Route choice modelling in fire evacuation simulators

Kristian Bladström

Fire Safety Engineering
Lund University
Sweden

Brandteknik
Lunds tekniska högskola
Lunds universitet

Report 5528, Lund 2017

Route choice modelling in fire evacuation simulators

Kristian Bladström

Lund 2017

Title
Route choice modelling in fire evacuation simulators
Modellering av vägval hos evakueringssimulatorer för brand

Author
Kristian Bladström

Report 5528
ISRN: LUTVDG/TVBB--5528--SE

Number of pages: 80
Illustrations: by author if not stated otherwise

Keywords
Route choice, algorithms, evacuation, path finding.

Abstract
This thesis makes a review of existing algorithms and methods for route choice. Algorithms and methods already implemented in fire evacuation simulators are examined, as well as those that are implemented in computer games, traffic models or theoretical models that are to date not implemented at all. An analysis is made to get an overview of their strengths and limitations to be able to determine which seem promising to develop further. This thesis has an extra focus on factors affecting route choice and their implementation in evacuation models. Factors found in literature are summarized. Furthermore, a series of 8 tests for verification of factors affecting route choice in evacuation simulators are suggested.

Brandteknik
Lunds tekniska högskola
Lunds universitet
Box 118
221 00 Lund

www.brand.lth.se

Telefon: 046 - 222 73 60

Fire Safety Engineering
Faculty of Engineering
Lund University
P.O. Box 118
SE-221 00 Lund
Sweden

www.brand.lth.se

Telephone: +46 46 222 73 60

ACKNOWLEDGEMENTS

The author would like to thank following person for his help with this thesis.

Enrico Ronchi
Associate senior lecturer at the Department of Fire Safety Engineering, Lund University, for his inciteful comments, directions, recommendations and general support during the whole process of writing this thesis.

Kristian Bladström
Lund 2017

SUMMARY

Evacuation models are used to simulate human behaviour in evacuation scenarios. In Fire Safety Engineering, they are used to calculate the time it takes to evacuate a building in case of fire. The total evacuation time can be calculated with evacuation models or hand calculations. The advantage of evacuation models is that it is possible to simulate human behaviour. When simulating evacuation, human behaviour is an important factor that needs to be addressed. Most of the algorithms and methods described in this paper do not fully implement human behaviour when making a route choice during evacuation. Some models do, but not all factors that might affect route choice are implemented. Since the validation and verification of evacuation models are the most important factors to users, tests for the assessment of route choice methods are needed.

The purpose of this thesis is to make a review of algorithms and methods affecting route choice, evaluate their strengths and limitations, summarize factors affecting route choice and suggest test for verification of route choice. To achieve the objectives, existing literature in the area is investigated, information about existing algorithms including their functionalities, strengths and limitations are collected, classified, compiled and analysed. The algorithms and methods are divided into the following categories: *shortest path, quickest path* and c*onditional path.* Variables or factors that might affect the route choice are summarized. These variables have been found in literature and are used in existing simulation models but are also variables that have been found to affect the route choice in real life evacuations.

The literature review revealed that most of the shortest path algorithms are derived from the A* or the Dijkstra's algorithm. In the quickest path category, most methods belonged to the logit family. In the conditional path category, there were not as many algorithms or methods. Those categorised as conditional were more frequently implementing human behaviour.

When evaluating the strengths and limitations of the algorithms, the following factors have been taken into consideration: c*omputational cost*, *fast searches, capability to handle dynamic environments, factors concerning human behaviour implemented* and o*ther (for example if it simulates unrealistic paths, or it is guaranteed to find a path)*. However, no complete information was available on the characteristics of all methods. Whether an algorithm or method implemented human behaviour or not was the only characteristic for which information was found for every method. The comparison between algorithms and methods has therefore mainly relied on that information.

Some of the factors that can affect route choice in fire evacuation can be evaluated using verification tests. Therefore, verification tests are suggested for the following factors affecting route choice: visual perception, queuing time, walking time, herding, leadership, radiation, smoke, temperature and obstacles. Nevertheless, also in this area there is a need for further investigation to be able to create verification tests for more factors.

The main factors implemented in route choice algorithms and methods today are queuing time, walking time, visual perception, herding, exit block/obstacles, affiliation, smoke, elevators and temperature. Among the most common evacuation models today based on a survey (Ronchi & Kinsey, 2011), FDS+Evac and EXODUS are those that implements most factors concerning human behaviour. Simulex is the only one of the more common models that do not explicitly represent human behaviour in its algorithm for route choice.

SAMMANFATTNING

Evakueringsmodeller används för att simulera mänskligt beteende under evakuering. Inom brandteknik används de för att beräkna den tid det tar att utrymma en byggnad. Detta kan göras med evakueringsmodeller eller handberäkningar. Fördelen med evakueringsmodellerna är att de gör det möjligt att simulera mänskligt beteende. Vid simulering av evakuering är mänskligt beteende en viktig faktor som behöver diskuteras. De flesta av de algoritmer och metoder som beskrivs i detta dokument implementerar inte mänskligt beteende vid vägval under evakuering. Vissa modeller gör det, men då används inte alla faktorer som kan påverka vägvalet. Eftersom validering och verifiering av utrymningsmodeller är vad som är viktigast för användarna vid val av modell, behövs verifieringstester för vägval i modellerna.

Syftet med examensarbetet är att göra en sammanfattning av algoritmer och metoder som påverkar vägval, utvärdera deras styrkor och begränsningar, sammanfatta faktorer som påverkar vägval och föreslå tester för verifiering av vägval. För att slutföra detta undersöks befintlig litteratur på området. Information om befintliga algoritmer inklusive deras funktion, styrkor och begränsningar samlas in och klassificeras, sammanställas och analyseras. Algoritmerna och metoderna för vägval är indelade i följande kategorier: *kortaste vägen, snabbaste vägen* och *villkorlig väg*. Variabler eller faktorer som kan påverka vägvalet sammanfattas. Dessa variabler har återfunnits i litteraturen och används i befintliga simuleringsmodeller, men är också variabler som har visat sig påverka vägvalet vid verkliga evakueringar.

De flesta av algoritmerna för kortaste vägen är härledda från A* eller Dijkstras algoritm. För kategorin med snabbaste vägen tillhörde flest metoder familjen av logit-funktioner. I kategorin för villkorlig väg fanns inte lika många algoritmer eller metoder. Men i kategorin för villkorlig väg var en större del mer avancerade och implementerade mänskligt beteende.

Vid bedömningen av styrkor och begränsningar har följande parametrar varit av intresse: *beräkningskostnad, snabba sökningar, möjlighet att hantera dynamiska miljöer, faktorer för mänskligt beteende implementerade* och *övrigt (till exempel om den simulerar orealistiska vägar, eller garanterat kommer hitta en väg)*. Ingen fullständig information var tillgänglig rörande metodernas egenskaper. Den enda informationen funnen för samtliga algoritmer och metoder var huruvida de implementerade mänskligt beteende. Jämförelserna mellan olika algoritmer och metoder har därför till största del grundat sig på den informationen.

Några av de faktor som kan påverka vägval under evakuering vid brandscenarion kan utvärderas med verifieringstest. Verifieringstester föreslås därför för följande faktorer som kan påverka vägval vid utrymning: visuell uppfattning, kötid, gångtid, gruppinteraktion, ledarskap, strålning, rök, temperatur och hinder. Men även inom detta område behövs mer information för att kunna föreslå ytterligare verifieringstest för flera faktorer.

De faktorer som idag är implementerade i modeller för evakueringssimulering vid brand är kötid, gångtid, visuell uppfattning, gruppbildning, blockering av vägar, anknytning till en väg eller utgång, rök, hissar och temperatur. Bland de vanligaste modellerna för evakueringssimulering idag enligt en undersökning (Ronchi & Kinsey, 2011) är det FDS+Evac och EXODUS som implementerar flest faktorer för mänskligt beteende. Simulex är den enda av de vanliga modellerna som inte alls använder sig av mänskligt beteende i sin algoritm för vägval.

TABLE OF CONTENTS

1. INTRODUCTION

Evacuation models are tools for simulating human behaviour when evacuating buildings (Ronchi & Nilsson, 2016). In Fire Safety Engineering, they are used to investigate if a building has an acceptable level of safety when it comes to evacuating during fire emergencies. To calculate this a comparison is made between the available safe egress time (ASET) and the required safe egress time (RSET). ASET is the duration of time in which the conditions are acceptable and evacuation can be done. RSET is the duration of time that is required to travel to a safe place. For the building to be safe, the available time should be larger than the required time, i.e. ASET>RSET (Purser, 2003; Gwynne & Rosenbaum, 2016). This is an example of performance-based design which is defined as "*an engineering approach to fire protection design based on (1) agreed upon fire safety goals and objectives, (2) deterministic and/or probabilistic analysis of fire scenarios, and (3) quantitative assessment of design alternatives against the fire safety goals and objectives using accepted engineering tools, methodologies and performance criteria*." (SFPE, 2007).

The calculation of RSET can be done with evacuation models or hand calculations. The equations used in hand calculations assume that people move towards an exit in the same way as a fluid. However, these calculations do not implement human behaviour, which can affect the RSET by adding time to make a decision to evacuate for example. In contrast, evacuation models make it possible to simulate human behaviour, and some models today implement different behavioural sub-models (Ronchi & Nilsson, 2016).

The majority of evacuation models uses a time-line model. This model can be thought of as a simplification of human behaviour, adding time for different phases of the evacuation to a total RSET. The different phases that together add up as a RSET are (Purser, 2003; Gwynne & Rosenbaum, 2016):

- *The detection time*: the time from the start of the fire to when someone detects the fire.
- *The alarm time*: the time from the detection of the fire to when a general alarm goes off.
- *The pre-evacuation time*: the sum of *recognition time* and *response time*. *Recognition time* is the time from that the alarm goes off to before people (from here on out people is defined as occupants in the real world) start responding. *The response time* is the time from when people recognize the alarm and start responding, but not travel.
- *The travel time*: the time to travel from the current place to a safe place.

The space representation in evacuation models can be divided into four categories; coarse network, fine network, continuous or hybrid (Ronchi & Nilsson, 2016). The coarse network approach uses nodes connected with arcs, where the nodes usually represents rooms and the movement is done along the arcs. The nodes can be given certain characteristics, for example an upper limit for the number of agents in that node. The fine network approach can either use a grid with cells or a fine network of nodes and arcs. The cells in the grid can be of different shapes, for example squares or hexagons. With this representation of the geometry the agents move from cell to cell, not being more than one agent in each cell at one time. With the network of nodes and arcs, the agents move from node to node along the arcs, usually not being more than one agent at one time in each node. The difference from the coarse network approach is that the number of nodes and arcs are larger and therefore the number of possible routes is larger. The geometry in continuous models are not made up by fixed data points or

cells that the agents can move along or in. Instead the geometry is more complex being a continuous space that the agents can move freely in. The agent's location, body shape and size is described by coordinates. Furthermore, the agent has rules for the minimum distance to other agents and obstacles (Ronchi & Nilsson, 2016). The hybrid approach for space representation in evacuation models uses a combination of the aforementioned approaches, which is computationally efficient and at the same time represents the movement of agents in a realistic way (Chooramun, 2011).

The movement in the models are divided into two categories; macroscopic and microscopic (Hamacher & Tjandra, 2001; Ronchi & Nilsson, 2016). The macroscopic models are described as using an optimization approach where movement is considered only at an aggregate level. In these models, no consideration to human behaviour or individual differences when choosing exit route are made, and the agents (from here on, the term agents is used to define occupants in the evacuation models) are viewed on as a homogeneous group moving as a two-dimensional artificial fluid. The microscopic models on the other hand emphasizes individual movement and considers factors like walking speed, physical ability and interactions with other occupants (Hamacher & Tjandra, 2001; Ronchi & Nilsson, 2016).

