2018

Intergrating the Fruin LOS into the Multi-Objective Ant Colony System

Tirdad Kiafar
Technological University Dublin

Integrating the Fruin LOS into the Multi-Objective Ant Colony System

Tirdad Kiafar

D15126020

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Advanced Software Development)

2018

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Knowledge Management), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: ___________________________________

Date: **DD Month Year**

ABSTRACT

Building evacuation simulation provides the planners and designers an opportunity to analyse the designs and plan a precise, scenario specific instruction for disaster times. Nevertheless, when disaster strikes, the unexpected may happen and many egress paths may get blocked or the conditions of evacuees may not let the execution of emergency plans go smoothly.

During disaster times, effective route-finding methods can help efficient evacuation process, in which the directors are able to react to the sudden changes in the environment. This research tries to integrate the highly accepted human dynamics methods proposed by Fruin into the Ant-Colony optimisation route-finding method.

The proposed method is designed as a multi-objective ant colony system, which tries to minimize the congestions in the bottlenecks during evacuations, in addition to the egress time, and total traversed time by evacuees. This method embodies the standard crowd dynamics method in the literature, which are Fruin LOS and pedestrian speed.

The proposed method will be tested against a baseline method, that is shortest path, in terms of the objective functions, which are evacuation time and congestion degree.

The results of the experiment show that a multi-objective ant colony system performance is able to reduce both egress time and congestion degree in an effective manner, however, the method efficiency drops when the evacuee population is small.

The integration of Fruin LOS also produces more meaningful results, as the load responds to the Level of Service, rather than the density of the crowd, and the Level of Service is specifically designed for the sake of measuring the ease of crowd movement.

Key words: *Evacuation, Path-finding, Ant-Colony, Multi-objective, Fruin, Level of Service*

ACKNOWLEDGEMENTS

I take this opportunity to express my gratitude to my supervisor, Dr. Sarah Jane Delany, for her guidance and patience during the process of development.

I would like to thank Dr. Luca Longo, and Dr. Aneel Rahim for their time and effort, that made revising and publishing this research possible.

I would also like to thank my sister Shideh Kiafar, and her family for the support they provided during my studies. I could not have completed my studies without their support.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

LIST OF ABBREVIATIONS

AEC	Architecture, Engineering, and Construction
OOP	Object-Oriented Programming
DTO	data transfer object
ACO	Ant Colony Algorithm
TSP	Travelling Salesman Problem
Multi-ACO	Multi Objective Ant Colony Algorithm
MCDA	Multi-Criteria Decision Analysis
WPM	Weighted product model
LOS	Level of Service
ND-Set	Non-dominated solution set
MMAS	min-max ant system

1 INTRODUCTION

This chapter gives an overview of the scope of the research, the basic defining aspects of the research such as research question, and the objectives of the research. A short outline is also present at the end of the chapter.

1.1 Background

The path-finding during evacuation time is a process that tries to find proper paths for evacuees during disaster times. The proper path is defined by the objectives of the evacuation, which depends on the situation (Yi-fan, Jun-min, Jie, Ying, & Jin-hua, 2011).

From the major categories of the analytical models for human dynamics, optimisation is the mostly regarded method in the literature (Georgiadou, Papazoglou, Kiranoudis, & Markatos, 2007; Liu, Zhang, Ma, Pota, & Shen, 2016; Yi-fan et al., 2011). The reason behind this is that path-finding, due to complexity of the environment, the potentially high number of evacuees, and multiple objectives of an evacuation method, may have a humongous number of solutions, and finding the optimal path for evacuees is not feasible in this case. This is because of the high computational power needed to accomplish this and the limited time to react during hazardous times (Fang, Zong, Li, Li, & Xiong, 2011).

There are a variety of optimisation models used in the domain of emergency evacuation and some of them are able to address multiple objectives like egress time and congestion degree (Gwynne et al., 1999).

The Ant-Colony Optimisation (ACO) method is one of the common methods used in path-finding, and it has the ability to address multiple objectives. The emergency path-finding has its own needs, which ACO can address. These needs are listed below (Yi & Kumar, 2007):

- Ability to address any variety in population distribution.
- A need to produce different paths from different rooms to the exits.
- Possibility to generate statistics about evacuation, including egress time and congestion degree, which is an important risk factor that can help planning.

- Fast response. Method needs to produce results fast. This also ensures that any unforeseen changes in population density and distribution can be addressed quickly.
- Ability to optimise not only the path-length, but the capacity of the walkways, the risks associated to the environment, and the capacity of the exits. This is the multi-objective requirement for the model, and objectives depend on the evacuation situation.

1.2 Research Problem and Methodology

The research question is as follows.

Does integrating Fruin crowd-dynamics analysis methods into a Multi-Objective Ant Colony system result in a versatile Multi-ACO system?

This research tries to utilize widely accepted methods of congestion-degree and agent speed formula, that is Fruin methods, into Multi-ACO system and find out whether the results are able to address the path-finding in disaster time.

The versatility of the proposed system will be measured against a baseline method, that is shortest path method, described in the chapter two.

This research is defined as a primary quantitative research. It will implement the solution as a software suit and will gather the needed data from a battery of simulations.

It is expected that the proposed method gives sub-optimal results, as it is optimisation, which show drastic improvements from the baseline method. It is also expected that the current method produces more readable results for the congestion degree analysis.

1.3 Research Objectives

The objectives of the research are defined as follows:

- To develop a Multi-ACO solution capable of optimising congestions alongside egress time.
- Implement Fruin evacuee-speed formula into the Multi-ACO path-finding process.
- Implement Fruin Level of Service (LOS) congestion degree evaluation method into the Multi-ACO solution domination evaluation process.

- Measure the improvement of the results against the baseline method.

- Measure the feasibility of the method for a variety of different evacuation scenarios, including random population, random crowd distribution, and random enclosure types.

1.4 Scope and Limitations

The emergency evacuation planning process is complex and time consuming. This is due to variable condition such as population of evacuees and their distribution. The computer analysis models are great tools to aid emergency planners, and optimisation models that are able to optimise multiple criteria such as egress time and the congestion degree have been a major research area due to their good performance and acceptable results. The base method in this research is Multi-Objective Ant-colony optimisation (Multi-ACO).

It is mandatory to mention that the proposed method is not a precise simulation method, in terms of mimicking the human behaviour, and can only tackle the problem of path-finding during evacuation.

It is also important to note that each evacuation scenario needs different objectives, such as dynamic path blocking during a fire breakout, or the need for considering the risk of paths near the hazardous area. The evacuation scenario in this research does not address such objectives, hence the scenario can be defined as emergency evacuation with intact environment, for example evacuation due to an alarm of possible missile attack is one scenario that has the environment intact.

1.5 Document Outline

Chapter two, literature and work done, contains the description of evacuation science, the previous work done in the area, and the technologies used for the experiment.

Chapter three describes the design of the experiment, containing the mathematics of the Multi-ACO method, the software design part, and the main aspects of the Multi-ACO method, including enclosure design, and the population groups.

Chapter four contains the results of the experiment, the discussion about the results and the conclusion.

2 LITERATURE REVIEW

This chapter describes the algorithms, terms, tools, and technologies used in the research. The related work is also explored at the end of the chapter.

2.1 Evacuation Science

The rapid worldwide growth of population in the 20th century caused development of various new scientific branches in AEC industry (Abrahams, 1994). Safety was one of the popular branches that saw many updates during the last 60 years.

The first step of dealing with building safety is enforcing the architectural standards, including acceptable dimensions for building components and emergency components (Neufert, Neufert, & Kister, 2012).

The fact that disaster may strikes any time implies the need for preparation. Regardless of the type of disaster, the population in the affected area should be able to escape to a safe area efficiently.

Emergency evacuation is the act of urgent escape of people from an area where a threat to life or property exists. The hazardous condition are defined by the warning level, and based on the warning level, evacuation is divided in two categories, precautionary and life-saving operations (Saeed Osman & Ram, 2017).

Evacuations may be carried out before, during or after disasters such as natural disasters, traffic accidents, industrial accidents, fire, military attacks, structural failure, viral outbreak (Abrahams, 1994).

Evacuation is the last step of a chain of safety measures in AEC industry, which is accompanied by logistic support, which is out of the domain of this research. The following section tries to give some insight on the process of planning for disaster time.

2.1.1 The Emergency Response Plan (ERP)

During the disaster time, action taken in the first minutes are quite critical. Issuing a quick evacuation warning to employees could save many lives. Also, actions taken by employees with a proper knowledge of the building and the production process may control a leakage, and hence the damage to the facility and environment (Department of Homeland Security, n.d.).

Emergency response plan (ERP) contains information on different types of emergency measures, provided for specific premises, and includes building plans and evacuation procedures. It is accompanied by a set of instructions for specific types of disasters to cover different disaster scenarios (Singapore Civil Defense Force, 2013).

Current ERP development approach is called proactive planning, which involves developing and examining different plans in advance for different scenarios and finding among the available plans the most suitable one to be used whenever an incident occurs (Department of Homeland Security, n.d.).

According to Singapore evacuation planning guidelines (2013), the complexity of an Emergency Response Plan is dependent upon the following factors:

- The size of the premises and complexity of routes.
- The premises height.
- The number of occupants.
- Premises type.
- P&FM and Hazardous materials storage.
- Special risk associated with the premises.

2.1.2 Building Egress Analysis

The process of evaluating the evacuation paths in a building is called egress analysis. This analysis gives a quantitative complexity rank of an enclosure. For designing the means of egress in a building, there are international instructions in the architectural bible. there are also instructions from the fire-safety community, with detailed process for Egress analysis (Shen, 2006).

The elements of a means of egress system are listed here (BuildingCodeNYC, 2013):

- Exit Access
    - Dead end corridors
    - Travel distance
    - Number of exits
    - Distance between exit doors
    - Exit signage
- Exit
    - Exit stair width
    - Landing depth
    - Exit passageway width
- Exit discharge
    - Exit door width, and exit capacity

These elements cover the parameters used in evacuation path-finding methods, described later on in this chapter.

2.1.3 Bottlenecks During Evacuation

The evacuation process, in its basic form involves people moving toward a target. The fact that the process is done under pressure implies that the stress may have a negative impact on the flow of the event, including increased rate of accidents that may harm evacuees, however, the main risk is caused by the congestions formed in bottlenecks (Yi-fan et al., 2011).

A crowd of people behind a bottleneck is usually more dense than normal situations. The quality of how dense a crowd is and its relation to the crowd flow has been another research are that led to different quantitative and qualitative methods to measure the fundamental aspects of crowd flow (Yi-fan et al., 2011). This issue is thoroughly dis-cussed in chapter two.

The egress analysis mentioned before is aimed at finding the bottlenecks, however, the traditional egress analysis does not address small details of the building component (Shen, 2006). The computer analysis of the evacuation boasts simulation methods that are able to address this issue, and this matter is discussed in the next sections.

2.2 Computer Analysis of Evacuation

Before computing power became available to the public, analysis of evacuation was limited to qualitative analysis of egress components of the building, and limited quantitative methods to evaluate complexity of evacuation routes, regarded as egress analysis (Shen, 2006).

This limitation comes from the fact that simulating egress process needs to be repeated many times to get a reliable answer (Gwynne et al., 1999), which is not feasible for real-life experiment. To prove this, please tend to the hypothetical distribution of total evacuation time for a combination of a single building-population-environment shown in Figure 1.

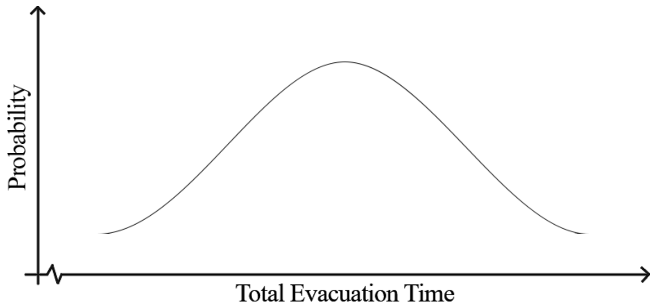


Figure 1.1. Distribution of total evacuation time for an evacuation drill (Gwynne et al., 1999)

With the growth of computational power, more complex analytical methods emerged to address complex problems of human dynamics and complicated path-finding in big facilities. The advantage of these models are the ability to run multiple times with low cost (Shen, 2006).

There are three major analytical categories which tackle evacuation problem in different manners: simulation, optimisation, and risk assessment (Gwynne et al., 1999).

Risk assessment is done mainly in ERP planning stage to understand the critical paths and areas, which helps revising the plan and allocation of assets during evacuation, such as helping hands, and first aids (Gwynne et al., 1999). Traditional risk assessment methods consider pre-disaster conditions such as the vulnerability and accessibility of the road network. Modern methods also take into account post-disaster factors including the impact or aftermath of the disaster and evacuee's routing behaviour (Chien, Wu, & Huang, 2014).

Simulation models represent the behaviour and movement during evacuation, not only to get acceptable quantitative results, but to represent paths and decisions during an evacuation (Gwynne et al., 1999).

Optimisation methods assume that occupants evacuate in an efficient way, ignoring peripherals and non-evacuation activities (Gwynne et al., 1999). These models treat occupants like a homogeneous ensemble, hence they are used for large number of people.

The method used in this research is multi-objective ant colony algorithm (Multi-ACO), which is an optimisation model.

Assessment of an evacuation model is possible by addressing the following criteria (Gwynne et al., 1999):

- Configuration: General and traditional building codes, including building layout, the type of building, number of exits, width of exists, travel distance.
- Environment: Impact of live load, that is obstacles such as shelves and disaster time factors such as effect of heat, smoke and toxic gasses on way finding capabilities.
- Procedures: Number of trained staff, level of training of evacuees, occupant's prior knowledge of enclosure, emergency signage, etc.
- Behaviour: The mental impact of forced evacuation on individuals vary and impacts the efficiency of evacuation drastically. Factors like age, mental fortitude, personal knowledge of the environment also affect the behaviour deeply.

These factors relate in evacuation process as shown in Figure 1.2.

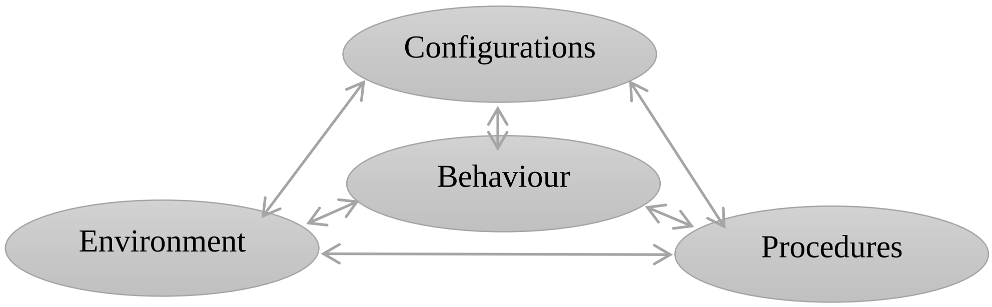


Figure 1.2. Interacting aspects of an optimal design enclosure (Gwynne et al., 1999)

Evacuation models can be divided based on how they represent enclosure (space), population (evacuees), and evacuee behaviour (Gwynne et al., 1999; Shen, 2006).

2.2.1 Enclosure Perspective

Two models are widely used to represent space.

Fine Grained Network: Space is presented as tiles, for example. Exodus, a simulation software, uses .5m * .5m tiles, where Simulex, another simulation software, uses .25m * .25m. These models usually use particle simulation, either agent based, individual evacuee simulation or gas simulation (Chen et al., 2012).

Coarse Grain Networks: Define geometry as partitions derived from actual structure, results in a graph in which nodes are rooms and arcs are connections (Gwynne et al., 1999).

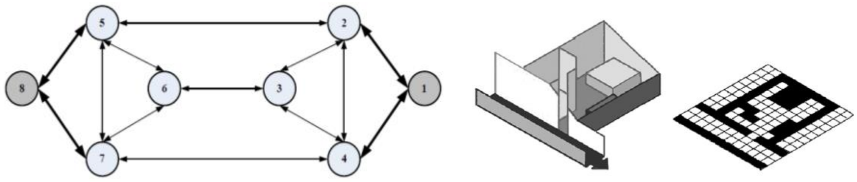


Figure 1.3. Coarse grain network (left) vs fine grain network (right) (Chen et al., 2012)

2.2.2 Population Perspective

Two models are used to represent a population of evacuees: individual (agent based), and global perspective (Gwynne et al., 1999).

Most agent based models allow user to assign personal properties manually or randomly (Gwynne et al., 1999). Then the behaviours is determined by a rule based or AI engine (Santos & Aguirre, 2004).

