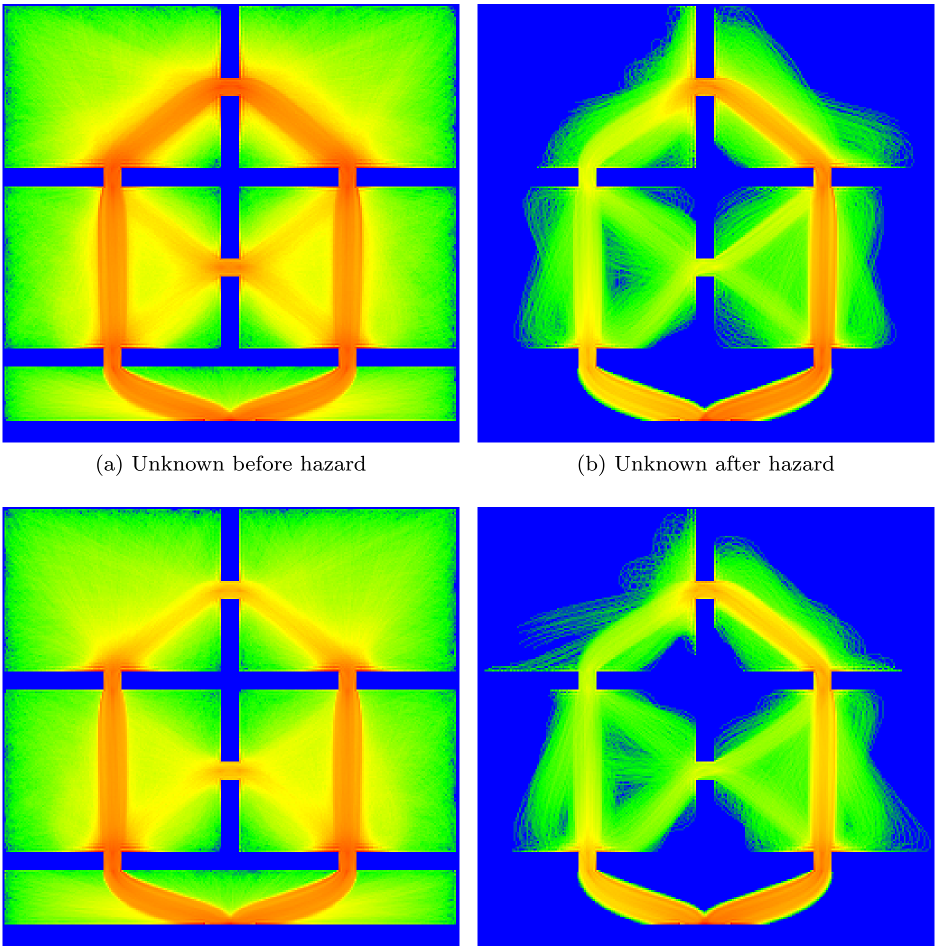
Figure 4.35: ET 6: Evacuation time results

A remark on the results is that the guided and unknown environments both have very similar evacuation times and standard deviations. However, it is important to remember that the results from the guided environment is after the guidance priority chance was reduced, giving a similar behavior equal to the model in the unknown environment. Still the guidance priority chance had to be reduced so that the model could be general enough to solve all the scenarios in the guided environment with hazards. The swarm agent model does therefore perform similar in the guided and unknown environments with hazards, with just a minor improvement in the evacuation times of the guided environment.

The heat maps of scenario 1 before and after the selected area becomes hazardous are shown in figure 4.36 together with the heat maps of scenario 1 from experimental test 4. Even though the evacuation times of the guided and unknown environments were quite similar, there are some noticeable differences between the heat maps. Looking at the heat maps before the area became hazardous, the swarm agents in the unknown environment shows more activity going back and forth between the areas in the top of the scenario. They do also use the hallway in the center of the scenario to a higher degree. These differences show the effect of the guidance priority chance even though it is small.

The heat maps of scenario 1 after the area becomes hazardous are more similar. The swarm agents in the unknown environment are still uncertain where to go in the top right and center right area of the scenario, because the evacuation objective has two possible hallways to select from in these areas, giving the extra activity shown.

The heat maps of scenario 2 and 3, illustrated in figure 4.37 and 4.38 shows the same trend as in scenario 1. There is some difference in that the swarm agents in the guided environment have a more directed movement towards the exits, but it is still similar to the one in the unknown environments. Environments with hazards introduce an extra difficulty to evacuations because the guidance directions can be inefficient, leading to hazardous areas. This challenge is reflected in that the model produces almost the same results for guided and unknown environments with hazards.