In a review by Kuligowski (2016), modelling methods are divided into the following subgroups: Behavioural models, Movement models and Partial behaviour models. Behavioural models are described as models where occupants can perform actions and incorporate decision-making, like for example when being exposed to certain conditions in the building. Movement models are described as models where occupants move from point A to point B without taking human behaviour into account when choosing exit route. Partial behaviour models are described as models where primarily occupant movement is calculated, but the models allow the simulation of some basic behaviours. Two more categories are mentioned and those are: Movement/optimization models and Behavioural model with risk assessment capabilities (Kuligowski, 2016).

For the agents in the model to move towards a safe place, an algorithm or method for route choice is implemented. There can be different algorithms to assign route choice globally and locally. Globally refers to the route that is used to reach the final exit, and locally refers to the route taken to intermediate goals, or if re-routing is made. The majority of the models has occupants that from the beginning of the simulation know which way to go. A few models however assign the route choice based on visual perception, i.e. the occupant has to find its way to the exit rather than knowing where to go (Ronchi & Nilsson, 2016).

Algorithms for route choice can for example use distance maps (Thompson, 1994). That means that a space has a numerical value which correlates to the distance that is needed to reach an exit from that space. Other algorithms used in path-finding to find the shortest route are the A* algorithm (Hart, Nilsson, & Raphael, 1968) or the Dijkstra's algorithm (Misa, 2010). These are however not very commonly implemented in evacuation models today according to Ronchi & Nilsson (2016). Another way of choosing the path is with the use of vector fields that gives the preferred directions of the occupants, which can be calculated in different ways. This can be done by approximating the evacuation to a flow of a two-dimensional incompressible fluid that flows towards the exit. The route choice can either be made by the user (i.e. deterministic), or calculated with an algorithm that is either deterministic or probabilistic (Ronchi & Nilsson, 2016).

With a deterministic algorithm, the agents will always choose the same exit route when simulating the same scenario multiple times, regardless of which algorithm is used. Probabilistic models assume that each agent chooses its path based upon probabilities for different routes. While the deterministic models get the same outcome with the same inputs, the probabilistic might not (Prato, 2009).

Two main categories of algorithms focus on the calculation of the shortest or quickest route. There are also algorithms that incorporate more complex decision-making and behaviour in humans in relationship to different conditions (Ronchi & Nilsson, 2016).

Several models that describes human behaviour in fire scenarios are available in the literature where the most important models are mentioned as Behaviour sequence model (Canter, 1990), Role-rule model (Canter, 1990; Tong & Canter, 1985), Affiliative model (Sime, 1985) and Social influence model (Deutsch & Gerard, 1955; Latané & Darley, 1970). The models make it possible to understand decision making amongst people during fire evacuations in a qualitatively way. Unfortunately, it is hard to use those models in evacuation simulators, because they are often difficult to implement and it may be difficult to get results that are of interest (Ronchi & Nilsson, 2016).

When using algorithms that implements human behaviour, different factors affecting route choice are applied. In the review by Kuligowski (2016) the variables mentioned in algorithms are knowledge of the building, interactions with other people like herding or following a leader and information about the fire like smoke or temperature. Knowledge of the building describes how familiar the occupant is with the layout of the building. If an occupant is more familiar with an exit it can be affiliated to it, meaning that it is more attached to that exit (Sime, 1985). An occupant can also be affiliated with another occupant or person, but in this thesis the focus will be on place affiliation. Radiation is also a fire related factor affecting route choice that is used in the BR-radiation model (Bae, Choi, Kim, Hong, & Ryou, 2016). Ronchi & Nilsson (2016) mentions factors like available egress capacity, smoke and group interactions. Thompson (1994) mentions factors like alarm awareness, type of alarm (auditory, visual, olfactory and tactual cues) and effect of signage. Signage is something that can affect significantly the chosen exit route, where exits that are not signposted, or has sings in languages the occupant is not familiar with might not be used or cause confusion (Thompson, 1994). When testing evacuation in a high-rise building in virtual reality, it was shown that wayfinding installations had a significant impact on the chosen route, when choosing between stairs and evacuation elevators. The experiments showed that when using wayfinding installations, the occupants were more likely to use the evacuation elevators (Andrée, Nilsson, & Eriksson, 2016). In traffic, drivers chosen route can depend on the presentation of the information from signage. Whether it is static or dynamic for example (Dia, 2002; Ronchi, Nilsson, Modig, & Walter, 2016; Olander, Ronchi, Lovreglio, & Nilsson, 2017). Xie (2011) lists evacuation models that use signage systems and explains how some models implements it. In the evacuation model EXODUS, version 4.06 (Gales, et al., 2006) for example, the line of sight between an occupant and a sign depends on the distance from which the sign can be seen, the occupant's height, the vertical placement of the sign and the height of possible objects in between the occupant and the sign. The angle from which the occupant is placed relative to the sign affects the probability to see the sign. There is however a lack of information about how signage affects occupants route choice (Xie, 2011). Ronchi

(2014) mentions the following factors; social influence, visibility of doors, line of sight and affiliation.

Familiarity is as an important factor when choosing exit route (Thompson, 1994). Many evacuation models assign the occupants to the closest exit, even though research has shown that it is more likely that occupants use an exit that is familiar, or they adopt the way they entered the building. If the building has staff that can inform the occupants of which exit routes to use, it could affect the chosen exit route (Thompson, 1994). In a study by Horiuchi et al. (1986) considering smoke-filled environment, the factor that had the biggest impact on the route choice was mentioned as amount of smoke. But also, familiarity with the building affected the chosen route. In the study, it was stated that if the occupant is familiar with the building, a regularly used route would be used. But if the occupants were not familiar with the building, they would follow or rely on others. Occupants familiar with the building can even find the exits in heavy smoke (Horiuchi, Murozaki, & Hukugo, 1986) Few attempts have been made to simulate complex psychological behaviour, for example the algorithm in FDS+Evac (Korhonen, 2015). Factors included in those algorithms are leadership, group flocking and decisions like helping other occupants, investigate the fire or leaving the building. However, there is a lack of data to develop and validate such algorithms (Thompson, 1994).

When choosing which evacuation model to use, validation and verification of the model seems to be the most important according to a survey (Ronchi & Kinsey, 2011). However, there is a lack of standard protocols for verification and validation of evacuation models (Ronchi , 2014). This applies also to the verification of route choice algorithms employed in evacuation models, which are of interest to this paper. By suggesting verification tests and listing factors affecting route choice that need to be subjected to testing, a first step towards better route choice algorithms is taken since it is possible to identify which factors are considered in route choice modelling. The validation is left out of this thesis since it would require collection and analysis of experimental data on route choice in different conditions which is not within the scope of this work.

1.1 Purpose and objective

The purpose of this thesis is to perform a systematic review of the route choice algorithms that are currently used and can be used in the future in fire evacuation models and evaluate the factors which are included in them. This is done to get a better understanding of the algorithms currently used, and to be able to further develop them in the future. The thesis focuses on the evaluation of different algorithms and assumptions used to assign the evacuation route to agents when the evacuation is simulated with computer models.

The objectives of this thesis are as follows.

- To analyse different algorithms used to assign evacuation route amongst each other.
- To identify the variables that are currently considered in route choice algorithms.
- To recommend a set of verification tests for the analysis of the route choice algorithm used in evacuation models.
- To review algorithms for route choice that are not used in fire evacuation today, but that could be potentially used in the future. These algorithms can currently be implemented in traffic assignment models, computer games or artificial intelligence.

The following questions should be answered.

- What is the state-of-art of route choice algorithms used in fire evacuation models?
- What are the strengths and limitations of existing route choice algorithms used in fire evacuation models?

1.2 Method

To complete the objectives and answer the questions in the chapter above, existing literature in the area is investigated, information about existing algorithms including function, strengths and limitations are collected and classified, compiled and analysed. Conclusions and a discussion are also done.

At first, a literature study was performed. In this phase a thorough review of the existing material that present information on how the algorithms work was done. This included reading manuals and technical references on how different programs assign the evacuation route, scientific literature etc. Also, a review of methods for route choice in applications other than evacuation modelling was performed to get at a detailed overview of what this report was analysing. A set of initial literature material was initially given by the supervisor, and in a later stage information for the literature study was found through searching *Google Scholar*, which searches the web for scholarly literature and *LUBsearch*, provided by Lund University. Through *LUBsearch* literature was found on other search engines for scientific literature, for example *ScienceDirect*. The literature study focused on finding the factors that affect the chosen evacuation route (Part 1). It also focused on finding algorithms for route choice that might have been of interest (Part 2). Finally, a last literature search was conducted to find information about existing testing for verification on route choice (Part 3). Table 1 makes a summary of the keywords used in the different parts of the literature study.

Table 1. Keywords used in the literature study.

Part 1	**Part 2**	**Part 3**
Route choice	Route choice	Verification
Human behaviour	Path finding	International maritime organization
Evacuation	Evacuation	Test
Pedestrian	Pedestrian	Route choice
Behaviour affecting	Algorithms	Path finding
	Shortest path	
	Quickest path	
	A*	
	Dijkstra	
	Path finding in games	
	Artificial intelligence	

The literature taken into consideration was reviewed and selected papers were chosen based on their content concerning human behaviour during evacuation, algorithms or methods for route choice and verification test of human behaviour or route choice. The content's relevance was decided mostly by reading the abstract of the literature source.

After the literature study was completed, the variables affecting route choice algorithms used in fire evacuation models and possible alternative route choice algorithms was classified and summarized to give a good overview of the existing state-of-art.

The next step was to analyse the gathered data to be able to draw meaningful conclusions about the route choice methods strengths, limitations and so forth. The algorithms and methods for route choice in the most used models was discussed.

After the reviewing of algorithms used for route choice was completed, new verification tests for route choice were developed and suggested. This was done for the factors that are today implemented in evacuation models.

Finally, conclusions were drawn and a discussion made about the state-of-art of route choice modelling algorithms and new verification tests that can be used to evaluate their implementation in fire evacuation models.

1.3 Delimitations and limitations

This report focuses mainly on the factors that affect evacuation route choice and the different algorithms available for route choice and path finding. The analysis of additional sub-models in evacuation simulation programs are out of the scope of this thesis. Furthermore, the time is not sufficient to examine each algorithm and parameter in depth, so this evaluation will only cover the basics of each algorithm. A validation of existing evacuation sub-models for route choice is out of the scope of the thesis since limited data on decision making when choosing a route for evacuation is available (Ronchi, Kuligowski, Reneke, Peacock, & Nilsson, 2013). This thesis is however a first step towards validated algorithms since it lists which factors that need to be included in route choice modelling and subsequent future validation tests.

2. CATEGORIZATION

To get a comprehensive overview of the different algorithms and methods, a categorization has been made. Different categorizations are used in the literature for the algorithms and methods to simulate route choice and human behaviour in evacuation scenarios. Nevertheless, none of them perfectly suits the purposes of the thesis. However, a short review of existing categorizations follows in this section, before the chosen categorization is presented.

In the review by Kuligowski (2016), a categorization is made of how the occupants chose the exit route, which are as follows:

- *Fastest/optimal route*: the route that takes the least time to travel.
- *Shortest route*: the route that has the least distance from start to exit.
- *User defined route*: the route decided by the user.
- *Conditional route*: a route decided by the conditions in the building, for example the fire conditions or the behaviour of other occupants. This route choice alternative usually has a function that allows the occupant to change its route choice if the conditions demands it.