Global models treat evacuee population as a homogeneous ensemble without personal identities, thereby this approach is called global perspective.

The advantage of agent based models is that unlike global models, they are able to determine the effect of events on individuals during evacuation, however, the accuracy depends on how Comprehensive the model is (Gwynne et al., 1999). Agent based models are usually very CPU greedy, so they cannot be used to determine routes during disaster times (Chen et al., 2012).

2.2.3 Behaviour Perspective

Evacuation models need a decision-making system, to move the population inside the enclosure. This aspect is known as population behavior system. Various behavior systems are listed below (Gwynne et al., 1999):

- No behavioral rules: Model relies on physical crowd movement.
- Functional analogy behavior: Applies a single, or a set of equations to the whole population. These equations are derived from other fields of study, such as magnetic crowd movement model derived from physics, rather than actual human behavior.
- Implicit behavior: Do not declare behavioral rules, but assume them to be implicitly represented through complicated physical models, derived from secondary data. Their accuracy is dependent on the relevance and accuracy of secondary data.

- Rule based behavioral systems: Most models which allow individual behavior, define this behavior with a set of rules, which are applied in certain circumstances, e.g. in a smoked filled room, I will leave through the nearest exit.
- Artificial intelligence based behavioral systems: Try to mimic human behavior. Accuracy depends on how well-trained the AI engine is.

During evacuation, behaviour has a complex relationship with the surroundings. A person may have human to human, human-structure or human environment interaction (Santos & Aguirre, 2004).

These interactions affect occupant's movement and therefore trigger decision making process. These interactions happen in three levels: Psychological, like fear; Sociological, like alarming another occupant, or teaming up; and Physiological, like intoxication because of irritant gasses during fire breakout (Santos & Aguirre, 2004).

2.2.4 Model Accuracy

Model accuracy, unlike previous factors, is not a role-playing factor in efficiency of the model. It is rather a measure of efficiency of the model, derived from the relationship of previous factors.

The single most lacking feature of all models is lack of a convincing battery of validation comparisons mostly because of a general lack of data suitable for validation (Gwynne et al., 1999). As there are very limited number of available public datasets, the matter seems more complicated considering the fact that an evacuation process is fundamentally affected by the following factors (Santos & Aguirre, 2004):

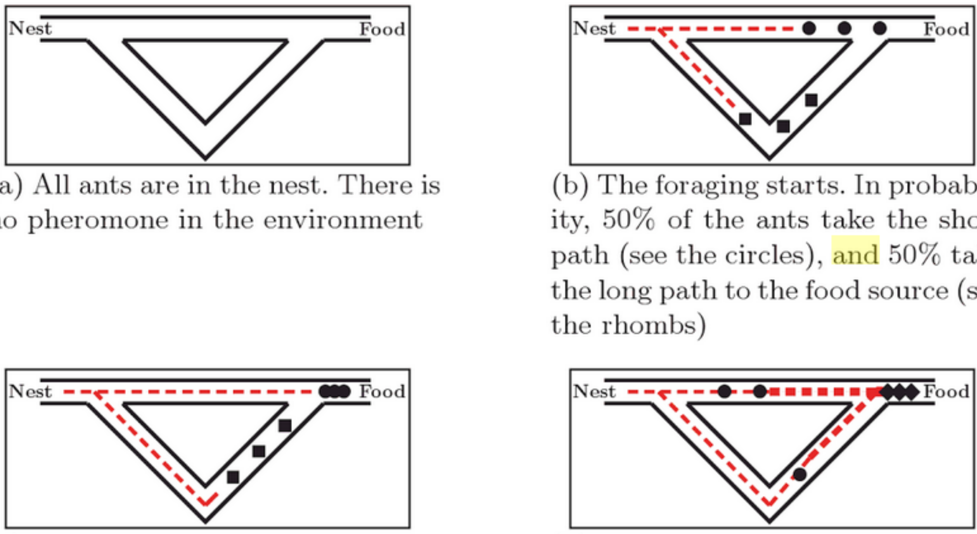
- Physical Nature of the Enclosure: number of floors, exits, their width, etc.
- Function of the Enclosure: offices, hospitals, schools, etc. have different characteristics, both for the occupants and physical nature of the building.
- Nature of the population: age, gender, level of training, etc.
- Nature of the Environment: time of day, debris, signage, etc.

Until a common systematic framework is adopted by the international fire-safety community this will remain the most important issue for both development and wide-scale acceptance of evacuation models (Gwynne et al., 1999), however, as the reports say (Gwynne et al., 1999; Pelechano & Malkawi, 2008) the generic evacuation simulation software suits have gained a fair accuracy by constant tweaking during the last 40 years, and many studies validate their data against the simulations by these methods, which are mostly agent based models.

2.3 Ant Colony Algorithms

Ant colony optimisation algorithm (ACO), introduced in early 90's, is an optimisation technique that is inspired by the real-life resource finding of ant colonies (Blum, 2005). The real-life ant colony path-finding works in such a way that each individual ant from the colony population leaves a chemical acid during traversing its path to food and back to home. The ants follow the strongest pheromone trail, and naturally, the shortest path to food gets more pheromone, because during a fixed time frame, more ants pass from the shortest path (Dorigo, Maniezzo, & Colorni, 1996).

This process is shown in Figure 2.1.



(a) All ants are in the nest. There is no pheromone in the environment

(b) The foraging starts. In probability, 50% of the ants take the short path (see the circles), and 50% take the long path to the food source (see the rhombs)

(c) The ants that have taken the short path have arrived earlier at the food source. Therefore, when returning, the probability that they again take the short path is higher

(d) The pheromone trail on the short path receives, in probability, a stronger reinforcement, and the probability of taking this path grows. Finally, due to the evaporation of the pheromone on the long path, the whole colony will, in probability, use the short path

Figure 2.1. An experimental setup to show the process of foraging by an ant colony (Blum & Merkle, 2008, p.47)

The digital ants in ACO are different from real ants in a variety of different ways, three major differences are listed here:

- Real ants move in an asynchronous way with their environment, where all artificial ants leave home, find destination and go back home in each iteration (Blum & Merkle, 2008).
- Artificial ants leave pheromone just on their way back, or better said only those that find the destination leave pheromone, for a one side trip. It is important to note that home to destination trip is the same as destination to home (Blum & Merkle, 2008).
- Where foraging behaviour of real ants are based on an implicit evaluation of the paths, that is the fact that shorter paths receive pheromone reinforcement quicker, the artificial ants have the possibility of explicit evaluation of paths based on more deciding factors that just pheromones (Blum & Merkle, 2008).

The ACO system was originally designed to solve traveling salesman problem (TSP) (Blum, 2005), which has a single objective of minimizing the trip distance. The last property of digital ants mentioned above is the critical aspect of digital ants which lets researchers define complex rules and formulas for ants, to overcome multi-objective problems.

With the background given on ACO, it is possible to declare ACO as follows:

ACO is a probabilistic technique, which does approximate optimisation based on swarm intelligence (Blum & Merkle, 2008). ACO belongs to metaheuristic optimisation techniques, which solves the problem by populating the abstract model of the world, represented by a graph (Dorigo et al., 1996).

2.3.1 Swarm Intelligence

Swarm intelligence is the collective behaviour of self-organised, decentralised system. The system, either artificial or natural, comprises a number of agents, dwelling an environment known as the world (Blum & Merkle, 2008). The idea was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems (Beni & Wang, 1993).

In the domain of swarm intelligence, in addition to the number of agents and their placement, it is the individual behaviour of the agents, and their interactions that forms the collective swarm intelligence (Blum & Merkle, 2008).

The examples of swarm intelligence in nature are bird flocking, ant colony, fish schooling, animal herding and microbial intelligence (Blum & Merkle, 2008).

2.3.2 Metaheuristics

Metaheuristics refers to a high-level algorithmic framework, which is problem-independent and offers a set of guidelines and rules to develop a heuristics optimisation algorithm (Glover & Sörensen, 2015).

Metaheuristics often implement a sort of stochastic optimisation, which means the solution(s) found depend on a set of random variables (Bianchi, Dorigo, Gambardella, & Gutjahr, 2009).

2.3.3 ACO from Enclosure Perspective

ACO uses an abstract world, presented as graphs. The edges can be either one side or two sided, and the graph may be open or closed (Blum & Merkle, 2008).

Each vertex represents a room, and each edge represents connections between these rooms (Blum, 2005). An edge should have a length, where it can also have other properties such as area, capacity or any other static or dynamic property (Yi & Kumar, 2007).

2.3.4 ACO from Population Perspective

ACO uses populated graphs as environment. At the beginning of the process, the graph is populated by a number of agents (ants). The positioning of agents may be random or fixed. Each agent is assigned to a home node, and will seek the destination nodes during the procedure (Yuan & Wang, 2007).

2.3.5 ACO from Behaviour Perspective

ACO is an iterative algorithm. During each iteration, agents construct a path to the destination nodes (Yuan & Wang, 2007).

The behaviour of agents, in the case of ACO, is limited to selecting the next node, based on the possibility, calculated by a formula (Blum, 2005).

The basic deciding factor in the path-selection is the pheromone of the outgoing edges. Depending on the objectives of the ACO, there may be other deciding factors, such as edge capacity, edge load, and edge risk (Blum, 2005; Liu et al., 2016).

In ACO, during an iteration, the load of an edge may affect the probability of the edge getting chosen for other agents, so decision making of an agent may affect the future decision making of other agents, however, ACO does not suggest a collective behaviour for agents (Blum, 2005).

After each iteration, where all ants have constructed paths to exits, the total evacuation time and other objective functions will decide the efficiency of the current solution. If the solution is a dominant solution, the pheromones for edges which ants have traversed will be updated (Liu et al., 2016).

The new pheromone value changes the probability for each edge and in each iteration, and this way the efficient paths get marked after each iteration, and the solution will become more efficient.

2.3.6 Multi-Objective ACO

ACO system, originally, as it was proposed by Dorigo (1996), had one objective, and it was to find the shortest path to the exit. However, shortly after, multi-objective ACO (Multi-ACO) emerged, which had solution for problems with multiple deciding factor (Blum, 2005).

The implementation part of multi-ACO differs from ACO as follows:

- The probability for edges depend on factors from other objective factors as well. For example, if minimizing the congestion degree is an objective, the more an edge is loaded, less probable it becomes to be chosen.
- The efficiency of the current solution, in each iteration is calculated by a combination of all objective functions, not just the total traversed length, or total evacuation time.
- The pheromone updating function depends on the value of all objective functions, linearized and weighted.
- With multiple deciding factors for efficiency, known as objective function values, there is not just one best solution. There is a set of solutions which is called non-dominated solution set.

Multi-ACO in evacuation domain has been proposed for a variety of problems, featuring a wide range of objective functions. Examples of objectives are minimizing total evacuation time, minimizing total traversed path, minimizing total congestion degree, and minimizing total risk factor (Koo, Hong, & Kim, 2015; Yuan & Wang, 2007)

2.3.7 Multi-Criteria Optimisation

Multi-criteria Decision Analysis (MCDA) covers evaluation of a multi-criteria problem (Greco, Figueira, & Ehrgott, 2005), such as multi-criteria optimisation, for multi-ACO.

There are a variety of MCDA solutions. Based on ACO, there is a need for a solution capable of storing alternative solutions, as well as ranking them.

In MCDA, problem is captured by a matrix. The alternative solutions are the rows of the matrix, and the columns are the score of each objective function, calculated for each solution (Greco et al., 2005).

For each iteration in ACO simulation, a solution is generated. This solution is added to the matrix as a row. Two approaches are widely used in the literature. First approach involves a quick ranking method to purge the dominated solutions in the matrix, and the second approach uses a quantitative ranking method to sort the results based on the weight of the objective functions.

Pareto optimal solution compares each single score of objective function with alternative solutions. According to Deb (1999) (as cited in Fang, Zong, Li, Li, & Xiong, 2011), the MCDA can be stated as follows:

Min $f(x) = (f_1(x), ..., f_n(x))$

n is the number of objective functions, and $\mathbf{x} = (x_1, ..., x_m)$ is the vector of decision variables. $\mathbf{X}$ is the decision variable space. $f(x)$ is the vector of objectives.

It is possible to say $\mathbf{u} \in \mathbf{X}$ dominates another decision variable $\mathbf{v} \in \mathbf{X}$, if and only if (Fang et al., 2011):

$\forall i \in \{1, 2, ..., n\}, f_i(u) \leq f_i(v)$ **and** $\exists i \in \{1, 2, ..., n\}, f_i(u) < f_i(v)$

This means that a solution $\mathbf{x} \in \mathbf{X}$ is pareto optimal if there is no other decision vector that dominates **x**. The pareto-optimal solutions belong to a set, named non-dominated set. When a new solution is found, it will be checked against all of the solutions in non-dominated set (ND-Set) which is the matrix mentioned before, and if the new solution dominates any solution in the set, that solution is purged from the set and new solution is added to the set (Fang et al., 2011).

After a simulation, the ND-Set contains multiple solutions, for each solution at least one objective function is pareto optimal. The second quantitative method used in this paper is Weighted product model (WPM) which is capable of comparing two solutions, based on the weight of the objective functions. WPM method is defined as follows (Triantaphyllou, 2000):

$$P_{(\frac{A_K}{A_L})} = \prod_{i=1}^{n} \left(\frac{a_{K_i}}{a_{L_i}}\right)^{\omega_i}$$

This formula compares two alternative solutions $\mathbf{A_k}$ and $\mathbf{A_L}$. The $\mathbf{a_{Ki}}$ and $\mathbf{a_{Kj}}$ are the column values, that is the objective function values for $\mathbf{A_k}$ and $\mathbf{A_L}$. $\boldsymbol{\omega_i}$ is the weight of the respective objective functions. To understand the formula the following table may be helpful:

	C_1	C_1	C_1	WPM_{score}
Weights (ω)	0.55	0.35	0.1	
A_1	22	75	233	11.73
A_1	18	120	155	11.90

Table 2.1. An example of WPM.

Rows A_1 and A_2 are the solutions in Table 2.1, and C columns are criteria. The WPM formula can be rearranged as bellow:

$$P_{(\frac{A_K}{A_L})} = \frac{\prod_{i=1}^{n} {a_{K_i}}^{\omega_i}}{\prod_{i=1}^{n} {a_{L_i}}^{\omega_i}} = \frac{P_{A_K}}{P_{A_L}}$$

So, for each solution, a WPM score can be generated and then compared with the rest of the solutions. This method is more efficient for a solution matrix with lots of rows. WPM for each row (P_{A_K}) can be calculated as follows:

$$P_{A_K} = WPM_{score} = C_1^{\omega_1} \times C_2^{\omega_2} \times \ldots \times C_n^{\omega_n}$$

The objective functions and their weights are discussed in chapter 3.

2.3.8 Crowd Dynamics

Level of Service (LOS) is a concept within the domain of pedestrian flow dynamics. Pedestrian flow dynamics is a well-researched area that tries to understand the dynamics of crowd flow, and model some critical aspects of the crowd flow (Vanumu, Ramachandra Rao, & Tiwari, 2017).

While the pedestrian flow is a wide domain, and lot of variables and formulas have been suggested to model different aspects of the pedestrian flow (Still, 2000; Vanumu et al., 2017). The needed parts for this research is limited to the flow speed, relative to density, and a standard for density measurements. This is because ACO is an agent based model, however, since the world is represented as graph, the quality of space is not a deciding factor for ACO.

As for a measurement for density, Level of Service (LOS) is used. Level of service is a quantitative measure for quality of traffic (Still, 2000). LOS is used to rank the quality of sidewalks, walkways, crosswalks, stairways, highways, and traffic flow mediums like these (Still, 2000). As for the measurement for quality of space per pedestrian, the research done by Fruin (1971) proposes Fruin LOS, which ranks space per pedestrian with alphabets, from A to F. Figure 2.2 shows an overview of what LOS looks like, and Table 2.2 presents the description of Fruin LOS.

LOS	Ped.Volume (f) min pr/ft pr/m	Average Area (a) ft2/pr m2/pr	Description
A	7 or less 23 or less	35 or more 3.3 or more	Threshold of free flow. convenient passing, conflicts avoidable.
B	7-10 23-33	25-35 2.3-3.3	Minor conflicts, passing and speed restrictions
C	10-15 33-49	15-25 1.4-2.3	Crowded but fluid movement, passing restricted, cross and reverse flows difficult.
D	15-20 49-66	10-15 0.9-1.4	Significant conflicts, passing and speed restrictions, intermittent shuffling.
E	20-25 66-82	5-10 0.5-0.9	Shuffling wall: reverse, passing and cross flows very difficult; intermittent stopping.
F	Flow variable up to maximum	5 or less 0.5 or less	Critical density, flow sporadic, frequent stops, contacts with others.