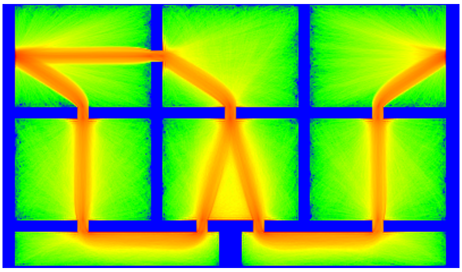
(a) Unknown before hazard

(b) Unknown after hazard

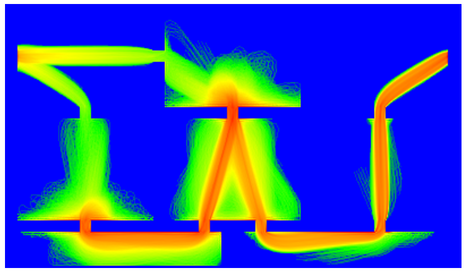
(c) Guided before hazard

(d) Guided after hazard

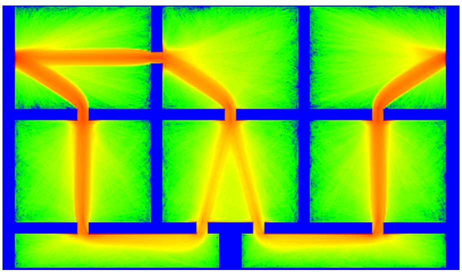
Figure 4.36: ET 6: Heat map comparison of scenario 1



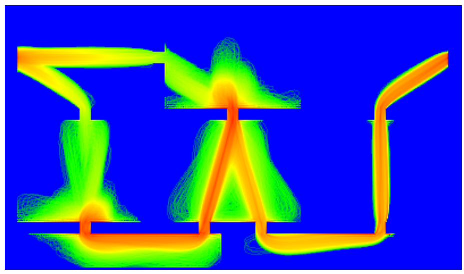
(a) Unknown before hazard



(b) Unknown after hazard

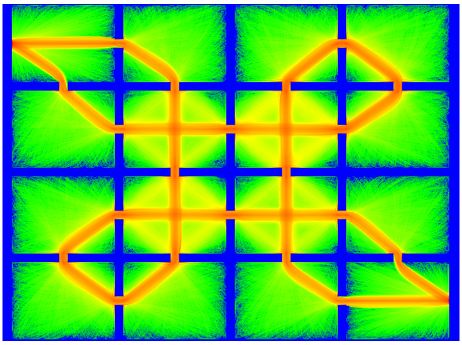


(c) Guided before hazard

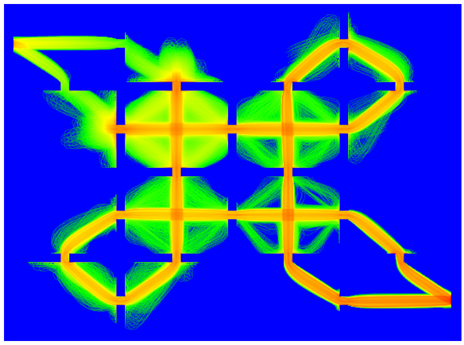


(d) Guided after hazard

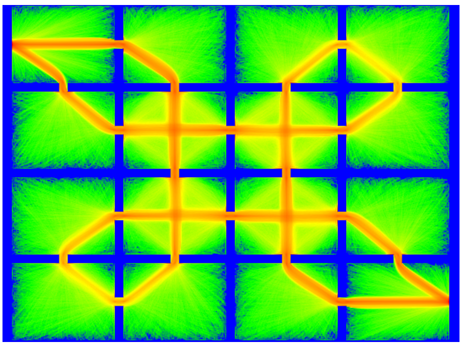
Figure 4.37: ET 6: Heat map comparison of scenario 2



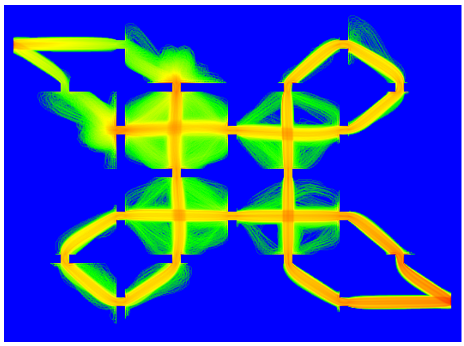
(a) Unknown before hazard



(b) Unknown after hazard



(c) Guided before hazard



(d) Guided after hazard

Figure 4.38: ET 6: Heat map comparison of scenario 3

Being able to solve evacuations in both guided and unknown environments with and without hazards between different scenarios, it is evident that the swarm agent model is able to generalize between the environments. When the hazards were introduced in phase 2 it became clear that the model could not follow the guidance directions blindly because they could lead to dead ends with hazardous areas. The final parameters of the swarm agent model after the experimental tests were completed are listed in table 4.5. The guidance priority chance is marked in blue because it was updated during the tests.

Simulator	
Parameter	**Value**
Simulation Iteration Count	150
Swarm Population Size	500
Initial Agent Distribution	Equal
Swarm Agent	
Parameter	**Value**
Ignore Room Chance	0.01
Max Acceleration	1.0
Proximity Range	10.0
Stuck Count Limit	10
Guidance Priority Chance	0.09
Evacuation Objective Weight	0.7
Separation Weight	1.0
Alignment Weight	2.0
Cohesion Weight	0.2

Table 4.5: Final Model Parameters

Chapter 5

Conclusions and Future Work

This chapter presents a discussion of the model based on the results from the experimental tests in the context of the research questions and research goal, followed by the contributions of the work herein. Finally, the future work for the swarm agent model is presented, suggesting improvements and additional possibilities that can be explored.