In a paper by Prato (2009) another categorization is made, which is as follows:

- *Deterministic shortest path-based methods*: this is described as the largest group for methods for path finding. The algorithms sorted under this category aims at finding the shortest path.
- *Stochastic shortest path-based methods*: as the name suggest, this also aims at finding the shortest path. But with this method the route is chosen from a group of routes with different probability distributions.
- *Constrained enumeration methods*: this is describes as using the behaviour in occupants and assumes that the occupants chose route depending on individual behaviour rather than the shortest or the quickest route.
- *Probabilistic methods*: this method assigns each route with a probability determined by different characteristics of the model.

With these categorizations in mind the chosen categorization for the algorithms in the present thesis is as follows:

- *Shortest path*
- *Quickest path*
- *Conditional path*

The three categories are further divided into sub-categories whether they are deterministic or probabilistic.

2.1 Description of each category

This section describes the chosen categorization and explains what types of algorithms that are included in each category.

2.1.1 Shortest path

The algorithms sorted under this category are algorithms that search for the shortest route in the chosen environment. The way they find the shortest path may be different, and they might

not always find that path, but the common denominator in this category is that the algorithms aims at finding the shortest route from one point to another.

2.1.2 Quickest path

The algorithms sorted under this category are algorithms that search for the route which takes the least amount of time to travel. As in the shortest path category, the algorithms might not find the path that actually takes the least amount of time to travel. But the algorithms under this category are algorithms that aims at finding the route which takes the least amount of time to travel from one point to another.

2.1.3 Conditional path

The algorithms under this category are algorithms that do not only aim at finding the shortest or the quickest route. In contrast, these algorithms make use of different factors or variables that affects the path which is chosen by the occupant. Variables or factors that might affect the route choice are summarized in Table 2 and are all mentioned in the introduction section of this thesis. These variables have been found in literature and are used in existing simulation models but are also variables that has been found to affect the route choice in real life evacuations.

Table 2. Factors that can affect the route choice in evacuation situations.

Human behaviour	**Fire conditions**	**Building construction**
Social influence/Other occupant's behaviour	Smoke	Wayfinding installations/Signage (static or dynamic)
Affiliation	Temperature	Exit block/obstacles
Visual perception	Radiation	Elevator use
Queuing time		Available egress capacity
Walking time		Alarm type
Group interactions/ Herding/ Family interaction		
Leadership		
Helping other occupants		
Investigate fire		
Alarm awareness		
Staff giving directions		

Following is a short definition of each factor.

Social influence/Other occupant's behaviour: This refers to how one group or person affects others in the same area when choosing exit route. More specifically it means that if one person or group decides to move towards a certain exit, other might feel the need to do the same (Deutsch & Gerard, 1955; Kinateder, 2013; Nilsson & Johansson, 2009).

Affiliation: This refers to how familiar the occupant is with the building and therefore might be affiliated with a certain exit (Sime, 1985). Familiarity with an exit will in this thesis mean that the occupant is affiliated to that exit.

Visual perception: This factor refers to the exploration of the space used to find the way based on what is seen (Ronchi & Nilsson, 2016; Moussaïd, Helbing, & Theraulaz, 2011). What is seen depends on the line of sight which is described as a straight line that an observer has an unobstructed image along (Oxford Dictionaries, 2016). In this thesis, the line of sight might be blocked by for example smoke, obstacles or other occupants.

Queuing time: This describes for how long time the occupants must wait in line at an exit or a certain route (Korhonen, 2015; Ehtamo, Heliövaara, Hostikka, & Korhonen, 2008).

Walking time: This describes how long time it takes to walk from the current location to the target exit (Korhonen, 2015).

Group interactions/ Family interaction/Herding: This refers to when groups stick together (herding) and because of that evacuate together, family being a kind of group (Korhonen, 2015; Georgoudas, Sirakoulis, & Andreadis, 2008).

Leadership: Leadership is when a group has a leader that makes the decisions while the others in the group follow him or her (Korhonen, 2015).

Helping other occupants: In this thesis, this is defined as when an occupant might take an alternative route or wait at some location to help another occupant to evacuate (Drury, o.a., 2009).

Investigate fire: This is when an occupant decides to control the severity of the fire.

Alarm awareness: This is how well the occupants understand that the alarm has gone off (Thompson, 1994).

Staff giving directions: This is when the building has personnel that is familiar with the environment and can give directions of where to go and what to do (Thompson, 1994)**.**

Smoke: This is simply the smoke produced by the fire.

Temperature: This is the elevated temperature caused by the fire.

Radiation: This is energy transferred by electromagnetic waves (Cutnell & Johnson, 2013). In this thesis, it is assumed to be a product of the fire, causing objects or humans to experience raised heat.

Wayfinding installations/Signage: This is when signage has been put up to help with evacuation and how that affects the occupant seeing those signs (Thompson, 1994; Chu, Parigi, Law, & Latombe, 2015).

Exit block/Obstacles: This is when the building has obstacles blocking the exit (Korhonen, 2015).

Elevator use: This is when there is an elevator in the building that might be used by the occupants to evacuate (Kinsey, 2011).

Available egress capacity: Egress is defined as the action of leaving a place (Oxford Dictionaries, 2016). In this thesis, the available egress capacity refers to how large number of people that can leave the building at the same time.

Alarm type: This describes if the alarm is auditory, visible, a spoken message or so forth (Thompson, 1994).

3. ALGORITHMS AND METHODS

This chapter presents the algorithms and methods for path finding and route choice that have been reviewed. The description of each algorithm and method is quite brief, but as stated earlier in the report this thesis focuses mainly on the general outline of the algorithms and methods and will not go into detail of their implementation. Furthermore, additional information about the strengths and limitations of the algorithms and is presented in the following chapter. Detailed information about some of the algorithms and methods have been more difficult to be found, thus they are presented only at a general level using a brief explanation of their main assumptions.

A note is made whether the algorithms or method uses a deterministic or probabilistic approach and if they implement factors concerning human behaviour, see Table 2. These factors are described in the introduction. It needs to be stated however, that an algorithm implementing human behaviour is not automatically better than one that do not. If it is for example not correctly implemented it can alter the result in a negative way. The inventor of each algorithm or method is also mentioned, if able to be found.

In addition, the chapter contains a section on the algorithms and methods that are used by common evacuation simulators today.

3.1 Shortest path algorithms and methods

The algorithms and methods that belong to the above category are summarized in this section. At first a sub-chapter for algorithms searching for the shortest path is presented, followed by a sub-chapter with a few methods for finding the shortest path.

3.1.1 Algorithms

This sub-chapter describes the algorithms used to find the shortest route. The dominating algorithms in this category is the A* algorithm, which will be presented at first, and a lot of algorithms derives from it. But before getting in to the A* algorithm and its successors, a summary will be done of the Dijkstra algorithm on which the A* algorithms is based upon.

Dijkstra's algorithm
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: E. Dijkstra (Misa, 2010).

In 1959 a Dutch computer scientist called Edsger Dijkstra presented an algorithm for finding the shortest path from one point to another. The path starts at a node and the algorithm aims at finding the closest node from the start node, and then the closest node to the second node and so forth in a weighted graph (the representation of connectivity with edges and nodes). The edges weight represents a cost when moving between the nodes, and the path that has the least cost from the start node to the goal node is then the shortest (Stout, 1997; Anguelov, 2011; Cui & Shi, 2011). Every node gets assigned a distance value, were the start node has a value of 0, and every other node have a value of infinity. The nodes get divided into visited and unvisited sets, where initially the start node is visited, and the rest is unvisited. The distance from the current node (initially the start node) to all the unvisited neighbouring nodes gets calculated. The distance value for the current node gets compared to the distance values calculated for the neighbouring nodes. If one or more of the newly calculated distances (the distance to the current node plus the distance from the current node to the neighbouring

nodes) is less than that of the current, the distance that is the smallest will replace the old distance. The current node will then be regarded as "visited". The node that will give the smallest distance value will then be the new current node, and the procedure will be repeated until the goal node is reached (Li & Wu, 2016). A node that then belongs to the visited set, will not be visited again (Anguelov, 2011).

In Dijkstra's algorithm, the cost for every possible path between the nodes is calculated on top of the investigation of the shortest path. Because of this, the shortest route is guaranteed to be found. Since the algorithm is computationally expensive there have been several attempts to modify this algorithm in order to optimize (Stout, 1997; Anguelov, 2011; Cui & Shi, 2011).

A* algorithm
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: P. Hart, N. Nilsson and B. Raphael (Hart, Nilsson, & Raphael, 1968).

The A* algorithm was presented in 1968 as an optimization of Dijkstra's algorithm (Anguelov, 2011). It is a search algorithm that can be used to find the least cost path, and is probably the most used algorithm in computer games (Cui & Shi, 2011; Botea, Müller, & Schaeffer, 2004). The cost of a path can be defined as almost anything. It can for example be how dangerous a path is, its distance or time (Stout, 1997). It is for example used in common computer games such as Age of Empires (Microsoft Game Studios, 2016) and Civilization V (Firaxis Games, 2016). If a path exists, then the A* algorithm is guaranteed to find a route from the start to the goal. Also, the A* algorithm will expand less nodes than any other search using this heuristic function (Stout, 1997). Unlike Dijkstra's algorithm it uses a heuristic approach, meaning that only the paths that is estimated as more promising are examined, resulting in less examined nodes (Cui & Shi, 2011). Each node gets ranked based on the best route through the node with the formula presented in equation (1).

$$f(n) = g(n) + h(n) \tag{1}$$

Where

f(n) is the cost of node n

g(n) is the cost of arriving at node n

h(n) is the cost to the goal from n made by the heuristic estimate.

Many efforts have been made to optimize the A* algorithm and several derivatives of the A* algorithm exists, aiming at for example improving the heuristic search, the map representation or minimize the computationally cost (Cui & Shi, 2011; Botea, Müller, & Schaeffer, 2004).

The A* algorithm is often used in navigation meshes (NavMesh) for artificial intelligence in 3D worlds (Cui & Shi, 2012). A NavMesh is a representation of the walkable floor in a 3D environment using convex polygons. An agent starts in one polygon with the goal being placed in another polygon. The first thing that is done is to decide the next polygon to move to. This is done until the agent is in the same polygon as the goal, then the agent can move to the goal in a straight line. With this approach a near optimal path is guaranteed to be found searching less data and simulating pathfinding behaviour better than with a waypoint graph (Cui & Shi, 2011; Anguelov, 2011; Botea, Müller, & Schaeffer, 2004; Tozour, 2002).

Anytime repair A*
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: M. Likhachev, G. Gordon and S. Thrun (Likhachev, Gordon, & Thrun, 2003).

The anytime repair A* algorithm will give a solution for the shortest path by starting with finding a sub-optimal path and improving that path over time using a heuristic search, much like the A* algorithm. This way the algorithm will always be able to return a solution regardless of the time used searching. If there is a time constraint on the search for an optimal path, a sub-optimal path is better than no path at all. But if given sufficient time searching, an optimal path will be found. The anytime search A* algorithm is meant to be used in complex static environments, due to the lack of possibility to deal with changes in the environment (Likhachev, Gordon, & Thrun, 2003; Anguelov, 2011).

Dynamic A* Lite/D* Lite
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: S. Koenig, M. Likhachev (Koenig & Likhachev, 2002).

This is a version of the A* algorithm that makes it possible to calculate the shortest path for moving occupants and has been successfully used in unknown environments and in robot navigation. The problem with moving occupants is that the start node changes as the agent move, making the shortest route change simultaneously. This is solved by reversing the search direction, making the goal node the first examined node and then moving backward towards the occupant, or the start node (Koenig & Likhachev, 2002; Anguelov, 2011).