Table 2.2. Fruin LOS description for Walkway (Fruin, 1992).

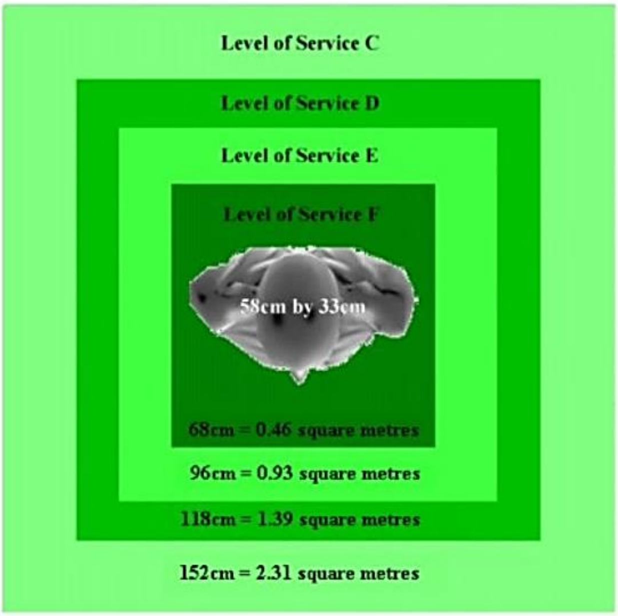


Figure 2.2. Fruin (1971) LOS area per person as cited by Still (2000).

To address pedestrian speed, relative to crowd density, there are a variety of methods in the literature. Pedestrian speed is known to be relative to many factors, including the country of origin, age groups or pedestrians, chaos factor, walkway type and material, and density of crowd (Alhajyaseen, Nakamura, & Asano, 2011). Figure 2.3 visualizes the fundamental characteristics of human flow, with different methods:

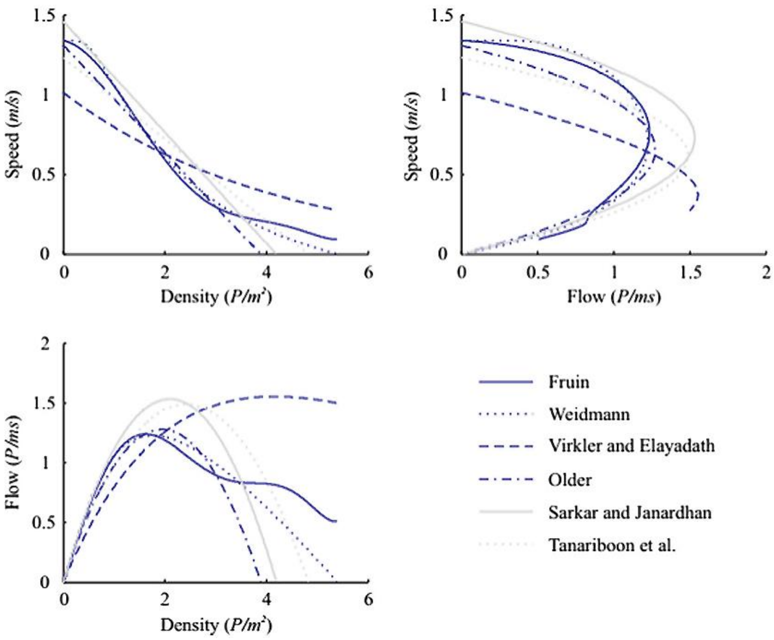


Figure 2.3. Pedestrian flow fundamental diagrams (Daamen et al., 2005)

As it is obvious, the Fruin pedestrian speed is not linear in the figure, however, based on Fruin (1971) article, the speed can be simplified as a linear formula, relative to density. This is the formula chosen in this article, described as follows (As cited by Alhajyaseen et al., 2011):

$$v_p = 1.43 - 0.35 \times P_A \ , P_A\,(\text{person/m}^2) = \text{Passway}_{\text{load}}/\text{Passway}_{\text{area}}$$

As it was mentioned earlier, the pedestrian speed varies from country to country. It makes sense to ask why the Fruin formula is chosen over those of done in United Kingdom. The reason behind is that based on the Fruin formula, when density passes 4.08 person/m^2, the speed reaches zero, which means a complete clogging. This is in sync with Fruin level of service.

Besides, according to the literature, average speed depends on many factors, and it is best represented as a range. The base speed in the literature varies from 1.0 to 1.5 m/sec and the Fruin base speed of 1.43 is in the range.

2.3.9 Bottlenecks

Many factors play role in forming a bottleneck. The most basic factor is the width of the passway as its entrance, and passway area that determines its capacity (Tanimoto, Hagishima, & Tanaka, 2010). Figure 2.4 shows this effect. The exit #2 is 40cm wider and as it is seen, the population favours this exit.

The majority of the agent-based evacuation methods implement a method to address this issue for agent decision making (Wagoum et al., 2017).

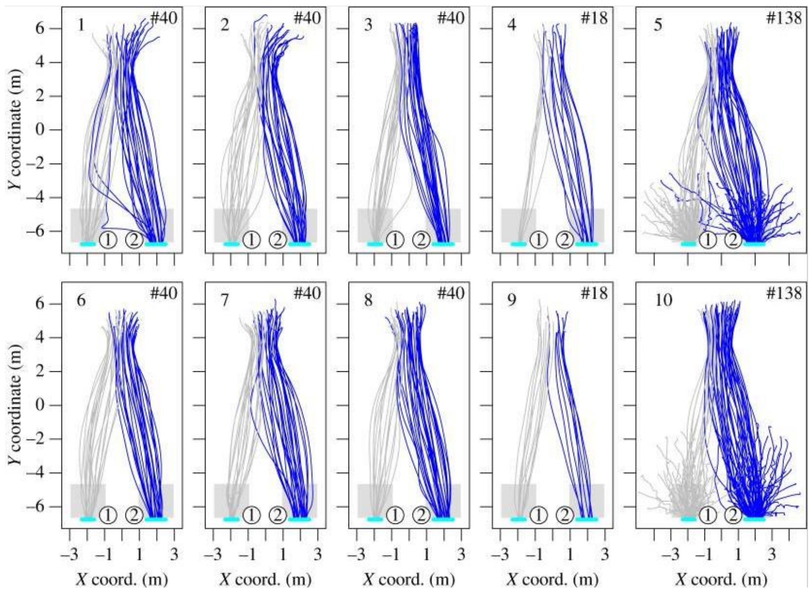


Figure 2.4. The visualization of path-selection relative to exit width, and different population. e1=0.7m, e2=1.1m (Wagoum et al., 2017)

Bottleneck analysis is an active research area; however, as Multi-ACO perceives the world as a graph, the passway width and area are the only possible properties that can be attached to the graph components, so the rest of the bottleneck science falls out of the scope of the current research.

2.3.10 Shortest Loopless Path Method in a Network

To validate the results, a baseline method is needed. Shortest path method is a method that is based on the method suggested by Dijkstra's shortest path (1959).

The shortest path method has lots of variants. The most common one uses a node as source and finds the shortest path from this source to other nodes in the graph, which constructs a shortest path tree.

Dijkstra algorithm is a greedy algorithm, which means for each node, it iteratively investigates outgoing connections to make optimal solution for shortest path. Because of its greedy nature, when the graph is complicated, this algorithm sees a steep drop in performance.

2.3.11 Multi-ACO Implementation

The general implementation steps of a multi-ACO implementation are as follows (Fang et al., 2011; Zong, Xiong, Fang, & Li, 2010; Yuan & Wang, 2007):

1. instantiation: In this step, graph is instantiated; the fundamental parameters are set; all the edges get the default pheromone value; the agents are instantiated and assigned to home nodes; and the non-dominated solution set (ND-set) is instantiated.
2. path construction: for each agent which is not moving, the next node is chosen based on a probability formula. The formula depends on pheromones as well as any heuristic function that may affect the probability of the next nodes, for example, a reverse correlation with edge load, or edge risk may define the heuristics function.
3. time is incremented and step 2 is called, until all agents find the destination (exit) nodes.
4. ND-set determination. The objective functions for each path is calculated, and summed up to get the final score of each objective function for this iteration. The current solution, which is a collection of paths, is tested based on the

mentioned score, against other solutions in the ND-set. If the solution is non-dominated, it will be added to the ND-set and any solution from ND-set that is dominated by current solution will be purged from the ND-set.

5. Pheromone updating: Pheromones evaporate by each iteration, and if agents pass from an edge, an amount of pheromone is deposited on the edge. The new pheromone value may depend on the objective function values, or it may not, depending on the ACO design.
6. The ants get back to the home nodes and step 2 is called, until the maximum number of iteration is reached. In this case the algorithm is terminated.

The technical specification of the multi-ACO is described in chapter 3: design, and hence it is not present in this chapter.

The first part of this chapter tried to lay the details of multi-ACO. The next part will give an overview of the literature in the domain.

2.4 Related Work

Human-flow dynamics has been a field of research form the mid-20th century. Human flow characteristics was dominant area of research through 50s to 80s. Shortly after, with the growth of crowd dynamics, the crowd movement simulation methods were born to aid forecasting of the complicated human flow patterns (Still, 2000).

Fruin investigated pedestrian dynamics and released his highly cited book “Pedestrian planning and design” (Fruin, j, 1971). His research has been accepted as standard in many subsequent building construction regulations and hence has been reflected in many researches afterward. He defined a measure for safety and comfort, which relates to crowd speed and density, that is Level of Service.

Many researchers followed Fruin and measured crowd dynamics in different countries, however, the acceptance and wide covered ground of Fruin research makes it a feasible solution for a research that involves multiple crowd dynamics criteria (Still, 2000).

The field of path-finding and crowd-simulation also goes back to mid-20th century. Dijkstra, a computer programmer and physics professor, proposed a solution to find the shortest path between nodes in a graph, proposed in his article “A note on two problems in connexion with graphs” (Dijkstra, 1959). His idea was utilized in many branches of computing, including path-finding for real life scenarios. His method was based on an iterative path-evaluation method starting from a node and building a distance tree for each node, updating from different branches to the minimum value. This gave birth to the graph based path-finding.

During the last decades of the 20th century, a variety of different computer-aided crowd simulation methods emerged. Gwynne et al. (1999) research identifies 22 evacuation models and categorizes them in three different major branches: Simulation, optimisation, and risk management. This research also defines the elements of the optimal design for evacuation.

Ant-Colony Optimisation (ACO) is an optimisation method proposed by Dorigo, in his PhD thesis (1992). This method is based on swarm-intelligence seen in real ants. Dorigo denotes that ACO can be applied to a broad range of problems that can be presented as path-finding in a graph.

Shortly after, Dorigo proposed the first ACO algorithm in his paper “Ant System: Optimization by a colony of cooperating agents” (1996). His ant-system algorithm was a solution to traveling salesman problem, as a proof of concept for ACO. With the effectiveness of the ACO known, the algorithm became a major research area quickly.

The ACO-metaheuristics, which is a form of ACO, capable of utilizing rule-based method for path-finding, was first proposed in 1999 (Dorigo, Caro, & Gambardella, 1999). The research uses a parametrized probabilistic model for pathfinding, and also describes a variety of ACO applications to combinatorial optimization and routing in communications networks.

The multi-objective ACO was first proposed in 2003-04. The article “Solving Multi-criteria Optimization Problems with Population-Based ACO” (Guntsch & Middendorf, 2003), proposed the multi-objective ACO for the first time. This article proposed methods which are the fundamental part of the todays Multi-ACO algorithms. The algorithm used a set to store dominant solutions found during iterations. It used a pheromone matrix for each optimisation criterion, and the decision making of ants were based on a weighted method that rasterized the pheromones from different matrices. This method becomes inefficient quickly as the population grows, because updating and combining the matrices is a CPU intensive task.

The next milestone was set by the research “On the Design of ACO for the Biobjective Quadratic Assignment Problem” (López-Ibáñez, Paquete, & Stützle, 2004), that defined objective function vectors and pheromone updating strategies. The experiment showed that the local search strategy and the correlation between objective functions play an essential role in the efficiency of the algorithm.

From 2006 on, the majority of the ACO researches in the field of evacuation science were trying to define innovative objectives for the ACO to address varying needs of different evacuation scenarios.

Yuan and Wang (2007) proposed a Multi-ACO solution that used both egress time and total traversed time to minimize the complexity of routes. They combined these two objectives with another objective function to minimize risk of the path, simply by taking into account the distance from the danger zone. The research evades testing the solution on huge networks and predicts performance hit for the large networks.

Fang et al. (2011) describe the inefficiency of Multi-ACO for high population numbers, over 10000, and big enclosures. They propose a hierarchical routing system in which evacuees navigate to a middle set of nodes, and not directly to the exit. They also refuse to rasterize the results of objective functions, and only use Pareto-optimal method to find the best solutions. This method may cause confusion as to the importance of the objective function are not mentioned and getting quantitative analysis is way harder.

The Multi-ACO during the last years has not seen fundamental changes. There are quite a few innovative methods proposed in these years, one of them is the research by Zhang and colleges about binary pheromone updating strategies. In their paper “Quantum ant colony algorithm-based emergency evacuation path choice algorithm”, they propose a solution to facilitate optimisation by updating the pheromones by a growing weight, gradually during iterations. They claim that this method prevents premature permutations and help improving the rate of optimisation during simulations.

In the recent years, in the domain of the evacuation path-finding algorithms, other algorithms have emerged, that some of them are compared to the ant-colony and the researchers claim a better bottleneck handling for evacuation. Evacuation routing using fish swarm algorithm is one of the eye-catching new algorithms in the field. This method which is based on swarm-intelligence like Multi-ACO, works based on intelligent divisions in the population, and not just the agent selection. The researchers claim that it is both quicker and more efficient than Multi-ACO, in terms of congestion handling, however, the egress time is left out of the equation, which makes it hard to compare them side by side.

2.5 Technologies

This section gives a review of the tools used to develop and test the solution.

2.5.1 C#

The programming language used to develop the Multi-ACO is C#. it is an object-oriented programming language developed by Microsoft that tries to combine the computing power of C++ with the programming ease of Visual Basic (Rouse, 2007). C# is based on C++ and contains features similar to those of Java.

The researcher was familiar with a variety of different programming languages; however, C# was chosen because of the following advantages:

- C# is native to the windows, unlike Java and Python it directly translates to object code, and runs without a medium. The performance is leveraged in this situation.
- C# is a EcmaScript compatible language, which is readable to anyone who is familiar with JavaScript and Java. It is also a simpler language than Java and C++; it does garbage collection automatically, manages instantiating and passing the references and values automatically, and has a lot of extra libraries and resources available online.
- C# has a powerful debugging tool inside the Visual Studio, the default editor.
- C# originally worked based on Microsoft .Net Framework, exclusive to windows. Nowadays the .Net Core library has made C# portable to any operating system, so if there is a need to develop the solution on a different machine and operating system in the future, it is possible with this language.

The risks associated with C# are as follows:

- Performance improves on native languages, however, this makes the evaluation of performance harder, as the system may have some services running in the background, affecting the performance. So, the test conditions should be monitored carefully.

- The part about passing instances by value or reference turns out to be a troublesome. When the instances of classes are automatically passed as reference, any changes to them is reflected in the main instance of the class. For example, any operation on the nodes of the world graph, affects the future parts of the application.

2.5.2 Tableau

The visualization software used to present the visuals of the research is Tableau. It is an interactive visualization tool that enables users to produce customized set of visualization in its dashboard with ease (Ahmed, 2017). The ease of using the interface along the reasonable variety of visualization styles made this a good choice for the research.

3 DESIGN

This chapter describes the specifications of the experiment, including the Multi-ACO algorithm technical specification, the algorithm from enclosure, and population perspective. The data gathering and comparison strategies are also described. In the second part of the chapter a small review of software design is presented.