5.1 Discussion

The overall goal of this work has been to investigate how swarm agents can be utilized to solve the challenges of unknown environments with dynamic hazards in evacuation planning. For this purpose the goal was divided into three research questions:

Research question 1 *Can some global behavior beneficial to evacuations be obtained by applying the boid model?*

Research question 2 *How well do the swarm agent approach generalize in different environments?*

Research question 3 *How do hazards complicate the behavior of the swarm agent model in guided and unknown environments?*

During the course of this work these questions were taken into account both when designing the model and afterwards when the tests were made.

The experimental tests from phase 1 compared the model with and without using the boid behavior rules in both guided and unknown environments. From both tests the results showed that the swarm behavior from following the boid rules were beneficial for the evacuation times in every scenario. Two behaviors beneficial to the evacuations were noticed when using the boid behaviors:

1. In both the guided and unknown environments the swarm agents achieved a better flow in their movement, resulting in less collisions and faster evacuation times.

2. In the unknown environment the amount of individual exploration was reduced as a result of the flocking behavior directed the swarm agents movement towards where the majority was heading.

In phase 2 the swarm agent approach was tested against scenarios with both static and dynamic hazards in guided and unknown environments. When the swarm agents were put in a guided environment with hazards, it became clear that the guidance directions could not be followed blindly and that the model parameters had to be adjusted to account for this. In general the model had to become almost completely independent of the knowledge given from the guidance directions in order to complete the guided scenarios. The appearance of the hazardous areas also made it more difficult for the swarm agents to evacuate in the unknown environment, especially in scenario 2. From the tests with dynamic hazards, the heat maps from before the areas became hazardous showed movement patterns similar to the tests with no hazards. The heat maps from after the areas became hazardous were similar to the tests with static hazards, with the exception of the additional movement from the swarm agents that were in the areas that became hazardous and had to move out. Based on these results, two observations relevant to how hazards complicate the behavior of the swarm agent model in guided and unknown environments were found:

1. The global knowledge of the scenario represented in the guided environment can be misguiding and cause the swarm agents to get stuck next to a hazard area.

2. The model handles dynamic hazards just as good as static hazards. Before any hazards appear its behavior is equal to the one observed in tests without hazards and after they appear it is similar to the ones observed with static hazards.

Whether the swarm agent approach was able to generalize between the environments is a more difficult question to answer. A first remark towards this is that the suggested swarm approach was able to complete all of the scenarios given the different environments with and without hazards using the same model parameters. However, these tests were limited to three scenarios and only selected areas were set hazardous. While the results of scenario 2 with hazards showed that the swarm agents had a hard time to complete it compared to the same scenario without any hazards, they should not be directly compared because the complexity of the scenario changes once an area becomes hazardous and evacuation paths become blocked. A second remark on the generalization of the swarm agent comes from the results in phase 2 where it was shown that it had similar performance in the guided and unknown environments after the model was adjusted. A third remark positive to the models generalization was its observed behavior in the scenarios with dynamic hazards where the swarm agent adapted to the new environment effortlessly because of model simplicity.

With respect to the discussion of the research questions in relation to the test results, this work has found that applying the boid technique to a simple agent model can be beneficial for the global movement of the agents. While the swarm model was able to obtain better results in the guided environment without hazards, these results were based on model parameters that were not able to generalize toward environments with hazards, which is an important aspect in evacuations. For the suggested swarm approach, dynamic hazards were no harder than static hazards, but the scenarios with hazards were in general more difficult to complete than without.

5.2 Contributions

Through the work conducted in this thesis a swarm model following simple interaction rules was created. It was shown that this model, using only local information, was capable of solving evacuations in unknown environments with dynamic hazards.

Given the generality of this model it could be applied to other building scenarios to evaluate how their layout facilitates for evacuations when the environment is unknown and how the placement of guidance equipment would affect the evacuation process. It could also be applied to evaluate potential building layouts. In addition to this model and its application areas, it was also found that:

- The behavior from using the boid rules facilitate good movement flows in building scenarios, beneficial for avoiding collisions and congestions.
- The flocking behavior from the boid model creates a global motion that reduces the individual exploration in unknown environments.

5.3 Future Work

In addition to the model presented in this work, this section presents possible extensions that could be of interest for further research.

Possible work on parameters:

- **Parameter optimization**: Limited by the scope of this work some of the parameters in the model were not tested with other values and others could have been tested further.
- **Swarm size**: The population size is important in both in swarm approaches and evacuation scenarios and could be tested further.
- **Adaptive parameters**: The swarm agent could be able to change its parameters during the evacuation scenario.
- **Heterogeneous parameters**: The parameters could be individual for each swarm agent, giving unique behaviors.

Possible work on movement:

- **Random walk**: Introduce a second movement behavior in addition to the evacuation objective that gives variance from this movement.
- **Increased randomness over time**: As time passes introducing a degree of randomness in the movement could be beneficial for more exploration.
- **Active hazard avoidance**: There could be some additional behavior that actively avoided hazard areas.