Anytime dynamic A*
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: M. Likhachev, D. Ferguson, G. Geoffm A. Stentz and S. Thrun (Likhachev, Ferguson, Gordon, Stentz, & Thrun, 2005).

This algorithm is a combination of the anytime repair A* and the dynamic A* lite algorithm. This is done to be able to use the anytime repair A* in dynamic environments. Otherwise it works in the same way as the anytime repair A* (Likhachev, Ferguson, Gordon, Stentz, & Thrun, 2005; Anguelov, 2011).

Hierarchical pathfinding A*
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: A. Botea, M. Müller and J. Schaeffer (Botea, Müller, & Schaeffer, 2004).

This version of the A* algorithm divides the environment into different hierarchically areas. When moving from start to goal the hierarchical pathfinding A* will firstly solve for the big picture regarding which way to go, and then making a more detailed plan of exactly where to go. The “big picture” and the “detailed plan” are two levels in this algorithm. But it is possible to implement more levels if necessary in larger environments. The search for the “big picture” is done at an abstract level making it faster and simpler. This algorithm has been proved to be ten times faster than a simpler A* algorithm. When adding more abstract layers to the problem the cost increases drastically (Cui & Shi, 2011).

Knowing the complete path from the beginning is not always needed. Just knowing which direction to start moving is enough. During the movement along the first path, an additional path to keep moving on will be calculated making the occupant able to change its plan when needed (Botea, Müller, & Schaeffer, 2004).

Iterative deepening A*
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: R. Korf (Korf, 1985).

This version of the A* algorithm is often used in computer games. It is not as fast as the A* algorithm, but it uses less memory (Botea, Müller, & Schaeffer, 2004). It was presented as a low memory alternative to the A* algorithm in 1985 (Korf, 1985; Anguelov, 2011). To reduce the A* algorithms space requirements the path from start node to goal node can be computed in small pieces. This is done in this algorithm by setting a maximum threshold for the cost. When this threshold is exceeded the path is cut off, and a new search starts (Korf, 1985; Cui & Shi, 2011; Anguelov, 2011; Stout, 1997). The reason for the low memory is that the algorithm does not keep track of already examined nodes. But since it does not save any data there is a lot of recalculations that must be made, making it slower (Anguelov, 2011).

The fringe search algorithm
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: Y. Björnsson, M. Enzenberger, R. Holte and J. Schaeffer (Björnsson, Enzenberger, Holte, & Schaeffer, 2005).

This is an optimization of the iterative deepening A* algorithm, trying to deal with the recalculation made in the mentioned algorithm. It does so by creating a "fringe" at the end of a search, making it possible to restart the search at the same place. Other than that, the search is done in the same way as the iterative deepening A* algorithm does (Björnsson, Enzenberger, Holte, & Schaeffer, 2005; Anguelov, 2011).

Lifelong planning A*
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: S. Koenig, M. Likhachev and D. Furcy (Koenig, Likhachev, & Furcy, 2005).

This algorithm makes the search for the shortest path in a similar manner as the usual A* algorithm. However, following searches are meant to repair detected local variations that might have occurred due to environmental changes, making it usable in dynamic environments (Koenig, Likhachev, & Furcy, 2005; Anguelov, 2011).

Memory-bounded search A*
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: H. Bast, S. Funke and D. Matijevic (Bast, Funke, & Matijevic, 2009).

This version of the A*algorithm demands less memory. When a predefined limit for the memory is reached the calculated nodes that are of least interest are discarded. Just like with the iterative deepening A* algorithm some information is lost and some recalculations has to

be made for those nodes. There is however an algorithm called "the simplified memory-bounded algorithm" that has a solution for this. But both of these algorithms have a very high computationally cost and are not suitable for use in time constrained environments (Anguelov, 2011).

Real time A*
Type: Deterministic.
Directly implements human behaviour: Yes: visual perception.
Inventor: R. Korf (Korf, 1990).

The real time A* algorithm was developed in 1990 and are a real time heuristic algorithm. It was developed to solve larger search problems that at the time were not solvable due to the memory requirements. This algorithm searches around the occupants' node to find the next move that is best, using a heuristic search. The algorithm only searches a limited distance, referred to as the search horizon. This means that the occupant will choose the best move as far as it can see. Doing this might result in the occupant moving towards a non-optimal path or towards a dead end. This problem is solved by the algorithm by having a heuristic function that learns with experience. The algorithm is fast and uses little memory since no path is generated. The occupant decides what the next move is first after completing the previous move (Anguelov, 2011).

The fact that it chooses the route depending on how far it can see, makes it fall under the category of making use of human behaviour. If one recalls Table 2, the factor "visual perception" is mentioned, which is exactly what this is.

Biased random walk
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: E. Frejinger, M. Bierlaire (Frejinger & Bierlaire, 2007).

This is an algorithm that uses a stochastic path generation approach using sub-paths. The sub-path is a sequence of links that has a probability based on its distance to the goal. The sub-paths add together in creating the final path depending on the sub-paths probabilities (Frejinger & Bierlaire, 2007; Prato, 2009).

3.1.2 Methods

This sub-chapter describes the methods used to find the shortest route.

Surplus distance method
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: P. Thompson (Thompson, 1994).

This method is used in the first version of Simulex, developed by Thompson (1994) to calculate the agent's route with a distance map. Nothing suggests that it is not implemented in the model anymore, although it is not explicitly stated in literature. With this method, each possible route's distance to an exit is calculated and compared to the optimal distance to the exit from the agent's location, where the optimal distance is the shortest possible route with straight lines. The distance to intermediate targets along the route is added to a total distance. The optimal distance is then subtracted from the total distance to creating a surplus distance.

The route closest to zero (0) meter in surplus distance is the chosen one. The calculation is described in equation (2) (Thompson, 1994).

$$Surplus\ distance = Total\ inermediate\ distances - Optimal\ distance \qquad (2)$$

The agents aim at moving along the shortest path in a calculated angle, but algorithms for overtaking is also used. This means that if an agents path gets obstructed by another agent it can move along a path around that other agent (Thompson, 1994; Thompson, Wu, & Marchant, 1997; Thompson & Marchant, 1994).

Fuzzy traffic assignment model
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: V. Henn (Henn, 1998).

This model is proposed as a solution for traffic assignments within dynamic environments. The idea behind this model is that drivers do not know every event that might or will happen along the chosen route, the exact length of the route or the speed they can travel in. In this model a route choice is made at every intersection instead of choosing the path from the current location to the destination. The chosen exits at the intersection is based on probability cost by fuzzy sets (Henn, 1998). "Fuzzy sets" originates from a paper by L. Zadeh (1965). It describes how true or false a statement is by assigning a number between 0 to 1 to it, where 0 is absolute false and 1 is absolute true (Zadeh, 1965; Novák, Perfilieva, & Močkoř, 1999). The fuzzy sets can represent how true or false it is that a possible route is long, or has disturbing traffic conditions. The driver chooses the route with the least cost depending on the driver's impression off the length to the final destination (Henn, 1998).

3.1.3 Summary

In Figure 1 an overview of the shortest path algorithms and methods described in this chapter is made. As seen most the algorithms is derived from the A* and the Dijkstra's algorithm, which makes the majority in this category deterministic. However, there are still some algorithms that aims at finding the shortest path that are not a variant of the A* algorithm.

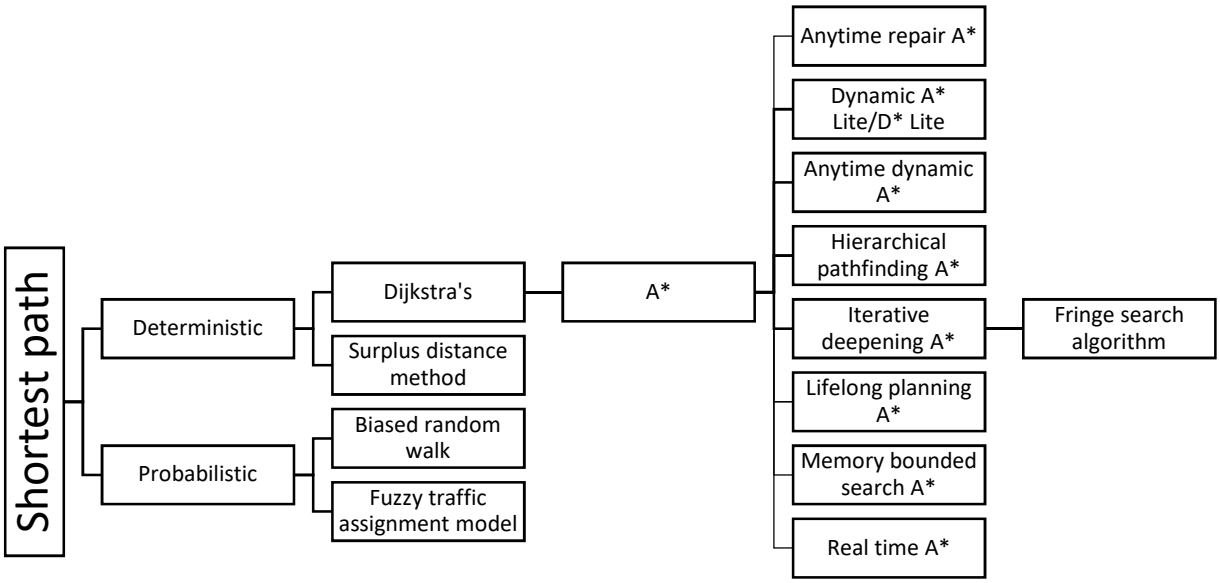

Figure 1. Overview of the shortest path algorithms and methods.

3.2 Quickest path algorithms and methods

This section makes a summary of algorithms and methods that attempts to find the quickest path. At first a sub-chapter for algorithms searching for the quickest path is presented, followed by a sub-chapter with methods for finding the quickest path.

3.2.1 Algorithms

This sub-chapter describes the algorithms used to find the quickest route.

A* algorithm
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: P. Hart, N. Nilsson and B. Raphael (Hart, Nilsson, & Raphael, 1968).

As mentioned in the earlier section about shortest path algorithms and methods, the A* algorithm can be used to find the least cost path, where the cost can represent many variables. The cost of time being one of those (Stout, 1997). For more information about the A* algorithm, read the section about shortest path algorithms and methods.

Advanced capacitated network flow transshipment algorithm
Type: Deterministic.
Directly implements human behaviour: No.
Inventor: Unknown.

This algorithm uses a network model to decide the optimal evacuation strategy trying to minimize evacuation time, solving transshipment problems (Santos & Aguirre, 2004; Kisko, Francis, & Nobel, 1998). A transshipment problem is a transportation problem where transportation is done to intermediate goals with the possibility to change transportation method at the intermediate goals (Winston, 2004). This algorithm is used by the evacuation simulator EVACNET, version 4 (Kisko, Francis, & Nobel, 1998).

3.2.2 Methods

This sub-chapter describes the methods used to find the quickest route. The logit models dominate this section in the same way A* algorithms dominate the shortest path category. Logit models are discrete choice models that calculates probabilities for different outcomes and have been implemented in traffic assignment models (Ben-Akiva & Bierlaire, 2003).

Multinomial logit
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: D. McFadden (McFadden, 1976).