3.1 *The Experiment Design*

This section documents the details of the design of the experiment, which includes the proposed Multi-ACO mathematical description, and Multi-ACO from enclosure and population perspective. The baseline method that is the shortest path is also described in this section,

3.1.1 Multi-ACO Algorithm Specifications

The multi-ACO is a multi-step algorithm, as described in chapter two. Here, the formula used in each step and their purpose, alongside the parameters are listed.
Variables and parameters are as follows:

- The environment is defined as a bi-directional graph **G(V, A)** where **N={1, 2, …, n}** is the set of nodes, and $\mathbf{V \subseteq N \times N}$ is a set of bi-directional arcs, defined by two nodes.
- e_{ij} is the edge that connects node **i** and **j**
- l_{ij} is the length of a_{ij} and $area_{ij}$ is its area.
- C_{ij} is the maximum capacity of the a_{ij}, based on Fruin velocity formula described in chapter two, the maximum capacity is 4 p/m^2, hence: $C_{ij} = \lfloor area_{ij} \times 4 \rfloor$
- M is the total number of agents, N is the total number of edges.
- t_{ij}^{k} is the time for agent k, to pass a_{ij}
- N_{ij} is the current load of a_{ij}, or the number of agents on that arc.
- ω_{ij} is the congestion degree of a_{ij}, x_{ij} is the number of agents on edge ij.
- S_k is the path agent k traversed, a path is a set of arcs in order.

- τ_0 is the initial pheromone value for edges, and τ_{ij} is the current pheromone of edge ij.
- dT is the discrete time frame value, by which time increases in a simulation.
- v_p is the speed of an agent, and $next_t$ is the next node arrival time.

The formulation of multi-ACO is as follows:
The objective functions:

$$min\ f_1 = max\ t_{Sk} \quad \text{(f1}$$

$$min\ f_2 = \sum_{t=0}^{t_{max}} \left(dT \times \sum_{e_{ij} \in G} \omega_{ij}\right) \quad \text{(f2}$$

$$min\ f_2 = \sum_{K_0}^{K_m} \sum_{e_0}^{e_{ij} \in S_k} t_{ij}^k \quad \text{(f3}$$

The first objective function (f1) tries to minimize evacuation time, which is equal to maximum egress time by any agent, and third one (f3) tries to minimize total traversed edge time by all agents.

Based on the Fruin LOS, congestion degree of an edge (ω_{ij}) is defined relative to Fruin LOS index. Fruin LOS index is calculated based on agent per cubic meter (P_A), so load for each edge, and in each incremental time frame, is calculated as follows:

- $\omega_{ij}(t_0)=0$, ω_{ij} is zero at the beginning
- $P_A > 3.25$, **level A:** $\omega_{ij} = (\omega_{ij} + 0)dT = (\omega_{ij})dT$
- $2.32 < P_A \leq 3.25$, **level B:** $\omega_{ij} = (\omega_{ij} + 1)dT$
- $1.39 < P_A \leq 2.32$, **level C:** $\omega_{ij} = (\omega_{ij} + 2)dT$
- $0.93 < P_A \leq 1.39$, **level D:** $\omega_{ij} = (\omega_{ij} + 3)dT$
- $0.46 < P_A \leq 0.93$, **level E:** $\omega_{ij} = (\omega_{ij} + 4)dT$
- $P_A \leq 0.46$, **level F:** $\omega_{ij} = (\omega_{ij} + 5)dT$

P_A itself is calculated as the current population of the edge on its area, as follows:

$$P_A = \frac{x_{ij}}{area_{ij}} \quad \text{(f4}$$

Total congestion degree for a time frame of dT, ω_{dT} for an edge, is the sum of all congestion degrees for all edges in a solution, multiplied by the discrete time amout, dT. For example, if time increase by 0.5 second, the congestion degree is multiplied by half. The sum of all congestion degrees for all time frames, is the ω_{total} which is the same as the second objective function.

Each iteration is an iterative process of path-construction by agents. Time starts from 0 and after all agents decide where they want to go, time is incremented by a discrete amount and the process goes on.

Agents that are already moving do not need to choose next node. When agents reach their next node, they need to choose where to go from that node. The traversed nodes for an agent are stored in a list. The agent cannot move to any node from this list.

Agent chooses the outgoing node as the next node based on probabilities. The probabilities of the next edges add up to one, and for an edge, is calculated with the following formula (f5).

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{n=1}^{N}\left(\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}\right)} & , w_{ij} < C_{ij}, j \in nontraveresed\ edges \\ 0 & , w_{ij} = C_{ij} \end{cases} \qquad \text{(f5}$$

Formula 5 denotes that if the capacity is reached, the probability of an edge being selected is set to zero.

η_{ij} is the heuristics function that alters the possibility based on the defined objectives. The heuristics function (f6) is defined bellow:

$$\eta_{ij} = \frac{C_{ij}}{\omega_{ij} \times e^{\frac{-2}{b_{ij}}}} \qquad \text{(f6}$$

The heuristics function has linear relation with capacity of the edge, and negative linear relation with the load of the edge. The more loaded the edge is, less possible to be chosen. τ_{ij} is the current pheromone of the edge. Constants α and β are the importance of functions, which are powers of the pheromone and heuristics function.

The last part of the denominator is an exponential function which relates to the width of the edge exit (b_{ij}). The pivot point of the $exp(-2/b_{ij})$ is when $b_{ij} = 2$. This means that for the lengths more than two, the possibility of being chosen spikes rapidly and for less than 2 it is reduced exponentially.

The formula (f6) addresses the variable throughput and width of the exit issue, defined in chapter two, congestion section.

After all the agents choose their next node, the next node arrival time is calculated based on the Fruin (1971) pedestrian flow speed, described in chapter 2. The formula is as follows:

$$\boldsymbol{v_p = 1.43 - 0.35 \times P_A} \quad \textbf{, } \boldsymbol{P_A} \textbf{ agent/m}^2 \textbf{ as described in (f4)} \tag{f7}$$

$$\boldsymbol{next_t = \frac{l_{ij}}{v_p}} \tag{f8}$$

When there is no agent left without a destination node, the time is incremented, and the statuses of the agents are checked again, if they reach the next node, the process if repeated again, until all the agents construct their path to exit. The egress time will be the maximum time taken by any agent to get to any exit on the map. This egress time is defined by the first objective function (f1).

This is the process that happens in one iteration. After an iteration, the objective functions are calculated, which are the egress time (f1), total congestion degree for all edges (f2), and total traversed time (f3).

Upon finishing the calculation of objective functions, the solution will be checked against the current ND-set. If it is non-dominated, the pheromones will be updated.

The pheromone updating process is described by formula bellow:

$$\boldsymbol{\tau_{ij}(T+1) = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}} \tag{f9}$$

The $\boldsymbol{(1\text{-}\rho)\tau_{ij}}$ covers the pheromone evaporation process. Each iteration, the current pheromones on an edge is reduced, and if agents passed the edge, new pheromone is deposited on the edge. The new pheromone is calculated as follows:

$$\boldsymbol{\Delta\tau_{ij} = \sum_{n=0}^{N} \Delta t_{ij}^{n}} \tag{f10}$$

$$\boldsymbol{\Delta\tau_{ij}^{n}} = \begin{cases} \dfrac{Q}{t_{S_k} \times \sum_{k=1}^{M} \omega_{ij}} & \boldsymbol{, if\ ant\ k\ passed\ ij\ in\ S_k} \\ \boldsymbol{0} & \boldsymbol{, Otherwise} \end{cases} \tag{f11}$$

The pheromone evaporation is controlled by $\boldsymbol{\rho}$ (rho). The new pheromone function (f10) is a sum of new pheromone values for each edge. This new pheromone value (f11) is defined as a constant **Q** as numerator, on the path egress time multiplied by the total path congestion degree. The amount of **Q** depends on the initial pheromone value of the edges, that is $\boldsymbol{\tau_0}$.

The traditional way of calculating the total path congestion degree is described by the following formula (f12):

$$\omega_{ij} = \frac{x_{ij}}{c_{ij}} \quad \text{(f12}$$

This research will try to substitute this with the new edge load formula as described before in LOS calculation section.

It is necessary to mention that the fact that pheromone updating function has negative relation with the two of the objective functions is not a coincident. This is the practice in many multi-ACO implementations (Fang et al., 2011; Liu et al., 2016; Yi & Kumar, 2007). The reason behind this is that when the objective function value is high, the solution is less-optimal and it is natural for agents to leave less pheromone. When the solution becomes more optimal during iterations, the amount of the pheromone deposits increases.

3.1.2 Design of the Simulation

Since this research aims at comparing the heuristic function in path-finding, it needs to be tested against a variety of different conditions. These conditions must ensure that:

- The constants in the Multi-ACO are optimised and do not affect the efficiency.
- The conditions should vary randomly to minimize the effect of randomly biased results.
- The conditions set for each instance of simulation, which is randomised for next instance, should be used to generate data for each algorithm.

The results from each variation is compared with baseline, which is the shortest path algorithm, and the traditional variation of multi-ACO that uses a simple, commonly found, congestion degree objective function, defined below (f13):

$$\min f_2 = \omega_{total} = \sum_{t=0}^{t_{max}} (dT \times \sum_{e_{ij} \in G} \frac{x_{ij}}{C_{ij}}) \quad \text{(f13}$$

During each iteration of a single simulation, in which time increases by a discrete amount, the congestion degree for each edge is calculated with the congestion degree formula, multiplied by the discrete time value. The final total congestion value is the sum of all congestion degrees, over simulation time.

The implemented Multi-ACO has two objective functions, one calculates total egress time and the other calculates total congestion degree.

Aside from the effect of the heuristics function, there are two variables that affect the total time, that need to be either randomized, or tested in multiple design groups. There are listed here:

- The physical form and capacity of the enclosure, that is the environment which is represented as a graph.
- The population of agents, and their positioning at the beginning.

The environment cannot be randomized, as it is represented by a graph, which is derived from the physical layout of the building, hence there is a need to test the algorithm against a variety of enclosures with different capacities. Because of the scale of the research, two different enclosures are designed for the test. The enclosures are described in the next section.

As for the population standpoint, which refers to the number and positioning of the agents. For the population, three different groups are defined, based on the real-life conditions, low-density population, medium-density population and high-density population environments. The details are described in the population section.

For the positioning of the agents, the feasible solution is to distribute population randomly on the home nodes. Hence the results must represent a battery of simulations. This is true for all the comparisons.

To fulfil the first objective, which is proving the effectiveness of heuristics function, a battery of simulations is done. For each simulation, it is done once with multi-ACO and once with shortest path. And for the next simulation, the population is relocated randomly. Then the battery of simulations is run for other population groups as well, meaning each battery gives results of fixed number of agents, with random positioning.

The first objective function also depends on the effectiveness of pheromone updating function, which itself depends on a set of parameters. These parameters will affect the rate of improvement of path-finding in each iteration of a simulation.

Since there are a variety of practices in the literature, and no dominant best practice exists, an experiment with different set of parameters also should be carried out to extract the parameters for the formula.

To measure the improvements, all the paths constructed by agents, in each iteration of a simulation, will be compared against the shortest path from the home node to the nearest exit, and the differences in each iteration will mark the improvement. This method writes off the second objective function which is the congestion degree function. Since the fundamental purpose of an evacuation simulation is to minimize the evacuation time, and the optimal answer to the congestion degree cannot be extracted, this solution is the most feasible one.

3.1.3 Enclosure and Population Perspective

The specifications suggest the need for at least two enclosure designs. The architectural plans for the first building block is shown in figure 3.1.

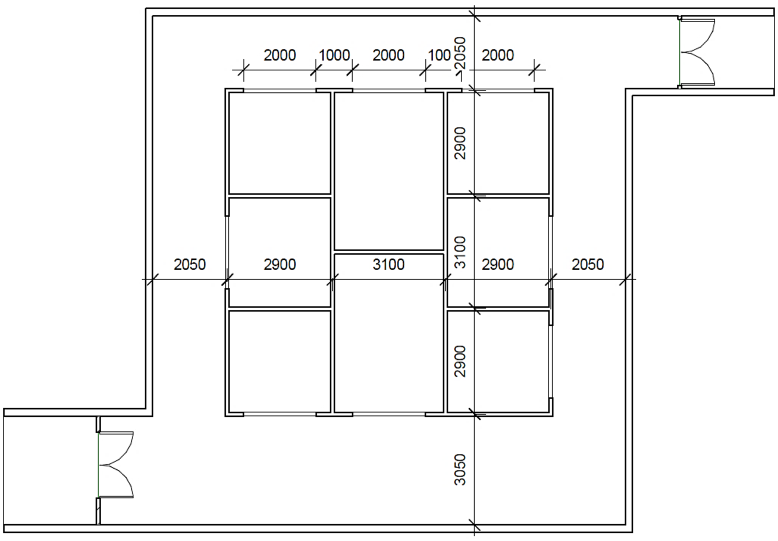


Figure 3.1. Architectural floor plan for the first block. Scale 1:200.

The building is a simple block of rooms in the centre, which can be populated. The design is generic and can be assigned many roles. An isometric view of the block is shown in Figure 3.2.

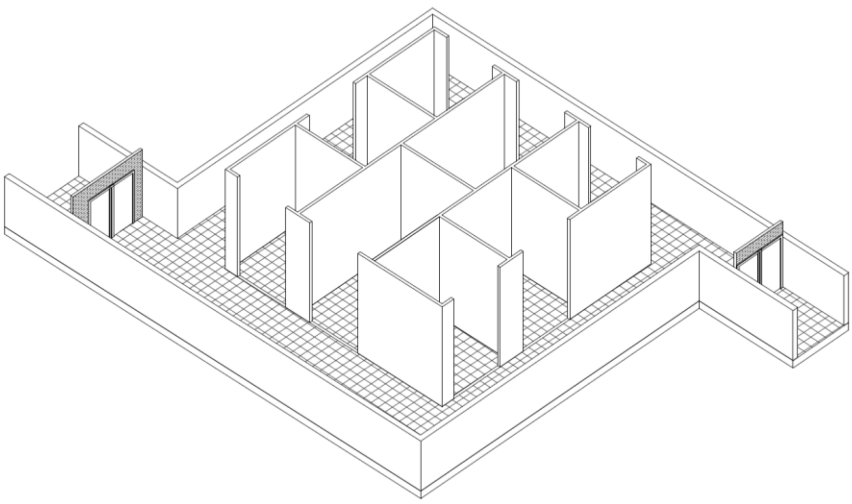


Figure 3.2. An isometric perspective of first block’s architectural layout.

The different roles mentioned for the building blocks is for real-life scenarios in which the block can be populated with different number of occupants, to address the need for different scenarios for congestion degrees. Three roles were designed for the experiment which are listed here:

1. Low-density residential service rooms. The total number of residents are 8 for small enclosure and 24 for big enclosure, each room may have zero to 2 residents. The occupants will be scattered randomly for each instance of the simulation. In this scenario, there is no congestion, because the capacity of the edges, which are the connections between rooms and intersections in the main hallways is way higher than being overload with 8 agents. This scenario gives the opportunity of comparing the total egress time of Multi-ACO with the shortest path algorithm, without comparing the congestions. This is important because the shortest path algorithm does not address congestions, which is assumed to have an impact on the egress time of the shortest path algorithm.
2. Medium-density office block during working days. Total number of residents will be 30 for small and 90 for big enclosure which makes 3.75 occupants for each room. Each room may have between 2 to 6 residents. This scenario will fill the gap between a low-loaded first scenario and a highly-loaded third scenario. According to the expectation, depending on the way occupants are scattered, the congestions may appear near the exit nodes, specifically in the intersections.
3. High-density exhibition centre. In this scenario, there are 60 occupants for the small enclosure and 180 occupants for big enclosure, between 6 to 10 occupants for each room. The congestion degree is expected to be high, especially in the narrower northern walkways. This phase of the experiment will shed light on the efficiency of the walkway width, which is present in the agent node selection probability formula. Also, the efficiency of the Multi-ACO compared to the baseline shortest path will be bold here.

The breakdown of the space to build the world graph is shown in Figure 3.3.

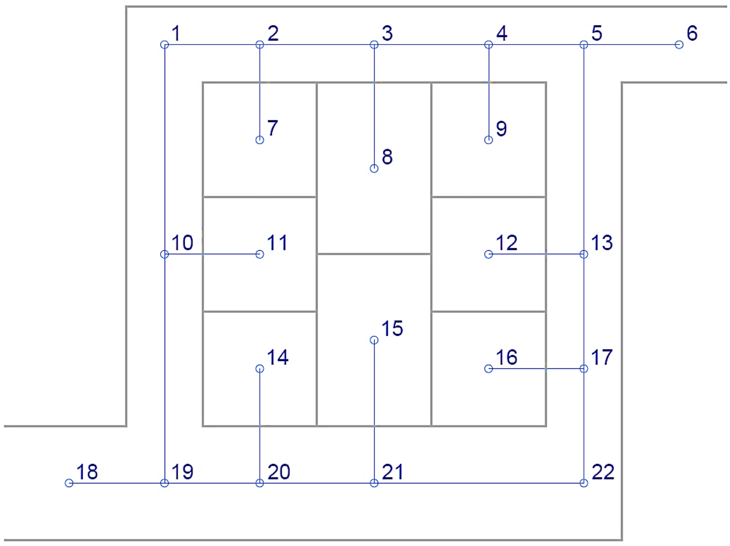


Figure 3.3. The graph of first building block, derived from architectural layout.