Possible work on the swarm agent's perceptions:

- **Agents in proximity range**: The current model percepts all agents 360 degrees in the proximity range. It could be restricted to not percept the agents directly behind it to make it more realistic.
- **Blocked vision**: The perceptions could be blocked by the other agents, so that the evacuation objective would not be able to see the hallways in the current area if other agents were standing in front of it.
- **Equal perceptions**: The proposed model is always able to see the hallways in the current area, but this vision could be restricted to the hallways within the agent's proximity range.

Possible work on environment:

- **Faulty guidance equipment**: The guidance equipment could be damaged or stop working at certain areas.
- **Heterogeneous knowledge**: Parts of the scenario layout could be known to some agents. In combination with swarm it could be interesting to see the resulting global behavior.

Bibliography

[1] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In *Robots and biological systems: towards a new bionics?*, pages 703–712. Springer, 1993.

[2] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, 2(1):14–23, 1986.

[3] Rainer E Burkard, Karin Dlaska, and Bettina Klinz. The quickest flow problem. *Zeitschrift für Operations Research*, 37(1):31–58, 1993.

[4] LG Chalmet, RL Francis, and PB Saunders. Network models for building evacuation. *Fire Technology*, 18(1):90–113, 1982.

[5] Gen-Huey Chen and Yung-Chen Hung. On the quickest path problem. *Information Processing Letters*, 46(3):125–128, 1993.

[6] Lien-Wu Chen, Jen-Hsiang Cheng, and Yu-Chee Tseng. Evacuation Time Analysis and Optimization for Distributed Emergency Guiding Based on Wireless Sensor Networks. *2012 International Conference on Connected Vehicles and Expo (ICCVE)*, pages 130–135, 2012. ISSN 10459219. doi: 10.1109/ICCVE.2012.32. URL `http://ieeexplore.ieee.org/document/6519556/`.

[7] YL Chen and YH Chin. The quickest path problem. *Computers & Operations Research*, 17(2):153–161, 1990.

[8] Waltenegus Dargie and Christian Poellabauer. *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[9] E W Dijkstra. A note on two problems in connexion with graphs: Numerische Mathematic, 1959.

[10] Lisa Fleischer and Éva Tardos. Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23(3-5):71–80, 1998.

[11] Dario Floreano and Claudio Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies.* The MIT Press, 2008. ISBN 0262062712, 9780262062718.

[12] Lester Randolph Ford Jr and Delbert Ray Fulkerson. *Flows in networks.* Princeton university press, 1962.

[13] David Gale et al. *Transient flows in networks.* Rand Corporation, 1958.

[14] Morten Goodwin, Ole Christoffer Granmo, Jaziar Radianti, Parvaneh Sarshar, and Sondre Glimsdal. Ant colony optimisation for planning safe escape routes. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7906 LNAI:53–62, 2013. ISSN 03029743. doi: 10.1007/978-3-642-38577-3_6.

[15] H.W. Hamacher and S.A. Tjandra. Mathematical modelling of evacuation problems: A state of art. Technical Report 24, Fraunhofer (ITWM), 2001. URL `http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-12873`.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968. ISSN 0536-1567. doi: 10.1109/TSSC.1968.300136.

[17] Javier Izquierdo, I Montalvo, R Pérez, and VS Fuertes. Forecasting pedestrian evacuation times by using swarm intelligence. *Physica A: Statistical Mechanics and its Applications*, 388(7):1213–1220, 2009.

[18] John J Jarvis and H Donald Ratliff. Note some equivalent objectives for dynamic network flow problems. *Management Science*, 28(1):106–109, 1982.

[19] Dimitrios Kagaris, Grammati E Pantziou, Spyros Tragoudas, and Christos D Zaroliagis. Transmissions in a network with capacities and delays. *Networks: An International Journal*, 33(3):167–174, 1999.

[20] Thomas M Kisko and Richard L Francis. Evacnet+: a computer program to determine optimal building evacuation plans. *Fire Safety Journal*, 9(2): 211–220, 1985.

[21] Qingsong Lu, Betsy George, and Shashi Shekhar. Capacity constrained routing algorithms for evacuation planning: A summary of results. In Claudia Bauzer Medeiros, Max J. Egenhofer, and Elisa Bertino, editors, *Advances in*

Spatial and Temporal Databases, pages 291–307, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31904-7.

[22] Sabrina Merkel. *Building evacuation with mobile devices.* KIT Scientific Publishing, 2014.

[23] Sabrina Merkel, Sanaz Mostaghim, Daniel Blum, and Hartmut Schmeck. Distributed swarm evacuation planning. *Proceedings of the 2013 IEEE Symposium on Swarm Intelligence, SIS 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, pages 276–283, 2013. doi: 10.1109/SIS.2013.6615190.

[24] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, August 1987. ISSN 0097-8930. doi: 10.1145/37402.37406. URL `http://doi.acm.org/10.1145/37402.37406`.