The multinomial logit model developed by McFadden (1976) is an economic version of Luce's choice axiom. Luce's choice axiom is a theory of individual choice behaviour where a selection has independence from irrelevant alternatives (IIA). This means that the probability of selecting one alternative over another is not affected by other alternatives that are or are not available (Luce, 1959). The logit model can predict outcomes of different alternatives (Koppelman & Bhat, 2006). It can for example be when travelling by car and whether you want to take the highway, or travel on smaller roads. It can be whether you want to use the bus, or ride a bicycle. The choice depends on different independent variables. In the latter example, the variables can be if there is raining outside, or whether you have money for the

bus. The aim for this model is to be able to correlate the independent variables with the result, so that a future result or choice can be predicted in beforehand. What the logit model does, is that it transforms the number of a factor into a probability between 0 and 1. For example if it is seen that people over a certain age are more likely to use the bus rather than the bicycle, it transforms the factor of age into a probability between 0 to 1 (Koppelman & Bhat, 2006). The exact mathematics of how the transformation is done is however not within the scope of this thesis. This basic function is the same for all the logit models, even though each new logit model has some minor alternations, which will be presented below.

C-logit
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: E. Cascetta, A. Nuzzolo, F. Russo and A. Vitetta (Cascetta, Nuzzolo, Russo, & Vitetta, 1996).

This model was presented as a solution to a problem with the multinomial logit model (Ben-Akiva & Bierlaire, 1999). The multinomial logit model would produce unrealistic probabilities for choices for paths that shared links (part of route) with other paths. The c-logit model solved this by adding a "commonality factor" that measures how similar different routes are (Ben-Akiva & Bierlaire, 1999; Cascetta, Nuzzolo, Russo, & Vitetta, 1996; Prato, 2009). However, the commonality factor only captures parts of the similarity of other routes (Prato, 2009). The chosen path depends on the cost in travel time (Cascetta, Nuzzolo, Russo, & Vitetta, 1996).

Nested logit
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: M. Ben-Akiva (Ben-Akiva, 1973).

The IIA property of the multinomial logit is a limitation because it assumes that all alternatives are equal. This is not the case with the nested logit model. In this model the alternatives that are similar gets grouped (nested) together. A group of nested alternatives is more similar to each other than those in another nest. The alternatives in a nest are more competitive against each other than to other alternatives in other nests (Prato, 2009). It can be compared to politics where one party are more likely to mainly "steals" voters from a similar party, then from a party with totally different beliefs. However, the alternatives in different nest are not allowed to interact with each other resulting in missed scenarios (Bierlaire, 2001).

Cross-nested logit
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: P. Vovsha (Vovsha, 1997).

Just like the nested logit model this model group the routes together in different nests. However, in this model the alternatives can belong to several nests (Ben-Akiva & Bierlaire, 2003). This model is more computational demanding because it is complex and requires longer calculation times (Prato, 2009). It is called the "generalized nested logit" in an analysis by Wen and Koppelman (2001).

Paired combinatorial logit
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: C. Chu (Chu, 1989).

This model has the same basic intent as the cross-nested logit model. It makes it possible for different alternative in different nests to be competitive against each other, which leads to a larger combination of probabilities (Koppelman & Wen, The Paired Combinatorial Logit Model: Propertiesm Estimation and Application, 2000).

Path-size logit
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: M. Ben-Akiva and M. Bierlaire (Ben-Akiva & Bierlaire, 1999).

This model is an attempt to fix the problem with paths that share links with other paths in the multinomial logit model. The c-logit introduced a "commonality factor" while this include a correction term for the path size indicating the fraction of that path compared to a complete path (Prato, 2009).

Locally quickest algorithm
Type: Deterministic.
Directly implements human behaviour: Yes: walking time, queuing time, elevators.
Inventor: C. Thornton, R. O'Konski, B. Klein, B. Hardeman, D. Swenson (Thornton, O'Konski, Klein, Hardeman, & Swenson, 2012).

This algorithm uses local information about the occupant's room, and then global information about the whole building. The occupant in the model is assumed to have knowledge of every door and their queues in the room they are in. Also, the algorithm assumes that the occupant knows the distance from the doors to the goal destination. The information about the doors are then used to give each door a cost, and the door with the lowest cost is then chosen as the one to use. After choosing a local target, or door, the path to that target is calculated with the A* algorithm. The door, or local target, gets a cost depending on the following variables: (Thunderhead, 2016; Thornton, O'Konski, Klein, Hardeman, & Swenson, 2012).

- Travel time of the room
- Queue time in the room
- Travel time from door to goal destination
- Distance travelled in the room

The occupants also have some factors that affect the door choice which can be used:

- Cost factor depending on the travel time of the room
- Cost factor depending on the queue time in the room
- Cost factor depending on the travel time from door to goal destination
- Preference of door
- Cost factor depending on distance travelled in a room

Since this algorithm also uses the A* algorithm to calculate the path between targets, one could argue that it only uses the A* algorithm for path finding. While this is true for the

transportation between targets, the author feels as if the path to a higher degree is decided by the local quickest algorithm, and is therefore categorized as a different algorithm.

The occupants accounts for queuing time and walking time which are factors for human behaviour included in Table 2 in the categorization section of this paper. Furthermore, the occupants can be assigned behavioural goals, meaning that they must either wait at a place for a specified event to happen, or move through certain targets (rooms for example), before they can move towards an exit. The sub-model allows representing elevator usage too, which is mentioned in Table 2 as human behaviour affecting route choice (Thunderhead, 2016).

The inventor of this algorithm is the developers of the evacuation model Pathfinder (Thornton, O'Konski, Klein, Hardeman, & Swenson, 2012) and it is used in Pathfinder 2016 (Thunderhead, 2016).

STEPS method
Type: Probabilistic.
Directly implements human behaviour: Yes: queuing time, herding, exit block/obstacles.
Inventor: Mott MacDonald Group Limited (Mott MacDonald Group Limited, 2016).

The name of the method or algorithm used by STEPS have not been found, and is therefore called the "STEPS method" in this thesis and is referring to the method used in STEPS version 4.0. It uses a microscopic model to simulate the movement of agents in a fine-network space, i.e. it considers individual movement. The agents get assigned following characteristics (Waterson & Pellissier, 2010):

- Unimpeded walking speed
- Awareness of the environment
- Patience when queuing
- Association to a group

These characteristics are normally assigned to a sub-group, but can be assigned to individuals too (Waterson & Pellissier, 2010). The movement of the agents is driven by the desire to move with their unimpeded speed to a target in the shortest time, without colliding with obstacles or other agents. The obstacles can be entered manually by the user (Kuligowski, 2016). The decisions of route choice by the agents are randomised. Since STEPS can be used to simulate movement of agents in other scenarios than evacuation, the agents can move around in the building or environment with the aim to fulfil different tasks at different locations (Waterson & Pellissier, 2010).

3.2.3 Summary

In Figure 2 an overview of the quickest path algorithms and methods described in this chapter is made. As with the shortest path category, one family of algorithms or methods dominates this category. It is the logit family that since the development of the multinomial logit have been further developed into several modified versions aiming at fixing problem with the multinomial logit model. Two of the algorithms in this category are used in evacuation models today, and those are the locally quickest algorithm and the STEPS method.

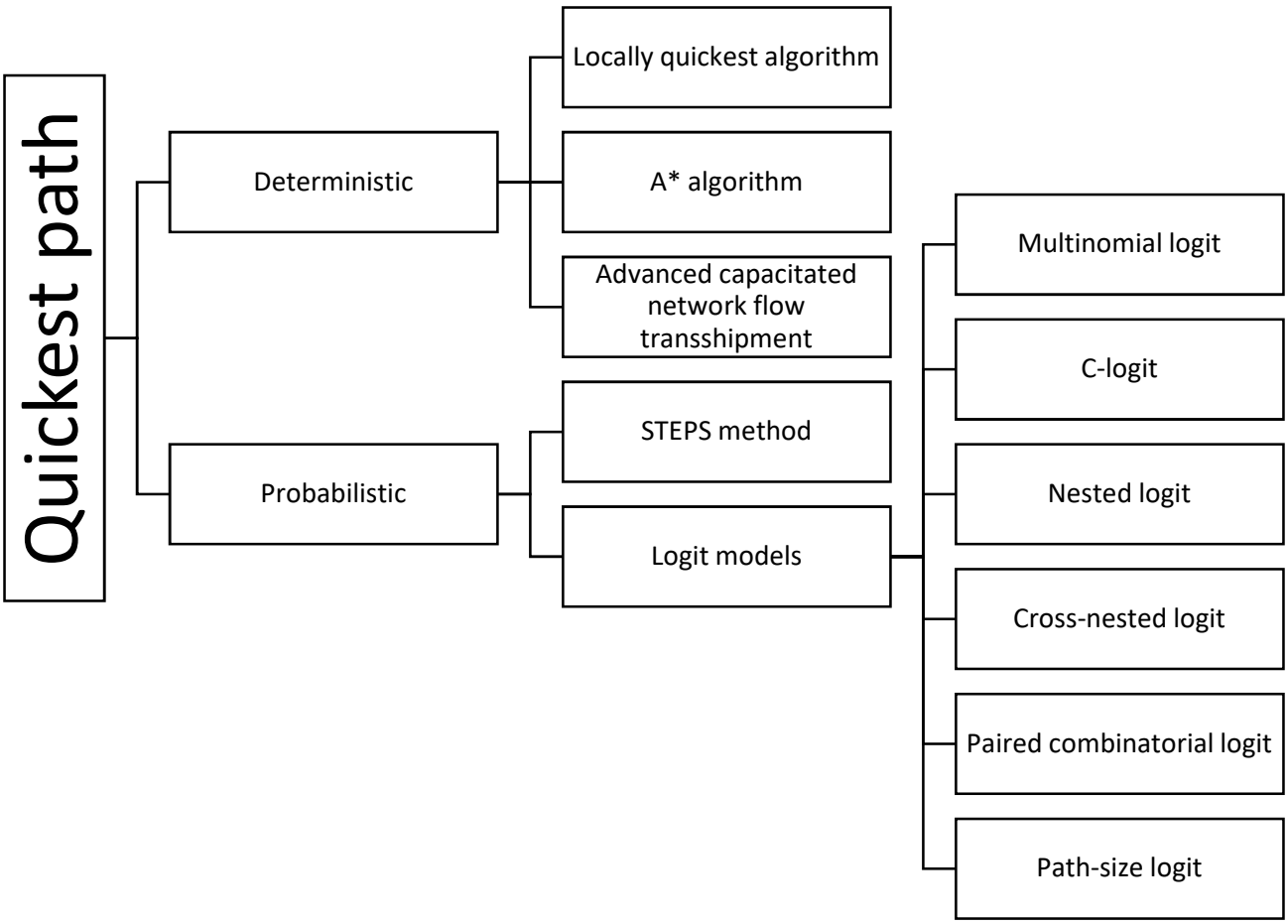

Figure 2. Overview of the quickest path algorithms and methods.

3.3 Conditional path algorithms and methods

This sections describes the algorithms that include a series of complex factors for route choice. At first a sub-chapter for algorithms searching for a conditional path is presented, followed by a sub-chapter with methods for finding a conditional path.

3.3.1 Algorithms

This sub-chapter describes the algorithms used to find a conditional path.

Multiobjective optimization genetic algorithm
Type: Deterministic.
Directly implements human behaviour: Yes: walking time, queuing time.
Inventor: Q. Li, Z. Fang, Q. Li and X. Zong (Li, Fang, Li, & Zong, 2010).

This model assigns the evacuation route according to three objectives; minimize the total evacuation time for the building, minimize the total distance travelled and to minimize the queuing during the evacuation. The algorithms can minimize these three objectives at the same time, even though the different objectives might conflict with each other. It is up to the user to set the importance of each objective to the occupants (Li, Fang, Li, & Zong, 2010).

The reason for it having "Yes" as the answer to if it implements human behaviour is that the occupants tries to minimize travel (walking) and queuing time. This is, as seen in Table 2 in the categorization section of this paper, factors affecting route choice amongst evacuating people.

3.3.2 Methods

This sub-chapter describes the methods used to find a conditional path.