The breakdown of rooms and their connections are visible in Figure 3.4.

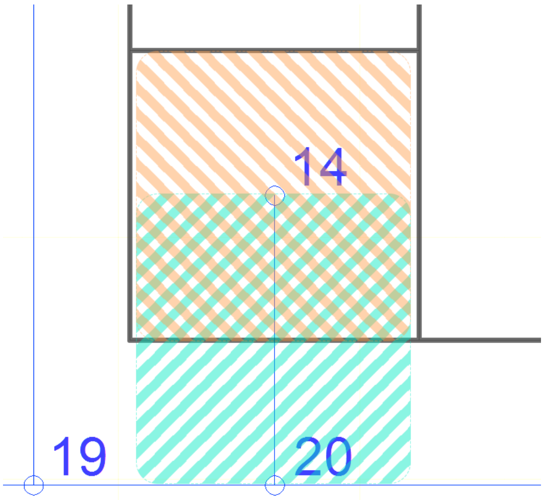


Figure 3.4. The breakdown of rooms and connections. Red are is the room and the blue are is the connection between the room, and intersection in the walkway.

From Figure 3.4. it is logical to ask about the fairness of how the node 14 represents the whole room, as it is located in the centre of the room. In defence of this design, it is necessary to note that first, the occupants are logically scattered across the room, so if it takes less time for those near the exit, the randomness of scattering means that there may be other occupants that need to cover longer distance to reach the next node. The same logic goes for the edge that connects node 19 and 20. The occupants that rush out may be in the half near or far from the 19.

The other aspect that makes this design logical is that the Multi-ACO, as a network optimisation method, considers the occupant movements optimal, and hence the disruptions caused by collisions in the intersections is not a part of agent behaviour in this method. So, it is possible to look at occupants like scattered moving chess pieces.

The second block needs to be more complicated to address more real-life egress scenario. The problem with the first building block is that the rooms are symmetrically distributed from the exits, so the shortest path and Multi-ACO results for the low and medium density scenario may not vary much. The second block is designed with asymmetric exits and more complex layout. Isometric and world graph view of the second model is shown in Figure 3.5 and 3.6.

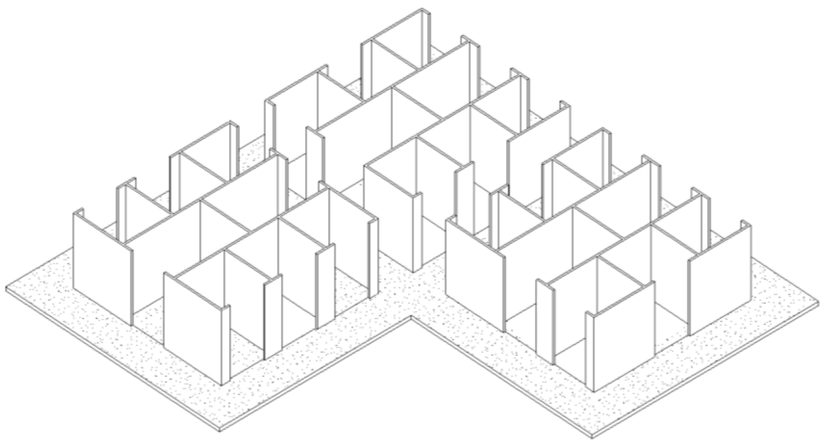


Figure 3.5. Isometric view of the second building block.

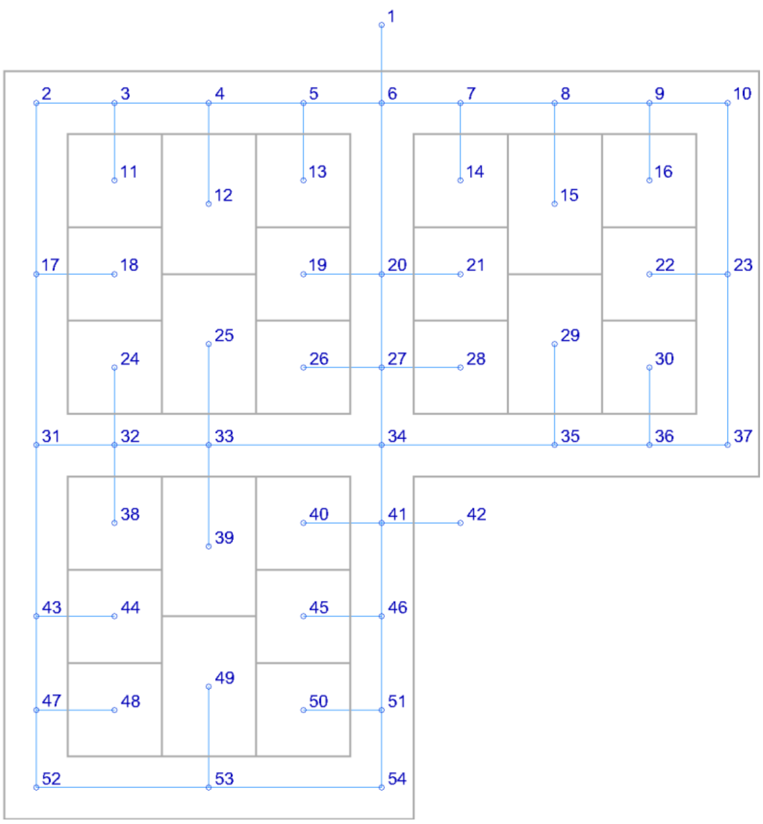


Figure 3.6. The world graph for second building block. Nodes 1 and 42 are the exits.

3.1.4 Shortest Path Algorithm

The shortest path algorithm which is the baseline, is a method that constructs a tree of nodes with their shortest distances from the original node.

This method is based on the Dijkstra’s algorithm, the variation for finding the list of shortest loopless paths in an open graph.

The algorithm is quite simple in its nature. The steps to produce the graph is show in Figure 3.7.

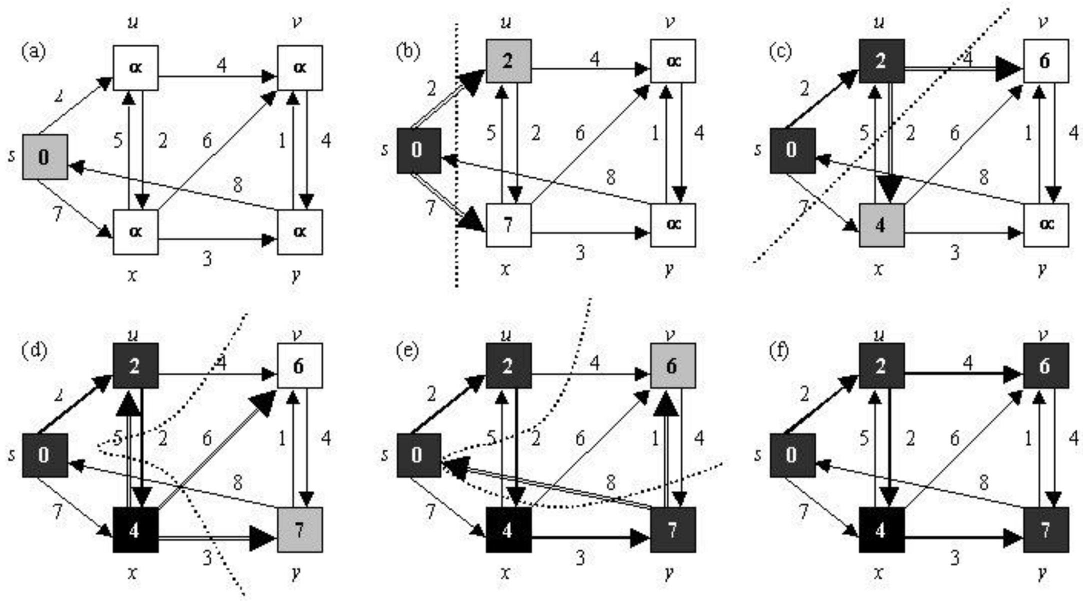


Figure 3.7. The Dijkstra’s algorithm. steps from a to f (Smith College, 2015).

The Dijkstra’s algorithm process shown in the figure above is described in the following steps:

1. First node distance is marked zero and the others are marked infinity.
2. The outgoing edges have lengths, the direct nodes from the first node are marked as the distance of their edges, as the distance from the home node.
3. For each secondary node, the process from step 1 is repeated, forming new branches. The branches make a graph of nodes that resembles the world graph, however, the process shown in Figure 3.7 is a visual one on paper, so it does not show the built graph. The new nodes get the distance of current node plus the distance of their edge.

4. If any of the outgoing edges already have a distance, if the new distance is less than what it already has, the new distance is the shortest distance from the home node. This is visible in the Figure 3.7, for node ‘x’, from step b to c.
5. The process is repeated for different branches until they reach the home node, or a dead-end is reached. For the fact that the process is loopless, a list of visited nodes for each branch is stored, so each branch cannot visit a node twice, however, different branches may reach a node multiple time.

The Dijkstra’s algorithm produces a list of shortest paths for each node. The problem here is that this research needs the list of the shortest paths to each node as well. So, the list of the shortest paths will be stored in each node, and updated alongside the shortest path value, during the branching process.

The shortest path algorithm in this research needs another feature to address multiple exits on the world map graph. So, the process of shortest paths is repeated for each exit, and this results in two instance of shortest path data, for each exit.

When the shortest path data is constructed for each exit, the process of simulation may begin. The shortest path simulation process used in this paper is described in the following steps:

1. The world graph is constructed and the rooms are marked.
2. The shortest path data is calculated for each exit, according to the Dijkstra’s algorithm described before.
3. The agents are placed on the map, assigned to home nodes, which are rooms.
4. The process of simulation is quite like ACO, there is a timeline starting from zero. Agents select the nearest exit based on the shortest path to any exit. If the path is the same distance, they choose either exits randomly.
5. The agents move to next nodes, which means they will populate edges. Then the time will increase by discrete intervals. For each interval, the load for each edge is calculated the same way it is calculated for the Multi-ACO. In the case the edge capacity is reached, the agents will not move, until when the edge capacity is less than maximum capacity.
6. The process from step 5 is repeated and time will increase until all agents are evacuated.
7. The results are gathered and stored as the final solution for the current configuration of occupant population and positioning.

Unlike Multi-ACO, the shortest path algorithm is run one time to get the final results. The occasion in which the shortest path is run is described in the next section.

3.1.5 Data Gathering and Comparison Strategies

This research gathers data from the self-managed simulations. The process of simulation is described for both Multi-ACO and shortest path algorithm. This section tries to point out where, and when the data is logged, also the type gathered data needs to be known beforehand.

During a single Multi-ACO simulation, each iteration generates a collective set of paths and paths are ranked by objective functions. At the end, there remains a set of ND-Set of solutions with their objective function values. This set of solutions make the final solution for the simulation. These solutions and their values are logged as the final solution, inside a matrix with a solution as a row, and objective functions as columns.

To get a fair distribution of population positioning, the simulation for each enclosure and population will be repeated 100 times. Each time, the results will be added to the result matrix. There will be different matrices, which means the results will be grouped by the population count, as well as enclosure.

There are two enclosures, and three population groups, which makes 6 matrices in general, for Multi-ACO.

The results for shortest path is also stored in a matrix, however, the shortest path algorithm generates a single solution for each simulation. The shortest path simulation is run right after each Multi-ACO simulation to preserve the population count and distribution of the Multi-ACO simulation.

The comparison of the results is done through visualization of the matrices. Since the results are a distribution of numbers, the box-plot method for each objective function seems a feasible solution.

To visualize the fluctuations against the shortest path, results will be summed up. A mean of the medians of the data for each scenario is calculated and visualize as bar charts.

The last part is to calculate the significance of constant **Q** and its effect on the path-updating formula. The new pheromone formula for each edge, as defined by f7 and f8 was as follows:

$$\Delta\tau_{ij} = \sum_{n=0}^{N} \Delta t_{ij}^{n}$$

$$\Delta\tau_{ij}^{n} = \begin{cases} \dfrac{Q}{t_{S_k} \times \sum_{k=1}^{M} \omega_{ij}} & ,\text{if ant } k \text{ passed } ij \text{ in } S_k \\ 0 & ,\text{Otherwise} \end{cases}$$

The denominator has two factors, the egress time of the path, multiplied by total congestion degree of the current edge. These factors are dynamic and depend on size of the premise and number of the agents, and how optimal current solution is, however, regardless of the mentioned factors, Q is always constant in the literature. The reason is that according to pheromone updating function, mentioned bellow, the pheromones get evaporated as iterations increase, and even if $\Delta\boldsymbol{\tau}_{ij}$ is a small amount compared to the initial edge pheromone value, the edge pheromones, $\boldsymbol{\tau}_{ij}$, will evaporate to smaller amounts, and eventually they will level with the $\Delta\boldsymbol{\tau}_{ij}$, so new pheromone becomes effective in path-selection, as it should be. As a reminder, pheromone updating function is as follows:

$$\tau_{ij}(T+1) = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}$$

The only issue is that **Q** should not be so small that it takes a lot of iterations for initial pheromone value to evaporate and they become level. It also should not be so big that in the first iteration, new pheromone value for edges exceeds the initial pheromone value. If this happens, the final results will be biased toward the first edges that were chosen randomly by agents, and this renders the Multi-ACO useless.

Another factor in the pheromone updating function is the evaporation rate $\boldsymbol{\rho}$. This rate clarifies how fast the current pheromone will evaporate, and how fast new pheromone is deposited. As the conclusion, the relation between **Q** and $\boldsymbol{\rho}$ and the denominator of new edge pheromone function which is the egress time multiplied by total congestion degree are the key role players for the efficiency of optimisation.

To watch this event, the results of the first objective function, another battery of simulations, with variable constants is needed. The values for constants **Q** and $\boldsymbol{\rho}$ are taken from the best practices in the literature. This experiment will compare the results of the rasterized objective functions to that of the optimal, defined by the shortest path and optimal congestion degree formula defined before.

The faster iterations get close to the optimal values, more successful the constants are. This experiment will also clarify the needed T_{max}, the maximum iterations needed for Multi-ACO. This is assumed because the material in the literature suggest that after around 150 iterations, the results will reach the maximum efficiency, and do not get better afterward.

3.2 Software Design

This section will present an overview of the implemented software solution.

C# manages packages by namespaces. Each file in a directory belongs to the same package, and classes in packages have access to their methods, except for private methods and properties. Each file may contain multiple classes.

The Object-Oriented Programming (OOP) mindset was chosen for the implementation. As the solution solution boast distinct objects like world graph with edges and nodes, ants, and two solvers: Multi-ACO and shortest paths, they can be easily broken into multiple classes, in different files. The classes were organized in three files, World manager; Shortest Path; and ACO. The class diagram of the solution is shown in Figure 3.8.