[25] Craig W Reynolds. Steering behaviors for autonomous characters. In *Game developers conference*, volume 1999, pages 763–782. Citeseer, 1999.

[26] J Ben Rosen, S-Z Sun, and G-L Xue. Algorithms for the quickest path problem and the enumeration of quickest paths. *Computers & Operations Research*, 18(6):579–584, 1991.

[27] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009. ISBN 0136042597, 9780136042594.

[28] Fang Jing Wu. A Sensor-Assisted Emergency Guiding System: Sensor-Centric or User-Centric? *IEEE Transactions on Vehicular Technology*, 67 (2):1598–1611, 2018. ISSN 00189545. doi: 10.1109/TVT.2017.2749003.

A

.1 Preliminary Tests

.1.1 PT 2

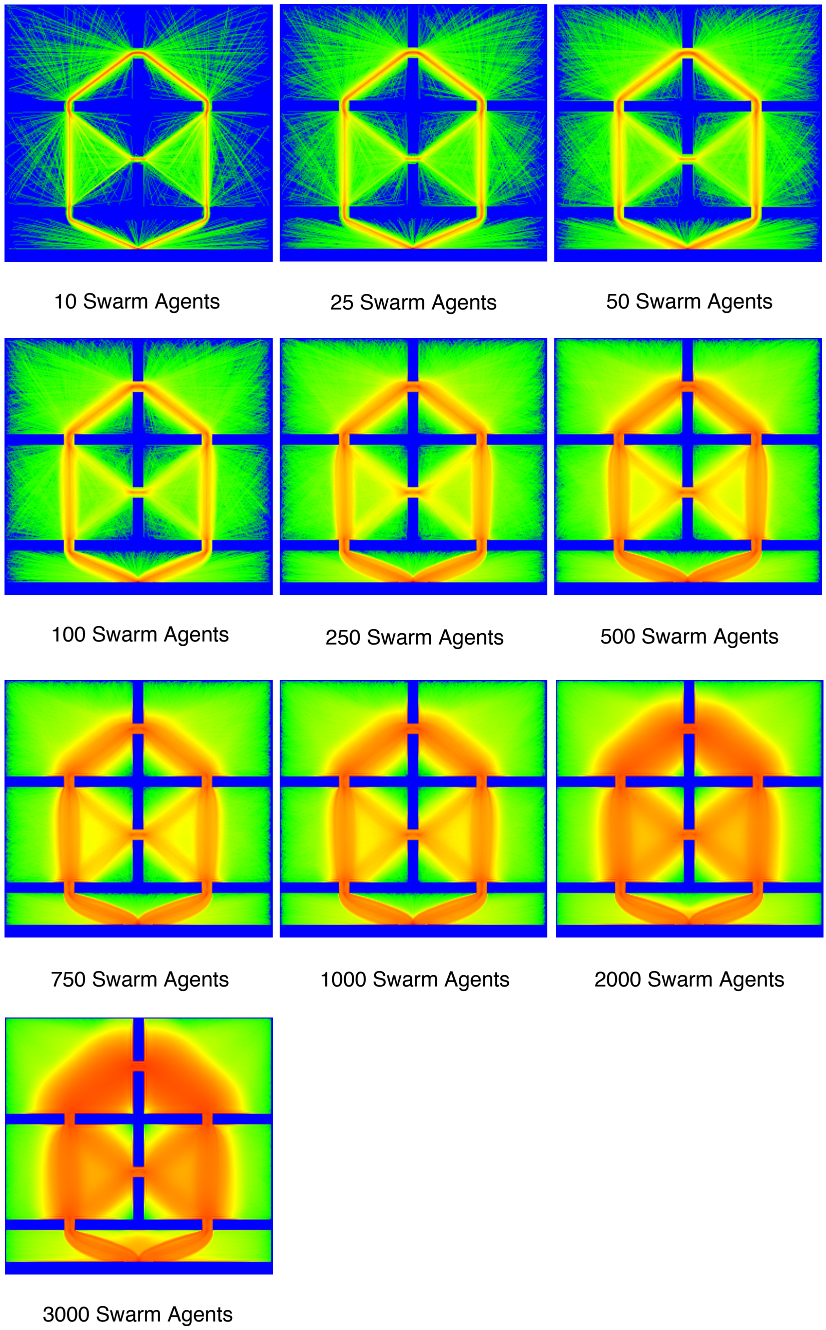


Figure 1: PT 2: Heat maps of scenario 1 using different population sizes

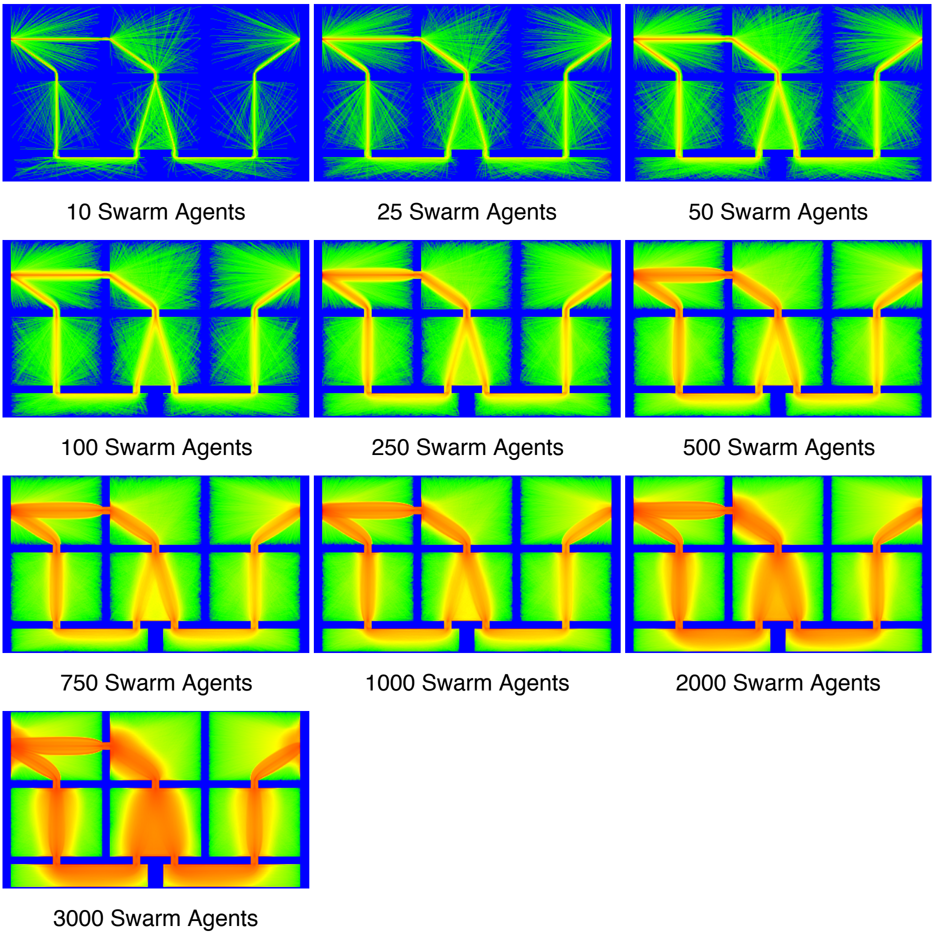


Figure 2: PT 2: Heat maps of scenario 2 using different population sizes

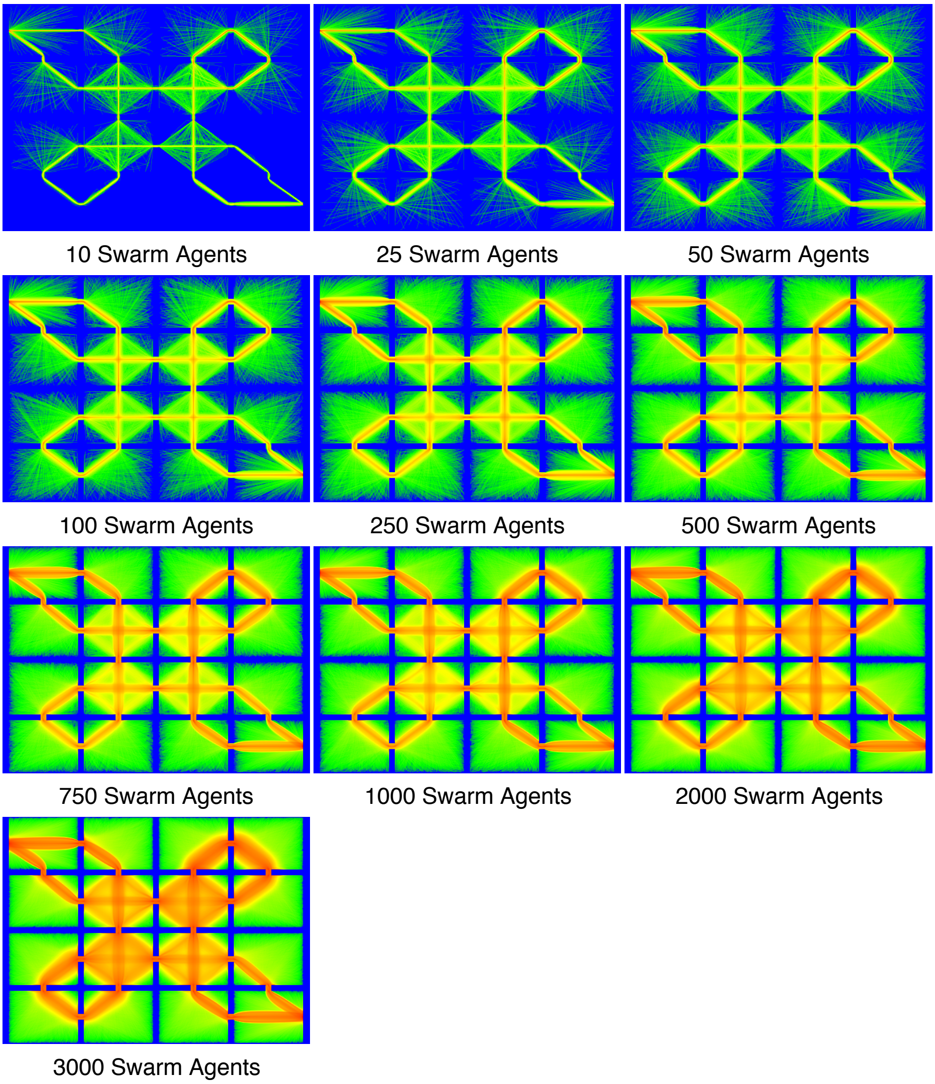