Adaptive decision making models
Type: Deterministic.
Directly implements human behaviour: Yes: affiliation, visual perception, queuing time, smoke, temperature.
Inventor: Fire Safety Engineering Group of the University of Greenwich (Fire Safety Engineering Group, 2016).

In the model the occupants can change their evacuation path a few times. The decision of evacuation path depends on the familiarity, visibility and the queuing time (Gwynne, Galea, Lawrence, Owen, & Filippidis, 2000). However, the exact formulas and factors of the model have not been published being that this model is developed for a commercial software (EXODUS) (Fire Safety Engineering Group, 2016).

In a paper by Gwynne et al. (2004) the model used is described more in detail. It says for example that it implements hazards from fire, like toxic gases and heat. Furthermore, each occupant gets assigned certain (over 20) attributes. The ones mentioned are: age, weight, gender, agility, patience, drive, travelled distance, and concentration of toxic gases. The occupant's behaviour can be either determined by the user or randomly selected. The geometry representation is made up by a fine network of nodes connected with arcs which the occupants move along (Gwynne, Galea, Owen, Lawrence, & Filippidis, 2004).

Game theoretic reaction function model
Type: Probabilistic.
Directly implements human behaviour: Yes: herding/family interaction, smoke, affiliation, visual perception, walking time, queuing time, exit block/obstacles.
Inventor: H. Ehtamo, S. Heliövaara, S. Hostikka and T Korhonen (Ehtamo, Heliövaara, Hostikka, & Korhonen, 2008).

In this model for exit choice there are factors that affect the human behaviour of occupants. There is a sub-algorithm that describes attraction or repulsion between occupants. This makes it possible to simulate herding, family interaction or a pair of fire fighters that enters the building together. The model is implemented in FDS 2.5.0 (Korhonen, 2015).

Occupants move towards the exit using vector fields from the flow solver in the Fire Dynamics Simulator FDS (a flow of a two-dimensional incompressible fluid that flows towards the exit) (Korhonen, 2015).This way the occupants often take a route close to the shortest. The vector fields will lead the occupants to the wider routes rather than the narrow ones (Korhonen, 2015).

When choosing exits, the algorithm uses the density of the smoke to categorize the exits depending of the visibility. Obstacles can be placed in the geometry by the user to block walkable areas. Game theoretic reaction functions and best response dynamic are also used when choosing exit route (Korhonen, 2015). This model is based on a concept used in the computer game world. The occupants observe the building and what other occupants are doing when choosing exit route (Korhonen, 2015). When using a game theoretic best-response dynamics function the agents try to maximize their individual payoff. The payoff depends not only on the agent's own strategy, but also on other agent's strategy for evacuating (Ehtamo, Heliövaara, Korhonen, & Hostikka, 2010; Lo, Huang, Wang, & Yuen, 2006). The chosen exit route is the one that is estimated to be the fastest one. The calculation of evacuation time is done by adding the walking time with the queuing time. The walking time depends on the distance to the chosen exit and the walking speed. The queuing time depends on actions and locations of the other occupants. If the occupants find an exit route that is clearly better than the chosen one, the occupants will change their course. To calculate this action a parameter is subtracted from the earlier estimated evacuation time (Ehtamo, Heliövaara, Hostikka, & Korhonen, 2008; Korhonen, 2015).

Factors that come in to play when choosing exit route are the fire conditions, the familiarity and visibility of the exits. This is accounted for by adding limitations to the evacuation time minimization problem. Familiarity of exits are decided by the user, or randomly by the model. The visibility of the exits depends on smoke and obstacles, but not on other occupants. Data regarding the fire in FDS determines if there are any disturbing conditions, like toxicity or smoke, on the exit route. If the disturbing conditions are lethal (high toxicity), the exit can not be used (Korhonen, 2015).

The model has four different types of agents as described below (Korhonen, 2015):

- *Conservative type:* this type can be seen as a costumer in a shopping mall. They know the main exits and rather not use special emergency exits. They are also assumed to have higher stress and are likely to change exit route.
- *Active type:* this type works almost like the conservative type, but instead they activly search their environment to find the quickest route and prefer all visible exit regardless of familarity.
- *Herding type:* this type is unfamiliar with the environment and only uses familiar exits. It represent lost agents that look at what other agents do and follows. However, if a familiar route is availabel, it overwrites the herding behaviour.
- *Follower type:* this type choses the same exits as the conservative type. But they also look at where the nearest agents are heading and include that exit in their own list of possible exits.

The algorithm for selection of exit route first divides the exits into groups, as seen in Table 3 below. Then an occupant is assigned the most preferred nonempty group by minimizing the evacuation time (Korhonen, 2015).

Table 3. "Preference order used in the exit selection algorithm. The last two rows have no preference. This is because the agents are unaware of the exits that are unfamiliar and invisible and, thus, can not select these exits. The last column shows the colours used in Smokeview to show the status of the agents, if this colouring option is chosen by the user." (Korhonen, 2015)

Preference	Visible	Familiar	Disturbing conditions	Colour
1	Yes	Yes	No	Black
2	No	Yes	No	Yellow
3	Yes	No	No	Blue
4	Yes	Yes	Yes	Red
5	No	Yes	Yes	Green
6	Yes	No	Yes	Magenta
No preference	No	No	No	Cyan
No preference	No	No	Yes	Cyan

As mentioned earlier the smoke affects the exit selection, where the exit is free to use if the smoke gas concentrations is below a certain value. The user can give the threshold for the visibility of the door. The door will be visible and free to use as an exit if the visibility is larger than half the distance to the door (Korhonen, 2015).

Labelling approach
Type: Deterministic.
Directly implements human behaviour: Yes: queuing time, walking time, affiliation.
Inventor: M. Ben-Akiva, M. Bergman, J. Daly and R. Ramaswamy (Ben-Akiva, Bergman, Daly, & Ramaswamy, 1984).

This approach is used in traffic simulations and makes use of human behaviour in case travellers have different objectives. While some want to minimize travel time, other do not want to travel through bigger freeways with lot of traffic. And while some travellers feel more comfortable driving through familiar areas, others like to look at new sights and do not care about time. These factors collectively create the individual optimal route and are labelled accordingly (Ben-Akiva, Bergman, Daly, & Ramaswamy, 1984; Ramming, 2002). This

approach only partially reproduces the choices of actual drivers, given the definitions of the labels and the complexity of guessing the traveller's objective (Prato, 2009).

Simulation approach
Type: Probabilistic.
Directly implements human behaviour: No.
Inventor: Y. Sheffi and W. Powell (Sheffi & Powell, 1982).

This method is originated from a traffic assignment model and uses the behavioural assumption that travellers perceive the cost of a path with an error. This is modelled by using cost functions drawn from probability distributions. There have been several implementations of this method in traffic assignments. Most have used a Monte Carlo technique to assign probabilities, using truncated values and normal distributions with mean and variances that equals link travel time (Sheffi & Powell, 1982). While some have used this method to look at the shortest path, others have used it to find the quickest path. This method generates a large number of attractive paths if a suitable probability distribution has been used (Prato, 2009).

3.3.3 Summary

In Figure 3 an overview of the conditional path algorithms and methods described in this chapter is made. Only the multiobjective optimization genetic algorithms is classified as an algorithm. The rest are more of methods. Not any of the sub-categories of deterministic or probabilistic are overrepresented.

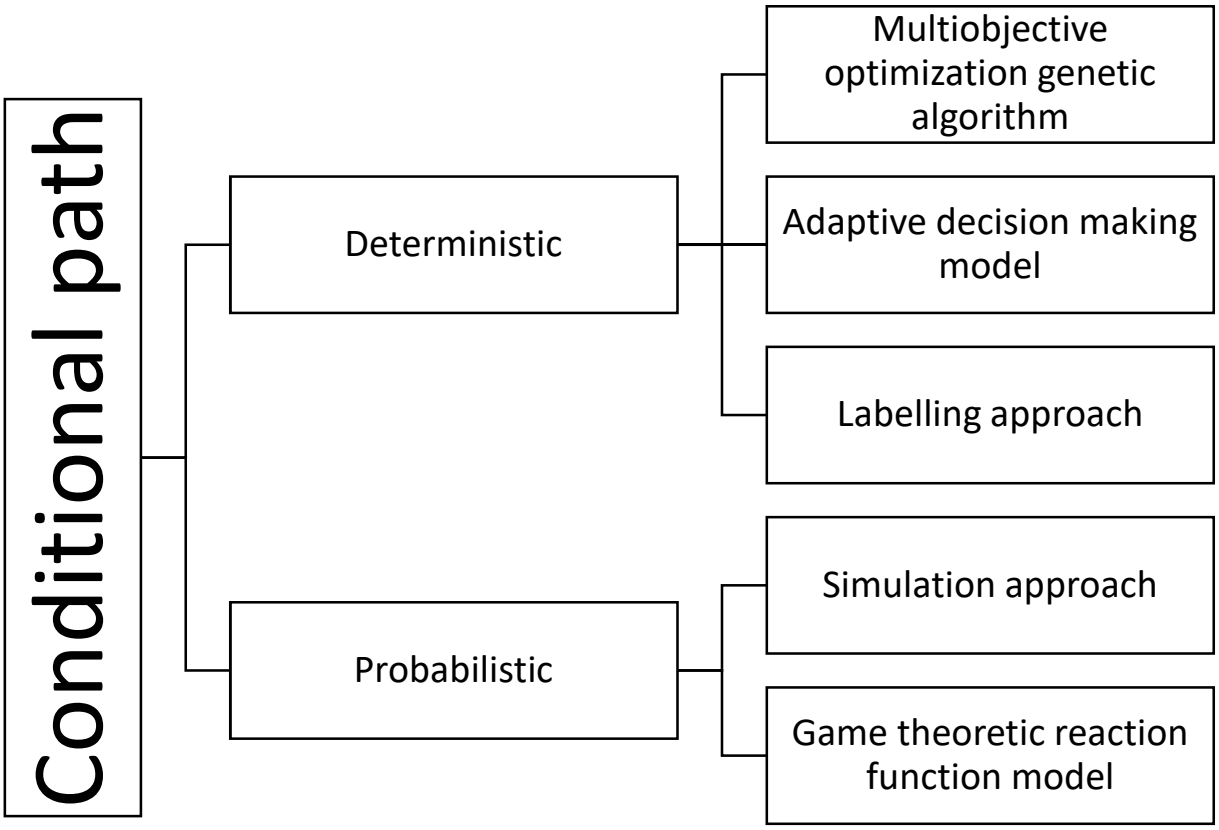

Figure 3. Overview of the conditional path algorithms and methods.

3.4 Algorithms and methods commonly used in evacuation simulation models

This section describes the algorithms and methods used in the most common models for evacuation simulation today. According to a survey on evacuation models users (Ronchi & Kinsey, 2011) the models that most users are aware of are EXODUS (Gales, et al., 2006), FDS+Evac (Korhonen, 2015), Simulex (IES, 2000), STEPS (Waterson & Pellissier, 2010) and Pathfinder (Thunderhead, 2016). A short summary of the algorithms or methods used by these models follows.

EXODUS: EXODUS 2.0 uses an "adaptive decision making model" (Gwynne, Galea, Lawrence, Owen, & Filippidis, 2000). This model allows the agents to change evacuation path and the route choice depends on familiarity, visibility and length of queues length (Gwynne, Galea, Lawrence, Owen, & Filippidis, 2000). Gwynne et al. (2004) describes the model as being able to implement hazards from fire, like toxic gases and heat. Furthermore, each occupant gets assigned certain (over 20) attributes. The ones mentioned are: age, weight, gender, agility, patience, drive, travelled distance, and concentration of toxic gases. The occupant's behaviour can be either determined by the user or randomly selected (Gwynne, Galea, Owen, Lawrence, & Filippidis, 2004).