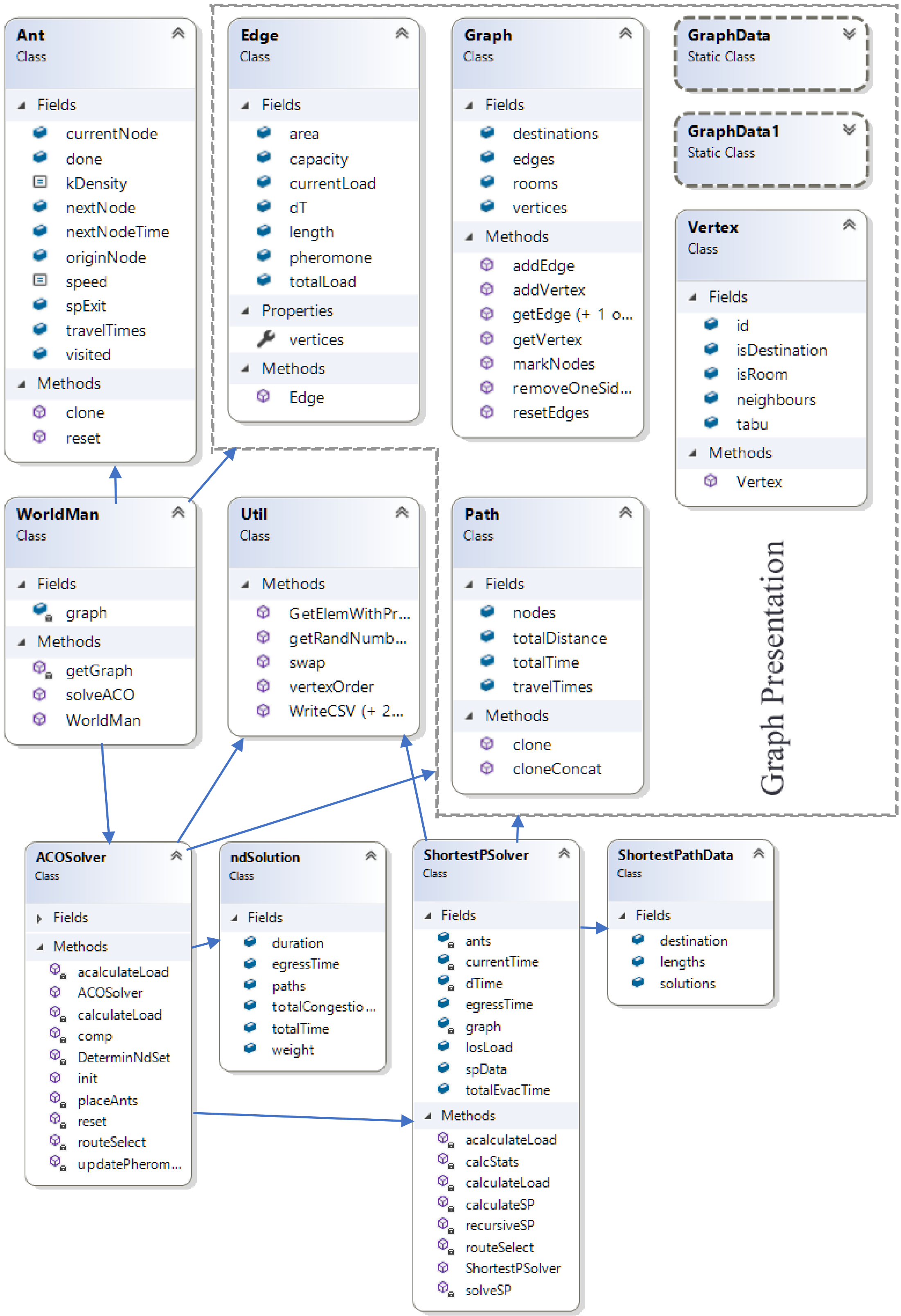


Figure 3.8. Class diagram of the software solution.

The dotted enclosure contains classes that represent the graph, and they are share by all the classes, so to simplify the graph they have been grouped in the diagram.
The order in which the application runs is described here:

1. The worldManager is instantiated by the static entry function (main).
2. The WorldManager instantiates the graph by reading graph data from the static class GraphData. Two GraphData classes are available to address to enclosures.
3. WorldManager instantiates the ACOSolver and passes the graph to it.
4. ACOSolver Instantiates the ShortestPath class and passes the graph to it.
5. ACOSolver solves the shortest path, stores the data inside the ShortestPath-Data class which is a data transfer object (DTO) class, and passes it back to ACOSolver.
6. ACOSolver then starts the iterative process of solution making, ND-Set determination and pheromone updating in a loop until maximum iterations are reached.
7. The output data will be logged using the CSV-Logger method in Util class.
8. If more simulations are needed, application entry will call step one in a loop until desired numbers of simulations are carried out.

4 RESULTS, ANALYSIS & CONCLUSION

This chapter describes findings of the experiment. The figures are presented and analysed during the chapter. The conclusion and the future work is also included in this chapter.

4.1 Findings

This section will discuss the findings during the experiments. First part is about the unexpected phenomenon of biased results, observed during the first instances of simulation, and the solution to this problem.

4.1.1 Biased Optimisation, and Solution

The first instances of the simulation were run for deciding the constant values showed an unexpected result. This simulation was run on the small enclosure, with low population. With these conditions, it is possible to get the optimal results for egress time and total traversed time from the shortest path method, and the optimal congestion degree is zero in this case, because there is no congestion for this population size.

Regardless of the values of the constants $\mathbf{Q}$, $\boldsymbol{\rho}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$, the results of objective functions were scattered over a big range. This conclusion is based on comparing the deviation of the results with the findings in the literature. This means that the results deviated too often from the optimal values. The reason was that in many cases, during a simulation, the objective functions would get stuck on suboptimal results far from the near-optimal results derived from the shortest path algorithm. The result of this experiment is listed in Figure 4.1.

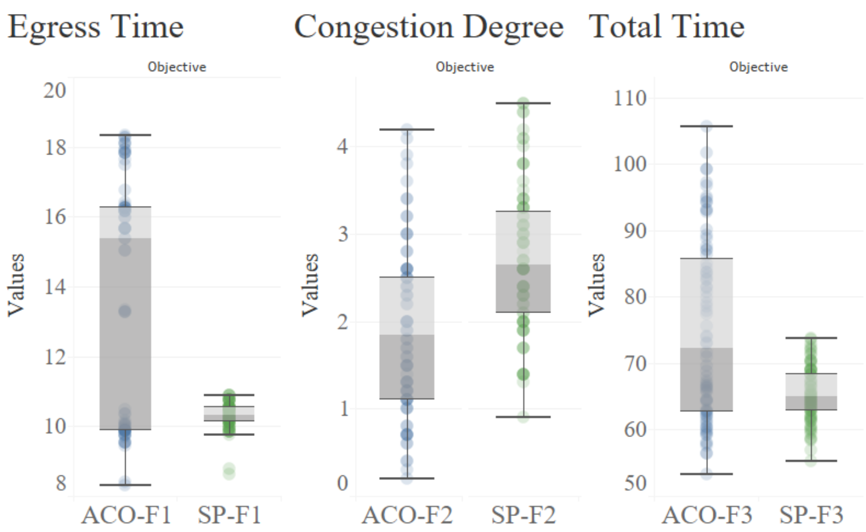


Figure 4.1. Objective function values visualized. ACO is Multi-ACO (blue); SP is the shortest paths (green).

For an optimisation solution, which works based on stochastic metaheuristics, this means the random part of the solution narrows the possibility for change toward a better solution.

This effect is because an outgoing edge from a node may be chosen multiple times in a row, based on random probabilities, and the pheromone on other outgoing edges get evaporated to a very low amount. This means the other edges may not get chosen because the possibility, decided by pheromone, leans toward the mentioned edge quickly, where the other edges may actually produce a more optimal result.

This effect is show in Figure 4.2.

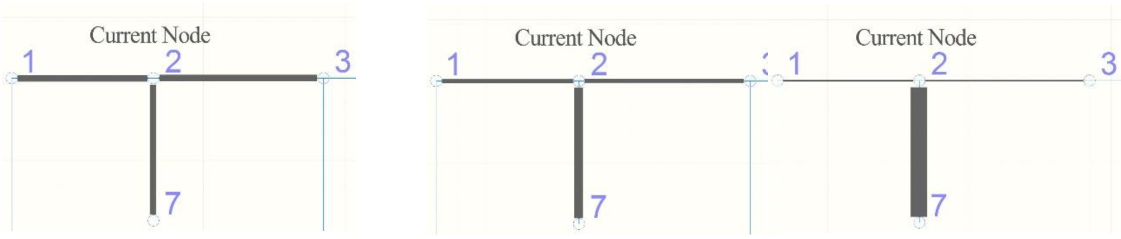


Figure 4.2. Biased pheromone updating. Left side is the default pheromone value. time is incremented for each picture from left to right.

The solution to this problem found in the literature is min-max ant system (MMAS) which sets lower and upper bounds for the pheromone value of an edge. The minimum and maximum amount are defined as multipliers to the initial edge pheromone, τ_0.

The optimal amounts of the minimum and maximum multipliers were found through another experiment, another battery of simulations, with fixed conditions for the small enclosure, and multiple min max values. The experiment decided **min=0.2** and **max=2**. The results of the experiment are shown in Figure 4.3.

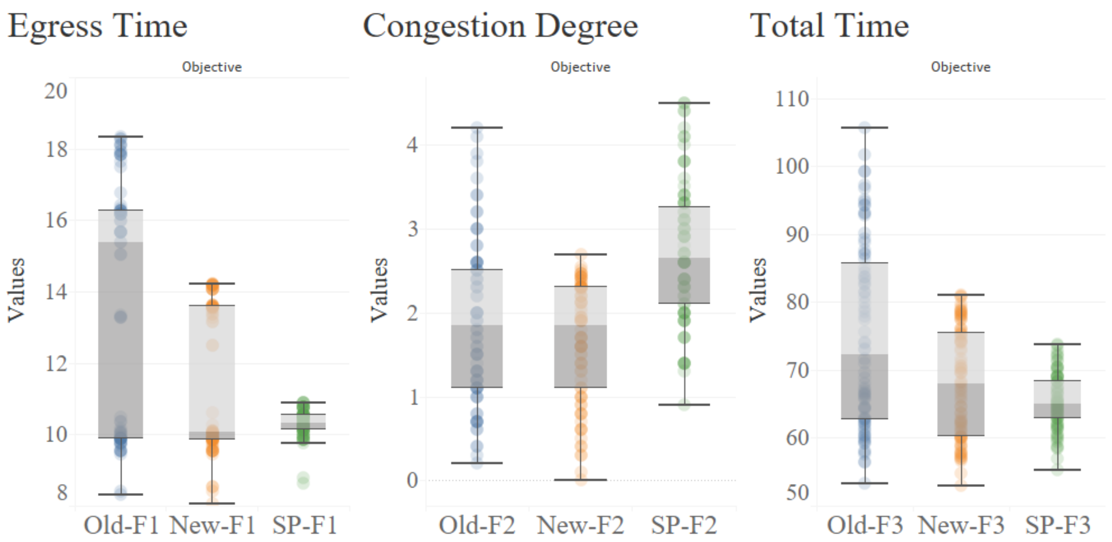


Figure 4.3. Effect of setting min-max pheromone limits on objective functions, as MMAS suggests. Old is Multi-ACO (blue); New is MMAS Multi-ACO (orange); SP is shortest path (green).

With this in mind the **Q**, $\boldsymbol{\rho}$, **α**, and $\boldsymbol{\beta}$ values were extracted from the best practices in the literature, in three groups, as described at the end of section 3.1.4. The objective function experiment with the three constant groups are seen in Figure 4.4 bellow:

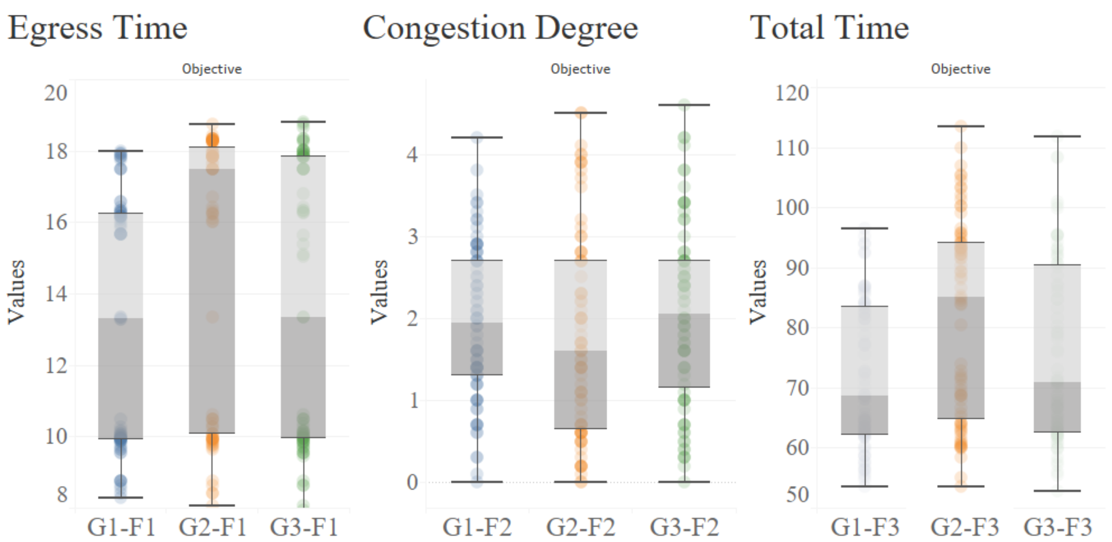


Figure 4.4. Objective function distribution for three constant groups: Group 1, blue (Yuan & Wang, 2007): Q=1, ρ=0.3, α=1, β=3; Group 2, orange (Zong et al., 2010): Q=100, ρ=0.7, α=1, β=3; Group 3, green (Duan, Xiong, & Jiang, 2012): Q=1000, ρ=0.8, α=0.7, β=0.3.

The fact here is that with the presence of min-max approach, the changes in the results, as the mean and the deviation are not as significant and drastic as that of the last experiment, however, there are meaningful fluctuations seen in the results of the next experiment, aimed at examining the improvement rate of the first objective function, seen in Figure 4.5.

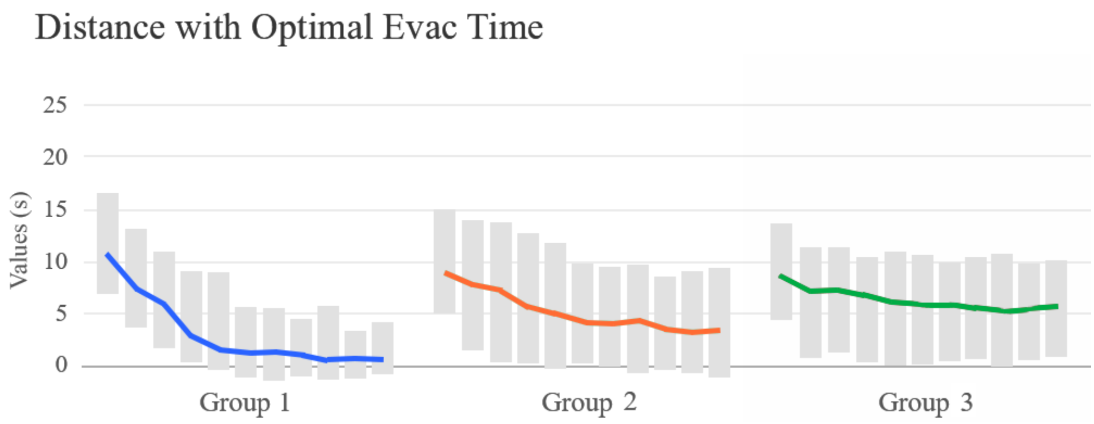


Figure 4.5. The rate of improvement of egress time during simulation iterations for three constant groups. Group 1, blue (Yuan & Wang, 2007): Q=1, ρ=0.3, α=1, β=3; Group 2, orange (Zong et al., 2010): Q=100, ρ=0.7, α=1, β=3; Group 3, green (Duan, Xiong, & Jiang, 2012): Q=1000, ρ=0.8, α=0.7, β=0.3.

According to the experiment, **Q=1**, ***ρ*=0.3**, **α=1**, ***β*=3** were decided. The results of the experiment are shown in Figure 3.zxc.

This conclusion is based on the fact that the amount of pheromone on the edges were monitored during the experiment. For high **Q** values, the pheromone on the edges would quickly reach the higher limit, 2. This means that the optimisation is not taking advantage of the pheromone updating function, and is solely working based on the stochastic part.

4.1.2 The Multi-ACO Comparison

As it is related to the main goal of the research, after deciding the constants, the battery of simulations was run according to the design specifications.

The results of traditional heuristics function, which utilizes the edge load and capacity, for the small enclosure and different population sizes, are seen in Figures below.

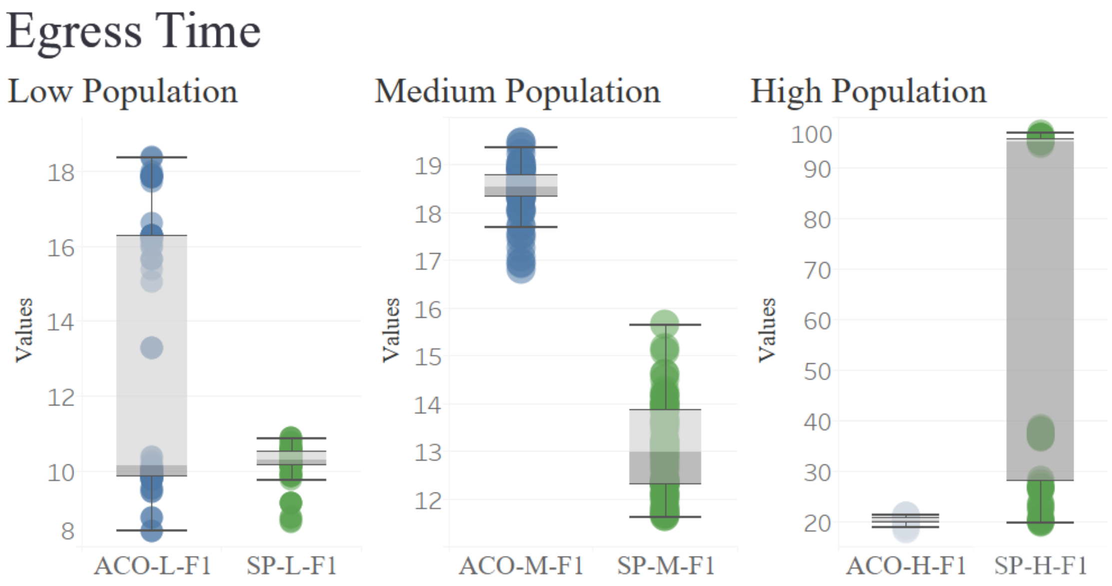


Figure 4.6. Egress time for small enclosure and different population sizes, from left to right, 8, 30, and 60 agents. Multi-ACO is blue, shortest path is green.