Figure 3: PT2: Heat maps of scenario 3 using different population sizes

.1.2 PT 5

Alignment Weight \ Cohesion Weight	0.1	0.2	0.3	0.4	0.5	0.6
0.1	258.87	396.49	421.15	3068.68	DNF	DNF
0.2	300.42	256.15	284.35	1196.19	DNF	DNF
0.3	278.64	166.43	214.42	513.35	DNF	DNF
0.4	344.21	159.67	179.66	342.33	DNF	DNF
0.5	228.47	269.94	170.09	280.38	DNF	DNF
0.6	329.60	146.23	146.08	308.65	1624.34	DNF
0.7	218.60	111.01	145.94	204.49	1291.49	DNF
0.8	189.78	131.86	124.35	184.31	911.63	DNF
0.9	206.13	131.85	120.25	176.96	758.39	DNF
1	202.47	125.98	123.59	167.85	589.84	DNF
1.1	202.34	105.04	129.17	162.49	DNF	DNF
1.2	159.26	157.36	159.25	133.75	403.05	DNF
1.3	121.69	132.03	119.27	128.99	DNF	DNF
1.4	115.63	120.47	119.38	132.22	497.39	DNF
1.5	147.03	135.11	115.22	154.41	DNF	DNF
1.6	139.08	120.95	131.89	144.21	DNF	DNF
1.7	168.52	125.98	114.77	141.81	397.76	DNF
1.8	157.40	136.44	369.48	134.20	DNF	DNF