FDS+Evac: In FDS+Evac 2.5.0, a game theoretic reaction function model is used. It has s a sub-algorithm that describes attraction or repulsion between occupants, making it possible to simulate herding and family interaction. Factors affecting route choice with this model is fire conditions, the familiarity and visibility of the exits (Korhonen, 2015).

Also used in FDS+Evac is a game theoretic reaction functions and best response dynamic (Korhonen, 2015). This means that the occupants observe the building and what other occupants are doing when choosing exit route (Korhonen, 2015).

The chosen exit route is the one that is estimated to be the fastest (Ehtamo, Heliövaara, Hostikka, & Korhonen, 2008; Korhonen, 2015).

With fire simulation in FDS it is possible to simulate dynamic obstacles. This is not the case in FDS+Evac, where the initial geometry will be used throughout the entire simulation. There is however a sub-model for door selection which can make doors unusable after user defined times (Korhonen, 2015).

Simulex: Simulex uses the surplus distance method that calculates the agent's route using a distance map. With this method, each possible route's distance to an exit is calculated and compared to the optimal distance to exit from the agent's location, where the optimal distance is the shortest possible route with straight lines (Thompson, 1994).

Algorithms for overtaking is also used. This means that if an agents path gets obstructed by another agent it can move along a path around that other agent (Thompson, 1994; Thompson & Marchant, 1994; Thompson, Wu, & Marchant, 1997).

STEPS: No specific name of the method or algorithm used by STEPS 5.3 is given, and is therefore called the "STEPS method" in this thesis. It simulates individual movement of the agents that are assigned a walking speed, awareness, patience, association and pre-movement (Waterson & Pellissier, 2010).

The agents aim at moving towards a target in the shortest time, without colliding with obstacles or other agents. They do not have knowledge of every possible exit. Instead, they move around searching an available exit (Waterson & Pellissier, 2010).

Pathfinder: Pathfinder 2016 uses a locally quickest algorithm. It uses local information about the occupants' room, and then global information about the whole building. The occupant in the model is assumed to have knowledge of the room they are in. Also, the model assumes that the occupant knows the distance from the doors to the exit. The information is used to give each door a cost, and the door with the lowest cost is then chosen. After choosing a local target, or door, the path to that target is calculated with the A* algorithm. The door, or local target, gets a cost depends on travel time, queue time, distance and preference (Thunderhead, 2016).

4. ANALYSIS

In this chapter the algorithms and methods will get analysed by their strengths and limitations, finishing with a summary of those.

4.1 Strengths and limitations

This section will present the algorithms and methods strengths and limitations one by one in a comprehensive way. When determining the strengths and limitations the author has been looking at following parameters mainly:

- *Computational cost*: if it is demanding in computer capacity.
- *Fast searches*: if the literature states that the algorithm quickly can find an exit route.
- *Capability to handle dynamic environments*: if it can find a route in dynamic environments, i.e. an environment where conditions like obstacles or fire impact changes.
- *Factors concerning human behaviour implemented:* the number of factors from Table 2 implements.
- *Other*: this is strengths or limitations that does not fit into any of the above categories.

It is worth mentioning that even though the parameter of factors for human behaviour is seen as a strength in this thesis, this might not be the case. In evacuation models that implements human behaviour, the outcome often depends on the inputs from the user (Ronchi & Nilsson, 2016). This makes it possible for users to assign unrealistic behaviour among occupants. But in this section of the thesis, it is assumed that the factors for human behaviour are implemented correctly by the user and the model, and can therefore be seen as a strength.

4.1.1 Shortest path algorithms

This sub-chapter describes the strengths and limitations of the algorithms for shortest path finding.

Dijkstra's algorithm

The strengths and limitations of the Dijkstra's algorithm found in literature is presented here and summarized in Table 4.

Table 4. Strengths and limitations of the Dijkstra's algorithm.

Computational cost	**Handles dynamic environments**
High	Information not found
Fast searches	**Factors for human behaviour**
No	0
Other	
Guaranteed to find path	

As mentioned in the earlier chapter, Dijkstra's algorithm is guaranteed to find a path, if there are one, from start node to goal node. However, due to the fact that it explores every node in the system which is unnecessary, the computationally costs get high (Cui & Shi, 2011).

A* algorithm
The strengths and limitations of the A* algorithm found in literature is presented here and summarized in Table 5.

Table 5. Strengths and limitations of the A algorithm.*

Computational cost	**Handles dynamic environments**
High (with large crowds)	Information not found
Fast searches	**Factors for human behaviour**
Yes	0
Other	
The node cost can represent many things	Guaranteed to find path

Just like Dijkstra's algorithm, the A* algorithm is guaranteed to find a path, if there are one, from start node to goal node. To lessen the cig computationally cost in Dijkstra's algorithm, the A* algorithm uses a heuristic search which only searches the most promising nodes. Furthermore, the Heuristic function in the A* algorithm is the most efficient among path finding algorithms. The A* algorithm expands fewer nodes than any other path finding algorithm. When simulating movement among big crowds, the computational cost can get too big, which leads to unrealistic behaviour among agents, as seen in for example the strategical computer game Age of Empires (Cui & Shi, 2011). A big strength is that the algorithms can find the least cost path for many different costs. This means that it can be used to find either the shortest or the quickest path, or the least dangerous, or almost anything (Stout, 1997).

Anytime repair A*
The strengths and limitations of the anytime repair A* algorithm found in literature is presented here and summarized in Table 6.

Table 6. Strengths and limitations of the anytime repair A algorithm.*

Computational cost	**Handles dynamic environments**
Information not found	No
Fast searches	**Factors for human behaviour**
Yes	0
Other	
Solution improves with time	Always returns solution

This algorithm always returns a solution to the shortest path problem, which can be useful if you have a time constraint. In the beginning the solution will be sub-optimal, and then it will improve over time. There is no function in the algorithm which can deal with changes in the environment, which is the reason for why it only handles static environments (Anguelov, 2011).

If you have a lot of time at your hand, then the fact that it only gives an optimal solution after a certain time is no problem. Then it is rather a good thing that it only gets better with time. If you have a time constraint and have to deal with a sub-optimal solution, then it might not be the best. But a sub-optimal solution is probably still better than no solution.

Dynamic A* Lite/D* Lite
The strengths and limitations of the dynamic A* lite/D* lite algorithm found in literature is presented here and summarized in Table 7.

Table 7. Strengths and limitations of the dynamic A lite/D* lite algorithm.*

Computational cost	**Handles dynamic environments**
Information not found	Yes
Fast searches	**Factors for human behaviour**
Information not found	0
Other	
Can deal with moving agents at start	

This algorithm is constructed to deal with the problem of moving agents at the start, which makes it the biggest strength of this algorithm (Anguelov, 2011).

Anytime dynamic A*
The strengths and limitations of the anytime dynamic A* algorithm found in literature is presented here and summarized in Table 8.

Table 8. Strengths and limitations of the anytime dynamic A algorithm.*

Computational cost	**Handles dynamic environments**
Information not found	Yes
Fast searches	**Factors for human behaviour**
Yes	0
Other	
Solution improves with time	Always returns solution

This algorithm always returns a solution to the shortest path problem, which can be useful if you have a time constraint. In the beginning the solution will be sub-optimal, and then it will improve over time. This algorithm is an improvement of the anytime repair A* that also works in dynamic environments (Anguelov, 2011).

Hierarchical pathfinding A*
The strengths and limitations of the hierarchical pathfinding A* algorithm found in literature is presented here and summarized in Table 9.

Table 9. Strengths and limitations of the hierarchical pathfinding A algorithm.*

Computational cost	**Handles dynamic environments**
High (when adding more complexity)	Yes
Fast searches	**Factors for human behaviour**
Yes	0

The main concept of this algorithm is that it breaks up the environment hierarchically which makes it very quick. The hierarchy is divided into levels, or sub-paths, which gives it the ability to change the route as the environment changes, due to the fact that it calculates each sub-path one at a time. Therefore, this algorithm is applicable in both static and dynamic environments (Cui & Shi, 2011; Botea, Müller, & Schaeffer, 2004; Anguelov, 2011). Test has shown that this algorithm gives solution to the shortest path problem that are less than 10 % sub-optimal (Anguelov, 2011).

Iterative deepening A*
The strengths and limitations of the iterative deepening A* algorithm found in literature is presented here and summarized in Table 10.

Table 10. Strengths and limitations of the iterative deepening A algorithm.*

Computational cost	**Handles dynamic environments**
High	Information not found
Fast searches	**Factors for human behaviour**
No	0

Since this algorithm divides the path in smaller pieces less memory is required (Cui & Shi, 2011; Botea, Müller, & Schaeffer, 2004; Anguelov, 2011; Stout, 1997). However, the search for the shortest path is slower and more inefficient than for the A* algorithm (Botea, Müller, & Schaeffer, 2004; Anguelov, 2011).

The fringe search algorithm
The strengths and limitations of the fringe search algorithm found in literature is presented here and summarized in Table 11.

Table 11. Strengths and limitations of the fringe search algorithm.

Computational cost	**Handles dynamic environments**
Low	Information not found
Fast searches	**Factors for human behaviour**
Yes	0

Because this algorithm is a variant of the iterative deepening A* that has been modified to deal with the inefficiency, it still demands less memory while performing fast searches for the shortest path (Anguelov, 2011).

Lifelong planning A*
The strengths and limitations of the lifelong planning A* algorithm found in literature is presented here and summarized in Table 12.

Table 12. Strengths and limitations of the lifelong planning A algorithm.*

Computational cost	**Handles dynamic environments**
Information not found	Yes
Fast searches	**Factors for human behaviour**
Information not found	0

This version of the A* algorithm can adjust to a changing environment by updating nodes that have been altered (Anguelov, 2011).

Memory-bounded search A*

The strengths and limitations of the memory-bounded search A* algorithm found in literature is presented here and summarized in Table 13.

Table 13. Strengths and limitations of the memory-bounded search A algorithm.*

Computational cost	**Handles dynamic environments**
High	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0

This algorithm demands less memory by deleting nodes with low interest which on the other hand results in loss of information, recalculation and a higher computationally cost (Anguelov, 2011).

Real time A*

The strengths and limitations of the real time A* algorithm found in literature is presented here and summarized in Table 14.

Table 14. Strengths and limitations of the real time A algorithm.*

Computational cost	**Handles dynamic environments**
Low	No
Fast searches	**Factors for human behaviour**
Yes	1

Due to the characteristics of this algorithm the occupant will only search for a path in the visible area (Anguelov, 2011). The author sees this both as a strength and a limitation depending on the situation. In an unfamiliar environment, it is comparable to the real life, but in familiar environment it is not. The fact that it only searches the visible path as the occupant travel makes the memory requirements smaller and faster. However, this algorithm is not applicable in dynamic environments due to the fact that it can not handle when a previously blocked cell becomes allowed to travel through (Anguelov, 2011).

Biased random walk

The strengths and limitations of the biased random walk method found in literature is presented here and summarized in Table 15.

Table 15.. Strengths and limitations of the biased random walk.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0

The authors who developed this method does not state any of the strengths or limitations that are of interest to this report.

4.1.2 Shortest path methods

This sub-chapter describes the strengths and limitations of the methods for shortest path finding.

Surplus distance method

The strengths and limitations of the surplus distance method found in literature is presented here and summarized in Table 16.

Table 16. Strengths and limitations of surplus distance method.