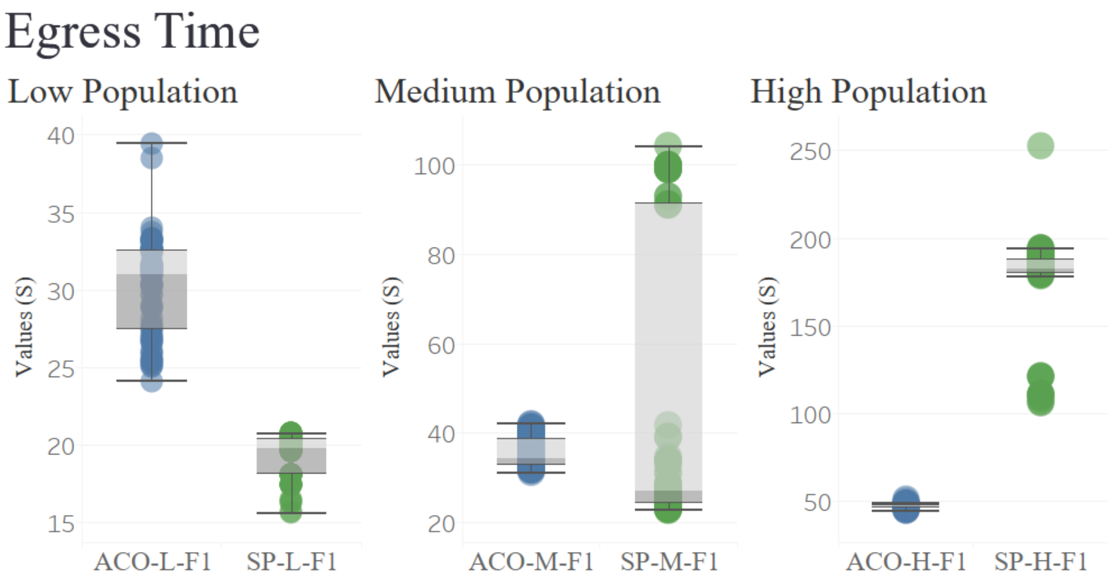


Figure 4.7. Egress time for big enclosure and different population sizes, from left to right, 24, 90, and 180 agents. Multi-ACO is blue, shortest path is green.

For small population sizes, the result of the egress time if the shortest path algorithm is supposed to be optimal. This is because the low congestion degree of such small population should not affect the egress time. The median of the Multi-ACO results is close to that of the shortest paths, however the upper bound bounces to 18 seconds. It is safe to conclude that since distribution of the population is random, some placements cause Multi-ACO perform poorer, and this is expected because of the stochastic nature of Multi-ACO and its pheromone updating process. The shortest paths handle the low population quite solidly for low population in both enclosures.

As it is seen, in the big enclosure, which has a higher route complexity, with the growth of the population, the efficiency of Multi-ACO becomes bolder than the shortest paths. This is also the case with small enclosure, except the low population scenario.

Multi-ACO has the best improvement rate for the high population size, and this becomes bolder in the more complex enclosure, which shows 300% improvement in the egress time.

The shortest path results for big enclosure and high population shows some out of bound data, which is the results of congestions in some randomly formed bottlenecks.

The pattern that is observable here is the steady growth of the Multi-ACO results when the population size grows, for both enclosure sizes. And the shortest path is the opposite of that. This is also the case for Multi-ACO congestion degree for both enclosures, seen in the figures next page.

Another feature of the Multi-ACO compared to the shortest path is the coherency of the produced results. Except for the small population samples in small enclosure, the rest of the results demonstrate denser distribution of the results compared to the shortest paths.

The results of the total traversed time objective function follow the same pattern of the egress time. The results are seen in Appendix A.

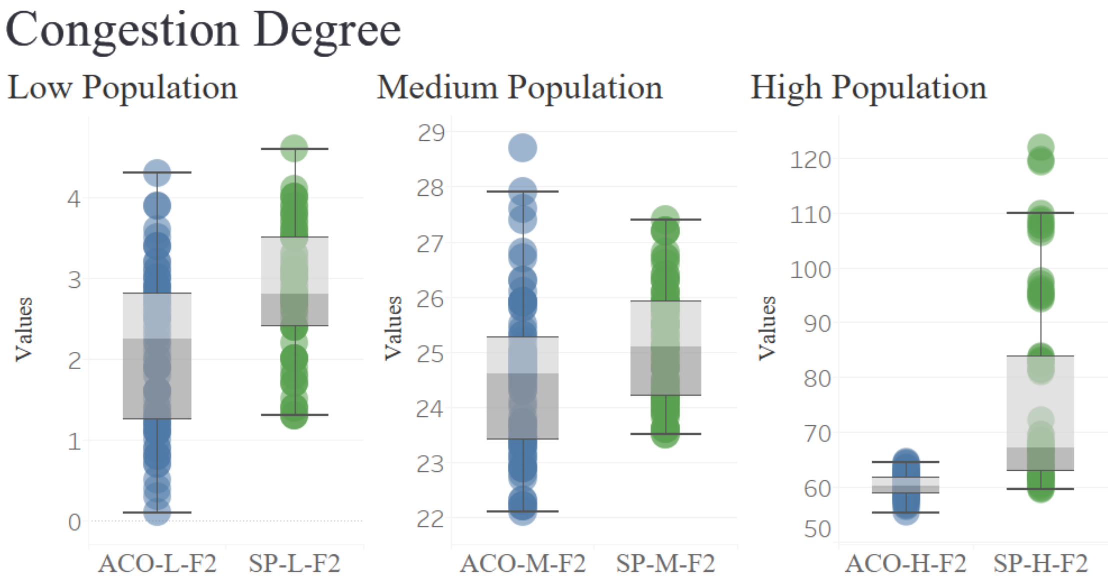


Figure 4.8. Congestion degree for small enclosure and different population sizes, from left to right, 8, 30, and 60 agents. Multi-ACO is blue, shortest path is green.

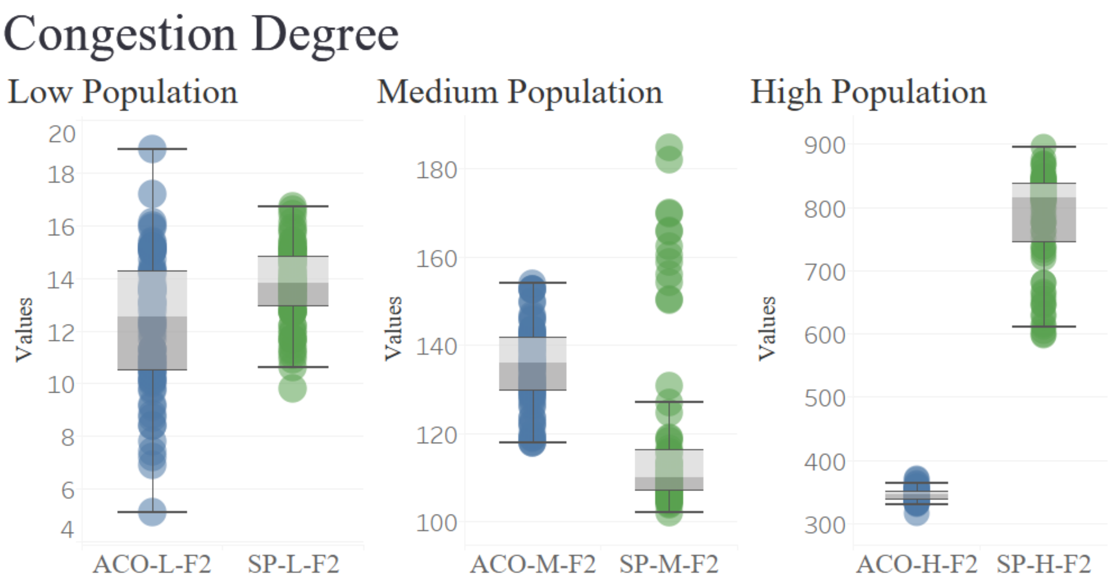


Figure 4.9. Congestion degree for big enclosure and different population sizes, from left to right, 24, 90, and 180 agents. Multi-ACO is blue, shortest path is green.

The congestion degree follows the same pattern for high population samples. Figure 4.9 shows the congestion degree for big enclosure, and Figure 4.8 is for the small enclosure. The only instance that shortest path excels in lowering the congestion is the medium population for the big enclosure.

It seems like the heuristics part of the Multi-ACO effectiveness is related to the population size and has a negative relation to the path complexity. The heuristics function is the η_{ij} defined in f6 in 3.1.1.

For the medium population size, it is expected that some congestions start to form, particularly on the edges closer to the exits. This congestion formation is reflected in the results of the total congestion degree of both Multi-ACO and shortest path algorithms, and becomes bolder for high population size, where shortest path results sky rockets, especially for the big enclosure and high population samples.

The results of the experiments are according to the expectations, except for the egress time of low population in small enclosure which shows a high variance.

The results of the experiment show that the Multi-ACO is conditionally a feasible solution compared to the baseline, however, two important unaddressed issues remain. First is the difference that new Fruin congestion degree calculations make with the trad-itional method, and the second issue is the performance of the Multi-ACO.

Before proceeding to the Fruin LOS evaluation it is important to mention that one of the failed aspects of the experiment was the usage of Fruin LOS data for tracking the total edge load during a single iteration. Since the proposed method is a discrete quantitative method, it produces a discrete number of results for the congestion which is not desirable for load calculation because the proposed method records zero load for LOS level A and this results in zero loaded edges in the denominator of the $\Delta\boldsymbol{\tau}_{ij}^{n}$ formula, described as f11 in 3.1.1. It also results in discrete ranges in next node arrival time for an agent, since the formula (f7) is a linear formula that is corelated to the edge load, which finally results in discrete time results, which is both unrealistic and problematic for ND-Set determination of egress time objective function.

Still, the total congestion degree calculation is done by the Fruin formula. It presents the final congestion degree, and since this objective function value is also used in the ND-Set determination function, hence may affect the final results. Figure 4.10 compares the results of Multi-ACO with Fruin and traditional load calculation method side by side.

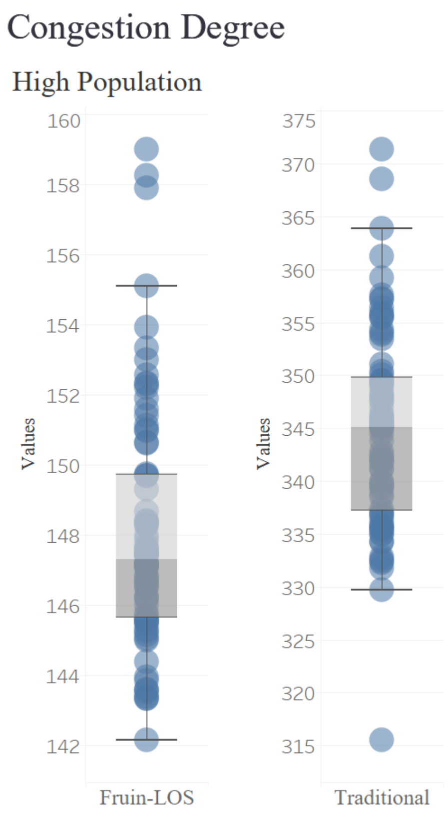


Figure 4.10. Congestion degree for big enclosure, high population. Left is Fruin LOS based Multi-ACO, right is the traditional crowd-density based Multi-ACO.

At the first glance, the congestion degree results for Fruin Multi-ACO are smaller than that of the traditional one. Aside from that, the shape and distribution of the results look the same, except one lower out of bound data for the traditional, which may be a product of stochastic method.

However, there is more to this comparison. As for each iteration, it is possible to calculate the mean congestion score, by dividing total congestion, divided by the number of populated edges. This value which can be calculated with the following formula:

$$\bar{C} = \frac{\sum_{k=1}^{M} \omega_{ij}}{\sum_{k=1}^{M} 1} \times dT$$

This gives the average congestion degree for one second. For Fruin, this value can be directly translated to a LOS, as for the experiment above, the results give a mean congestion value of .32 which according to the Fruin LOS values mentioned in 3.1.1, translates to LOS level C.

As for the traditional method, it gives a person/edge density. It is quite obvious that the Fruin LOS, as a widely accepted Level of Service measure, is more helpful for planners as it can be directly translated to how easy the crowd flows.

The last part of the experiment is the performance of the algorithm, in the manners of computational duration. The following visualization addresses the issue. The experiment was conducted on a machine with Windows 10, Inter® Core™ i5-3230 @ 2.6GH, and 8gb of RAM. The solution took about 25% of CPU load on average.

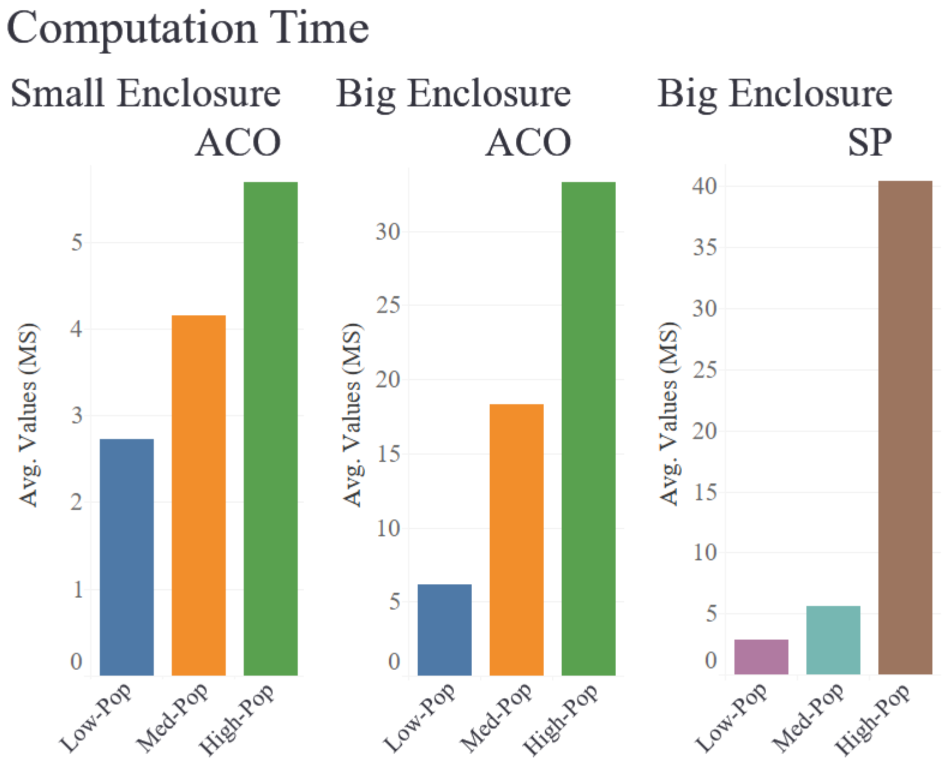


Figure 4.11. Computation time in millisecond, for Multi-ACO in small enclosure left, Multi-ACO in big enclosure middle, Shortest path in big enclosure right.

Please note that the Multi-ACO values are average of one iteration, so for 200 iterations that are run during one simulation, it takes 200 times that.

The Multi-ACO values raise in a linear fashion, relative to the population size. The enclosure size does not have a strong effect on small sizes, it certainly plays an important role in big population samples as the values for high population in big closure is 8 times greater than that of the small enclosure.

The other notable observation is the shortest path, which increases sharply, where it should be independent of the population size of Multi-ACO. This is perceived to be the effect of C# garbage collector during runtime, since the instance of ACO gets heavier and more ram greedy when the population grows. Hence the results for shortest path is incorrect and irrelevant to the point.