1.9	163.70	127.03	603.40	168.32	1416.17	DNF
2	166.94	153.15	361.26	217.82	DNF	DNF

(a) Scenario 1

Alignment Weight \ Cohesion Weight	0.1	0.2	0.3	0.4	0.5	0.6
0.1	346.73	787.40	1012.15	DNF	DNF	DNF
0.2	701.76	487.91	567.98	2805.58	DNF	DNF
0.3	DNF	365.65	566.56	1296.92	DNF	DNF
0.4	DNF	451.50	293.57	955.81	DNF	DNF
0.5	DNF	306.96	320.87	707.47	DNF	DNF
0.6	3623.87	DNF	324.82	630.71	4938.72	DNF
0.7	579.46	DNF	424.23	492.59	3062.97	DNF
0.8	359.05	DNF	376.28	530.01	2050.84	DNF
0.9	247.24	3731.80	1004.57	580.25	2198.03	DNF
1	281.93	DNF	DNF	684.18	1423.54	DNF
1.1	295.21	2069.77	DNF	574.20	1123.93	DNF
1.2	258.99	469.10	DNF	653.80	1201.31	DNF
1.3	273.36	1435.40	DNF	1719.07	1124.27	DNF
1.4	250.66	571.34	DNF	3161.57	958.59	DNF
1.5	287.52	465.99	DNF	DNF	DNF	DNF
1.6	231.66	392.75	DNF	DNF	1297.36	DNF
1.7	259.56	322.81	DNF	DNF	1356.44	DNF
1.8	299.44	274.94	DNF	DNF	938.34	DNF
1.9	276.94	218.78	DNF	DNF	DNF	DNF
2	287.72	226.78	DNF	DNF	DNF	DNF