The drastic increment in the computation power, where the population grows to 180 agents means that Multi-ACO takes more processing power relative to the population. This is also mentioned in the literature and could threaten the feasibility of the solution in a scenario with a very large population. One of the reasons that researchers pick optimization methods is that these methods are supposed to be quick. The conclusion is that the feasibility of the Multi-ACO from the computational standpoint also depends on the case, and is relative to the population and enclosure size.

4.2 Conclusion

The objectives of the research were tested in this chapter. A Multi-ACO was developed with the Fruin LOS and pedestrian speed integrated. The implementation was successful, however, there was an issue with the usage of LOS as edge load for pheromone updating formula, which resulted in fallback to edge crowd density for this particular formula. This did not affect the usage of Fruin LOS and speed formula for the objective functions.

Then the feasibility of the solution was tested in multiple evacuation scenarios. Two enclosures and three population sizes were tested.

During the tests, it was confirmed that Multi-ACO can improve the results compared to the baseline method which is the shortest path by up to 300%. However, it turned out that the solution is not feasible in all circumstances.

The performance of Multi-ACO depends on the population size and the enclosure size, known as network size. The solution is more effective on the large networks with more agents, however, with the growth of enclosure and the population, the performance of the solution, in terms of CPU cycles also go up. Defining a range for the feasibility of the solution was out of the scope of this research.

As for application of Fruin LOS integrated into Multi-ACO, it turned out that it gives readable results for the average level of service, which can be helpful during the design phase of the building and the ERP, as well as finding the flow degree of evacuees during an evacuation. This can be even more helpful by limiting the edges to a small set, to monitor a specific part of the building, changing them and getting the new average LOS to clarify if the part will act as bottleneck or not.

4.3 Future Work

This paper evaluated the proposed Multi-ACO against the shortest path as the baseline method. While the shortest path method is used widely across the literature, the inability of this method to handle any population shift, that is congestions, makes it hard to give a proper understanding of where Multi-ACO is placed among the methods that are capable of handling the issue, however, the shortest path is still a viable method, since it resembles the human behaviour, that is going to the nearest exit.

For the future work, the proposed method can be tested against another widely used method, like forward-backward shortest paths.

The other aspect that can be evaluated is the efficiency of the method for large enclosures, such as transportation network of a city, with lots of evacuees. This will address the concern of the computation time, which increases rapidly as the network grow.

BIBLIOGRAPHY

Abrahams, J. (1994). Fire Escape in Difficult Circumstances, Design Against Fire, Eds. *Stollard, P., Johnston, L., FN Spon, Great Britain*, 88–96.

Ahmed, R. (2017). What is Tableau? Visualizing Data Using Tableau. Retrieved January 3, 2018, from https://www.edureka.co/blog/what-is-tableau/

Alhajyaseen, W. K. M., Nakamura, H., & Asano, M. (2011). Effects of Bi-directional Pedestrian Flow Characteristics upon the Capacity of Signalized Crosswalks. *Procedia - Social and Behavioral Sciences*, *16*, 526–535. http://doi.org/10.1016/J.SBSPRO.2011.04.473

Beni, G., & Wang, J. (1993). Swarm Intelligence in Cellular Robotic Systems. In *Robots and Biological Systems: Towards a New Bionics?* (pp. 703–712). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-58069-7_38

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, *8*(2), 239–287. http://doi.org/10.1007/s11047-008-9098-4

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, *2*(4), 353–373. Retrieved from http://www.sciencedirect.com/science/article/pii/S1571064505000333

Blum, C., & Merkle, D. (2008). *Swarm intelligence : introduction and applications*. Springer. Retrieved from https://books.google.ie/books?id=6Ky4bVPCXqMC&pg=PA49&lpg=PA49&dq=artificial+ants+leave+pheromone+just+on+their+way+back&source=bl&ots=reazhjMCq9&sig=346mOm3HXCC5gn-OriAuFA7Qnfg&hl=en&sa=X&ved=0ahUKEwiBwJ2U6KDYAhXqB8AKHVsaA8EQ6AEILzAB#v=onepage&q=artifici

BuildingCodeNYC. (2013). Elements of a Means of Egress System. Retrieved October 18, 2017, from http://buildingcodenyc.com/sketches/elements-of-a-means-of-egress-system/

Chen, X., Kwan, M.-P., Li, Q., & Chen, J. (2012). A model for evacuation risk assessment with consideration of pre- and post-disaster factors. *Computers, Environment and Urban Systems*, *36*(3), 207–217. http://doi.org/10.1016/j.compenvurbsys.2011.11.002

Chien, K. F., Wu, Z. H., & Huang, S. C. (2014). Identifying and assessing critical risk factors for BIM projects: Empirical study. *Automation in Construction*, *45*, 1–15. http://doi.org/10.1016/j.autcon.2014.04.012

Daamen, W., Hoogendoorn, S., & Bovy, P. (2005). First-Order Pedestrian Traffic Flow Theory. *Transportation Research Record: Journal of the Transportation Research Board*, *1934*, 43–52. http://doi.org/10.3141/1934-05

Department of Homeland Security. (n.d.). Emergency Response Plan. Retrieved October 17, 2017, from https://www.ready.gov/business/implementation/emergency

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. http://doi.org/10.1007/BF01386390

Dorigo, M., Caro, G. Di, & Gambardella, L. M. (1999). Ant Algorithms for Discrete Optimization. *Artificial Life*, *5*(2), 137–172. http://doi.org/10.1162/106454699568728

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B IEEE Transactions on Systems, Man, and Cybernetics, Part–B, 2626*(11), 1–131. Retrieved from http://iridia.ulb.ac.be/dorigo/dorigo.html

Duan, P., Xiong, S., & Jiang, H. (2012). Multi-objective optimization model based on heuristic ant colony algorithm for emergency evacuation. In *2012 15th International IEEE Conference on Intelligent Transportation Systems* (pp. 1258–1262). IEEE. http://doi.org/10.1109/ITSC.2012.6338611

Fang, Z., Zong, X., Li, Q., Li, Q., & Xiong, S. (2011). Hierarchical multi-objective evacuation routing in stadium using ant colony optimization approach. *Journal of Transport Geography*, *19*(3), 443–451. http://doi.org/10.1016/J.JTRANGEO.2010.10.001

Fruin, j, j. (1971). PEDESTRIAN PLANNING AND DESIGN. Retrieved January 2, 2018, from https://books.google.ie/books?id=bRHgnQAACAAJ&dq=Pedestrian+Planning+and+Design&hl=en&sa=X&ved=0ahUKEwiBwuznmbrYAhVkF8AKHdd4DJ8Q6AEILjAB

Fruin, J. (1992). Designing for pedestrians. *Public Transportation United States.* Retrieved from https://ntl.bts.gov/DOCS/11877/Chapter_8.html

Georgiadou, P. S., Papazoglou, I. A., Kiranoudis, C. T., & Markatos, N. C. (2007). Modeling emergency evacuation for major hazard industrial sites. *Reliability Engineering & System Safety*, *92*(10), 1388–1402. http://doi.org/10.1016/j.ress.2006.09.009

Glover, F., & Sörensen, K. (2015). Metaheuristics. *Scholarpedia*, *10*(4), 6532. http://doi.org/10.4249/scholarpedia.6532

Greco, S., Figueira, J., & Ehrgott, M. (2005). Multiple criteria decision analysis. *Springer's International Series*. Retrieved from http://link.springer.com/content/pdf/10.1007/978-1-4939-3094-4.pdf

Guntsch, M., & Middendorf, M. (2003). Solving Multi-criteria Optimization Problems with Population-Based ACO (pp. 464–478). Springer, Berlin, Heidelberg. http://doi.org/10.1007/3-540-36970-8_33

Gwynne, S., Galea, E. R., Owen, M., Lawrence, P. J., & Filippidis, L. (1999). A review of the methodologies used in the computer simulation of evacuation from the built environment. *Building and Environment*, *34*(6), 741–749. http://doi.org/10.1016/S0360-1323(98)00057-2

Koo, C., Hong, T., & Kim, S. (2015). An integrated multi-objective optimization model for solving the construction time-cost trade-off problem. *Journal of Civil Engineering and Management*, *21*(3), 323–333. http://doi.org/10.3846/13923730.2013.802733

Liu, M., Zhang, F., Ma, Y., Pota, H. R., & Shen, W. (2016). Evacuation path optimization based on quantum ant colony algorithm. *Advanced Engineering Informatics*, *30*(3), 259–267. http://doi.org/10.1016/j.aei.2016.04.005

López-Ibáñez, M., Paquete, L., & Stützle, T. (2004). On the Design of ACO for the Biobjective Quadratic Assignment Problem (pp. 214–225). Springer, Berlin, Heidelberg. http://doi.org/10.1007/978-3-540-28646-2_19

Neufert, E., Neufert, P., & Kister, J. (2012). *Architects' data*. Wiley-Blackwell.

Pelechano, N., & Malkawi, A. (2008). Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. *Automation in Construction*, *17*(4), 377–385. http://doi.org/10.1016/J.AUTCON.2007.06.005

Rouse, M. (2007). What is C#? - Definition from. Retrieved January 3, 2018, from http://searchwindevelopment.techtarget.com/definition/C

Saeed Osman, M., & Ram, B. (2017). Routing and scheduling on evacuation path networks using centralized hybrid approach. *Computers & Operations Research*, *88*, 332–339. http://doi.org/10.1016/j.cor.2017.06.022

Santos, G., & Aguirre, B. E. (2004). A Critical Review Of Emergency Evacuation Simulation Models. Retrieved from http://udspace.udel.edu/handle/19716/299

Shen, T.-S. (2006). Building Egress Analysis. *Journal of Fire Sciences*. http://doi.org/10.1177/0734904106052549

Singapore Civil Defense Force. (2013). Evacuation Planning Guidelines. Retrieved from https://www.scdf.gov.sg/content/scdf_internet/en/building-professionals/fire-safety-permit-and-certification/emergency-response-plan1/_jcr_content/par/download_3/file.res/ERP - Evacuation Planning Guidelines v2.pdf

Smith College. (2015). Dijkstra's Algorithm. Retrieved December 29, 2017, from http://cs.smith.edu/~streinu/Teaching/Courses/274/Spring98/Projects/Philip/fp/dijkstra.htm

Still, K. (2000). *Crowd Dynamics*. University of Warwick. Retrieved from http://www.gkstill.com/CV/PhD/CrowdDynamics.html

Tanimoto, J., Hagishima, A., & Tanaka, Y. (2010). Study of bottleneck effect at an emergency evacuation exit using cellular automata model, mean field approximation analysis, and game theory. *Physica A: Statistical Mechanics and Its Applications*, *389*(24), 5611–5618. http://doi.org/10.1016/j.physa.2010.08.032

Triantaphyllou, E. (2000). *Multi-criteria Decision Making Methods: A Comparative Study* (Vol. 44). Boston, MA: Springer US. http://doi.org/10.1007/978-1-4757-3157-6

Vanumu, L. D., Ramachandra Rao, K., & Tiwari, G. (2017). Fundamental diagrams of pedestrian flow characteristics: A review. *European Transport Research Review*, *9*(4), 49. http://doi.org/10.1007/s12544-017-0264-6

Wagoum, A. U. K., Tordeux, A., & Liao, W. (2017). Understanding human queuing behaviour at exits: an empirical study. *Royal Society Open Science*, *4*(1), 160896. http://doi.org/10.1098/rsos.160896

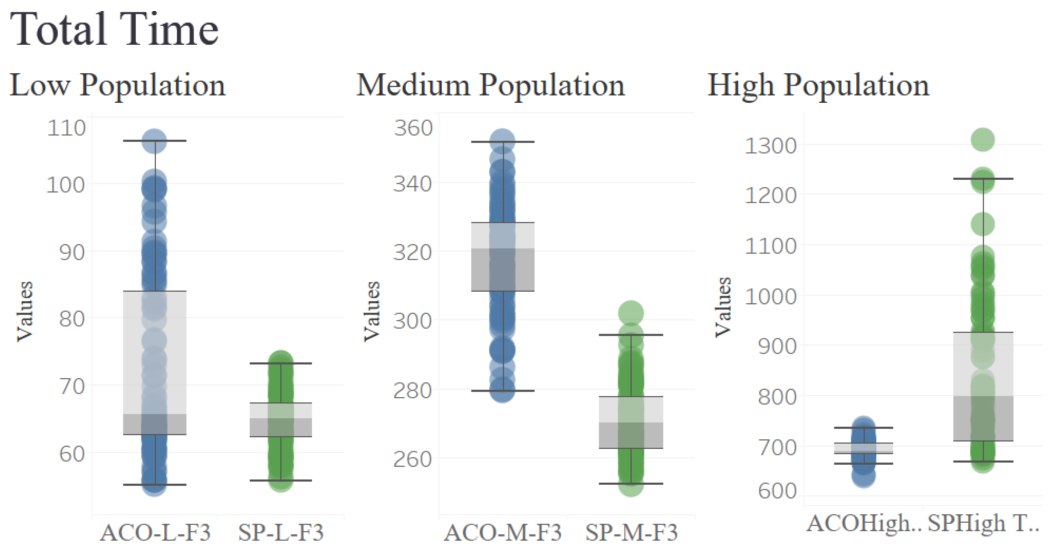
Yi-fan, L., Jun-min, C., Jie, J., Ying, Z., & Jin-hua, S. (2011). Analysis of Crowded Degree of Emergency Evacuation at "Bottleneck" Position in Subway Station Based on Stairway Level of Service. *Procedia Engineering*, *11*, 242–251. http://doi.org/10.1016/j.proeng.2011.04.653

Yi, W., & Kumar, A. (2007). Ant colony optimization for disaster relief operations. *Transportation Research Part E: Logistics and Transportation Review*, *43*(6), 660–672. http://doi.org/10.1016/j.tre.2006.05.004

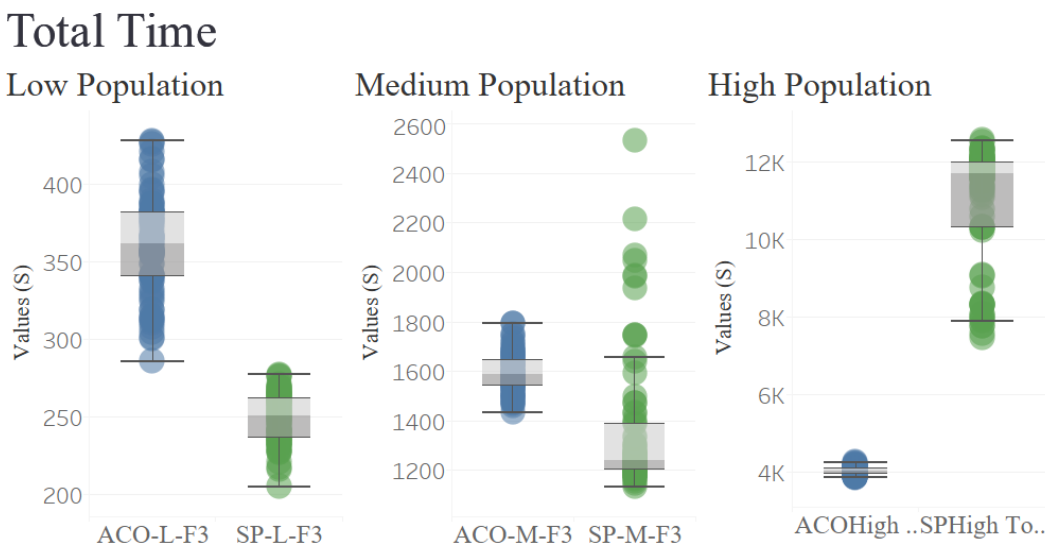
Yuan, Y., & Wang, D. (2007). Multi-Objective Path Selection Model and Algorithm for Emergency Evacuation. In *2007 IEEE International Conference on Automation and Logistics* (pp. 340–344). IEEE. http://doi.org/10.1109/ICAL.2007.4338583

Zong, X., Xiong, S., Fang, Z., & Li, Q. (2010). Multi-Objective Optimization for Massive Pedestrian Evacuation Using Ant Colony Algorithm. In *International Conference in Swarm Intelligence* (pp. 636–642). Springer, Berlin, Heidelberg. http://doi.org/10.1007/978-3-642-13495-1_78

APPENDIX A: TOTAL TRAVERSED TIME FIGURES



Total traversed time for small enclosure and different population groups. Blue is Multi-ACO, green is shortest path. Distribution of the results resembles that of the egress time.



Total traversed time for small enclosure and different population groups. Blue is Multi-ACO, green is shortest path.