(b) Scenario 3

Figure 4: PT 5: Standard deviations of the evacuation time results

.2 Experimental Tests

.2.1 ET 3

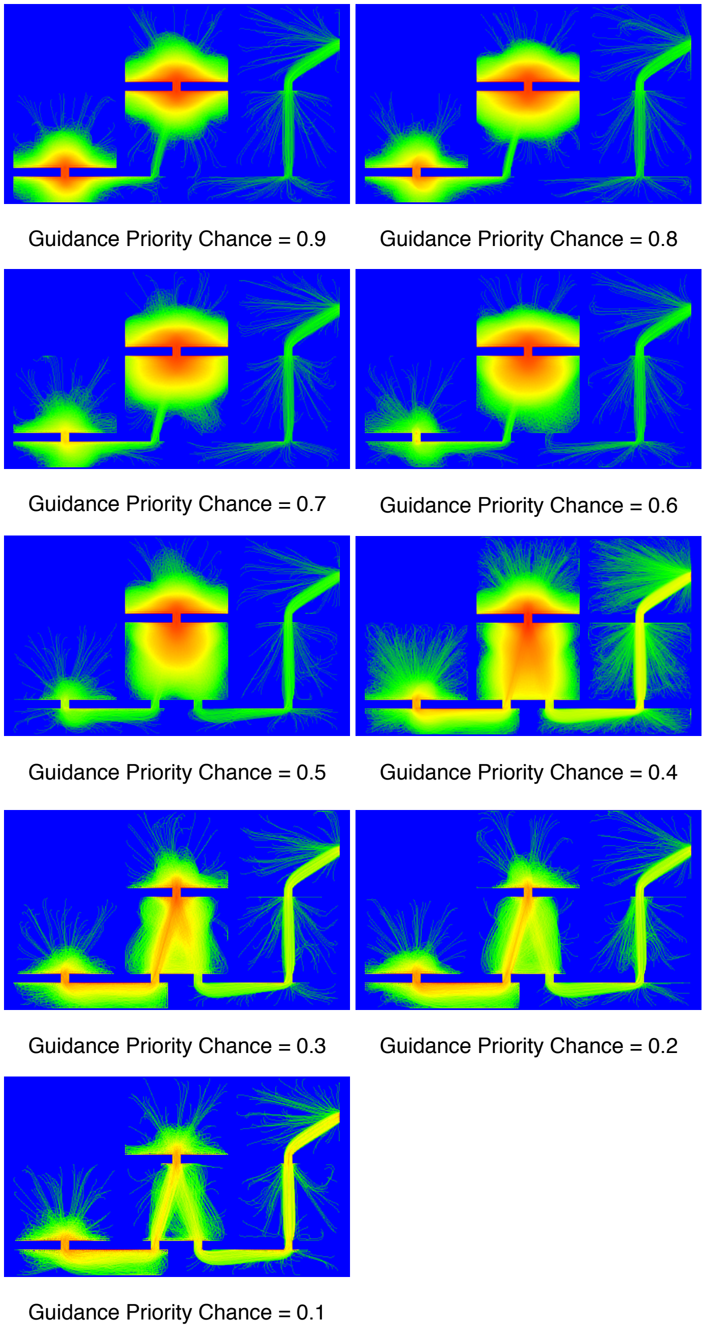
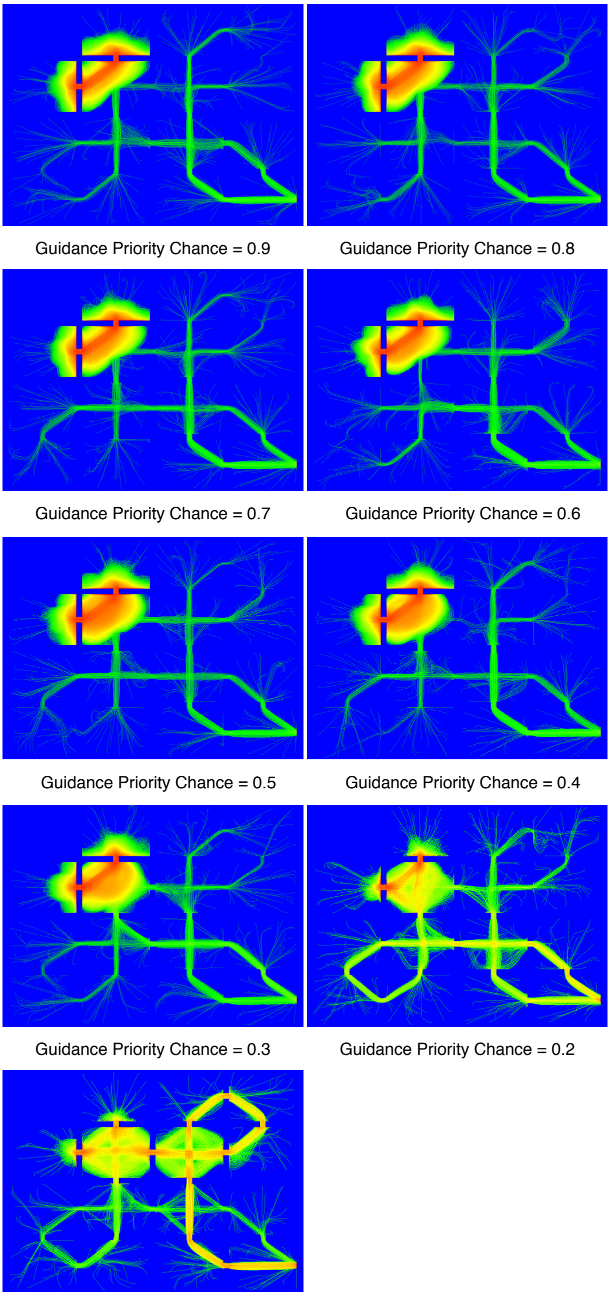


Figure 5: ET 3: Heat maps of scenario 2 from guidance selection chance test



Guidance Priority Chance = 0.9
Guidance Priority Chance = 0.8
Guidance Priority Chance = 0.7
Guidance Priority Chance = 0.6
Guidance Priority Chance = 0.5
Guidance Priority Chance = 0.4
Guidance Priority Chance = 0.3
Guidance Priority Chance = 0.2
Guidance Priority Chance = 0.1

Figure 6: ET 3: Heat maps of scenario 3 from guidance selection chance test

NTNU
Norwegian University of
Science and Technology