Computational cost	**Handles dynamic environments**
Information not found	Yes
Fast searches	**Factors for human behaviour**
Information not found	0

Since the movement of agents is recomputed every 0,1 seconds it can handle dynamic environments were for example obstacle appear in front of the agent (Thompson, 1994; Thompson, Wu, & Marchant, 1997). However, the evacuation process can not be displayed in real time. Instead, it is recorded on the hard drive making it possible to playback later (Thompson, Wu, & Marchant, 1997).

Fuzzy traffic assignment model

The strengths and limitations of the fuzzy traffic assignment model found in literature is presented here and summarized in Table 17.

Table 17. Strengths and limitations of the fuzzy traffic assignment model.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Yes	0

Because this model only decides the route choice at intersections, and not between complete routes from current location to goal destination, it saves a lot of time calculating (Henn, 1998).

4.1.3 Quickest path algorithms

This sub-chapter describes the strengths and limitations of the algorithms for quickest path finding.

A* algorithm

The strengths and limitations of this algorithms are seen in the sub-chapter about shortest path algorithms.

Advanced capacitated network flow transshipment algorithm
The strengths and limitations of the advanced capacitated network flow transshipment algorithm found in literature is presented here and summarized in Table 18.
Table 18. Strengths and limitations of the advanced capacitated network flow transshipment algorithm.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0

There is limited information about the advanced capacitated network flow transhipment algorithm and therefore limited information about its strengths and limitations, to say none.

4.1.4 Quickest path methods
This sub-chapter describes the strengths and limitations of the methods for quickest path finding.

Multinomial logit
The strengths and limitations of the multinomial logit model found in literature is presented here and summarized in Table 19.
Table 19. Strengths and limitations of the multinomial logit model.

Computational cost	**Handles dynamic environments**
Low	Information not found
Fast searches	**Factors for human behaviour**
Yes	0

Multinomial is computationally efficient because of its mathematical structure and therefore can make faster calculations (Koppelman & Wen, The Paired Combinatorial Logit Model: Propertiesm Estimation and Application, 2000).

C-logit
The strengths and limitations of the c-logit model found in literature is presented here and summarized in Table 20.
Table 20. Strengths and limitations of the c-logit model.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0

This model solved the problem with the multinomial logit model where it calculated unrealistic probabilities when different choices shared links (Ben-Akiva & Bierlaire, 1999). However, information about the characteristics regarding strengths and limitations that is searched for is lacking. This is the case for basically all the logit models.

Nested logit
The strengths and limitations of the nested logit model found in literature is presented here and summarized in Table 21.
Table 21. Strengths and limitations of the nested logit model.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0

This models solved the problem with the multinomial logit models where it assumed that all alternatives are equal (Prato, 2009). More information about strengths and limitations have not been found.

Cross-nested logit
The strengths and limitations of the cross-nested logit model found in literature is presented here and summarized in Table 22.
Table 22. Strengths and limitations of the cross-nested logit model.

Computational cost	**Handles dynamic environments**
High	Information not found
Fast searches	**Factors for human behaviour**
No	0

The model solved the problem with the nested logit model where alternatives in different nests could not be combined (Ben-Akiva & Bierlaire, 2003). This model is computational demanding because it is complex and requires longer calculation times (Prato, 2009).

Paired combinatorial logit
The strengths and limitations of the paired combinatorial logit model found in literature is presented here and summarized in Table 23.
Table 23. Strengths and limitations of the paired combinatorial logit model.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0

Just like the cross-nested logit model this model solves the problem with the nested logit model where alternatives in different nest could not be combined (Koppelman & Wen, The Paired Combinatorial Logit Model: Propertiesm Estimation and Application, 2000).

Path-size logit
The strengths and limitations of the path-size logit model found in literature is presented here and summarized in Table 24.
Table 24. Strengths and limitations of the path-size logit model.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0

This model is just like the c-log model an attempt to fix the problem with paths that share links in the multinomial logit model (Prato, 2009).

Locally quickest algorithm
The strengths and limitations of locally quickest algorithm method found in literature is presented here and summarized in Table 25.

Table 25. Strengths and limitations of the locally quickest algorithm.

Computational cost	**Handles dynamic environments**
Information not found	No
Fast searches	**Factors for human behaviour**
Information not found	3

Although the technical reference does not specifically discuss this, it does not look like the model can handle dynamic environments (Thunderhead, 2016).

This method assumes that the occupant has compete knowledge of doors and queue time in the current room. Furthermore, it assumes that the occupant has knowledge of the distance from the door to the final goal (Thunderhead, 2016). That it assumes knowledge of the current room is comparable with real life. That it assumes knowledge of distances outside of the room can be comparable with real life if the occupant is familiar with the building, and it can be unrealistic if the occupant is new to the building. This can be seen as both strengths and limitations depending on the situation. However, this is just speculations from the author.

STEPS method
The strengths and limitations of STEPS method found in literature is presented here and summarized in Table 26.
Table 26. Strengths and limitations of STEPS method.

Computational cost	**Handles dynamic environments**
Information not found	No
Fast searches	**Factors for human behaviour**
Information not found	3

STEPS method implements patience when queuing, herding and the possibility to block walkable areas. There are no signs of it being able to handle dynamic environments (Waterson & Pellissier, 2010).

4.1.4 Conditional path algorithms

This sub-chapter describes the strengths and limitations of the algorithms for quickest path finding.

Multiobjective optimization genetic algorithm

The strengths and limitations of the multiobjective optimization genetic algorithm found in literature is presented here and summarized in Table 27.

Table 27. Strengths and limitations of the multiobjective optimization genetic algorithm.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	1
Other	
Considers several factors	

This algorithm aims to minimize total evacuation time, travelled distance and queuing time. This is done by predefined values set by the user, which can be hard to assign correctly (Li, Fang, Li, & Zong, 2010).

4.1.4 Conditional path methods

This sub-chapter describes the strengths and limitations of the methods for quickest path finding.

Adaptive decision making method

The strengths and limitations of the adaptive decision making method found in literature is presented here and summarized in Table 28.

Table 28. Strengths and limitations of the adaptive decision making method.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	5

This method, used in EXODUS, is adaptive in the meaning that the occupants can change their target exit during the evacuation simulation. Furthermore, it implements some human behaviour like familiarity to exits, visibility of exits and queuing time (Gwynne, Galea, Lawrence, Owen, & Filippidis, 2000). Even fire attributes are implemented (Gwynne, Galea, Owen, Lawrence, & Filippidis, 2004).

Game theoretic reaction function model
The strengths and limitations of the game theoretic reaction function model found in literature is presented here and summarized in Table 29.

Table 29. Strengths and limitations of the game theoretic reaction function model.

Computational cost	**Handles dynamic environments**
High	Yes
Fast searches	**Factors for human behaviour**
Information not found	7

This model in FDS+Evac has started implementing human behaviour like herding, familiarity to exits and fire conditions like smoke density. The social behaviour is however not validated yet. The smoke density can, if reach a user defined value, make exits unusable. Agents can even get incapacitated if exposed too long to fire products in the smoke (Korhonen, 2015). The user guide mentions shortcomings like restrictions in the geometry because of the rectangular mesh, difficult for user to construct model and that it can be computational demanding (Korhonen, 2015).

Labelling approach
The strengths and limitations of the labelling approach found in literature is presented here and summarized in Table 30.

Table 30. Strengths and limitations of the labelling approach.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	3

This approach is used in traffic assignments and considers that traveller's objectives when driving are different; some may want to minimize travel time while some do not want to travel the bigger roads. However, there seem to be a lack of information about traveller's preferences for the approach to work as intended (Prato, 2009).

Simulation approach
The strengths and limitations of the simulation approach found in literature is presented here and summarized in Table 31.

Table 31. Strengths and limitations of the simulation approach.

Computational cost	**Handles dynamic environments**
Information not found	Information not found
Fast searches	**Factors for human behaviour**
Information not found	0
Other	
Can simulate unrealistic paths	

Because the calculated paths are results of a probability distribution, many different paths will be generated. However, because of truncation of negative values some unrealistic paths can get generated (Prato, 2009).

4.3 Summary

Information about computational cost, fast searches and the ability to handle dynamic environments have in several cases not been found, making it difficult to compare amongst each other. Because of this, the comparison focuses on the implementation of factors for human behaviour. The algorithms or methods that implements factors for human behaviour are shown in Table 32. Four out of six of those are used in some of the most common evacuation models. The algorithm for route choice used in Simulex is the only one of the common models that do not explicitly represent behavioural factors affecting route choice.

Table 32. Algorithms or methods implementing factors for human behaviour.

Algorithms/methods	Factors
Real time A*	Visual perception
Locally quickest algorithm (Pathfinder)	Elevators, queuing time, walking time
STEPS method (STEPS)	Exit block/obstacles, herding, queuing time
Multiobjective optimization genetic algorithm	Queuing time, walking time
Adaptive decision making model (EXODUS)	Affiliation, queuing time, smoke, temperature, visual perception
Game theoretic reaction function model (FDS+Evac)	Affiliation, exit block/obstacles, herding, queuing time, smoke, visual perception, walking time,
Labelling approach	Affiliation, queuing time, walking time

Table 33 shows in how many algorithms or methods the factors are implemented in.

Table 33. Frequency of used factors that affects route choice.

Factors	Times implemented
Queuing time	6
Walking time	4
Visual perception	3
Affiliation	3
Smoke	2
Herding	2
Exit block/obstacles	2
Elevators	1
Temperature	1

The algorithms and methods, implementing some behavioural factors seem promising. Human behaviour in evacuation scenarios is very complex and difficult to predict. However, it affects the chosen exit route and the implementation of different behavioural factors can have significant impact on the evacuation time (Georgoudas, Sirakoulis, & Andreadis, 2008).

When talking about evacuation simulation, the impact of human behaviour is key issue that needs to be addressed. Most of the algorithms and methods described in this paper do not fully implement the variety of behaviours that might impact route choice. Some models do, but far from every factor that might affect route choice listed in Table 2 in the category section is implemented. The factors not yet implemented explicitly in models are summarized in Table 34. However, most models have the flexibility to represent these factors implicitly

(i.e. the user can represent their impact manually). In Pathfinder for example, the user can assign occupants to move along a certain route. This can be done to simulate an occupant helping another occupant, or an occupant investigating the fire. To be able to implement all the listed factors, more research is needed.

Table 34. Factors that have not been directly implemented.

Human behaviour	**Fire conditions**	**Building construction**
Helping other occupants		Wayfinding installations/Signage (static or dynamic)
Investigate fire		Available egress capacity
Alarm awareness		Alarm type
Staff giving directions		

As seen above there are still some factors that do affect route choice and that need to be implemented explicitly.

5. TESTING

This section describes testing for verification of route choice. Verification is defined as the "*process of determining that a calculation method implementation accurately represents the developer's conceptual description of the calculation method and the solution to the calculation method.*" (International Standards Organization, 2015). Testing is done to be able to assess if the models work as intended. Today there is a lack of standard protocols for verification of evacuation models and this makes it difficult to compare evacuation models functions amongst each other (Ronchi , 2014). Since validation and verification of evacuation models is the most important to model users (Ronchi & Kinsey, 2011), it is important to further develop such tests for route choice. Since the development of validation tests demands an extensive work with collection and analysis of experimental data regarding route choice in different conditions, the focus have been on suggesting verification tests. This is the first step towards the inclusion of all factors which are needed in the correct simulation of route choice. The suggested tests in this section should be seen as a first step towards developing standardized tests for route choice in evacuation models. The verification tests for route choice available today is described in the following sub-chapter.

5.1 Current tests

There is a German standard that includes a test to verify that the algorithms for route choice in the evacuation programs work as intended. The test is called "Test 14" and the geometry is shown in Figure 4. In the test, the agents have to choose between using the shortest route, via stairs, or the longer route on the same floor, when moving from start to target (RiMEA e. V., 2016). To speak in general terms, the test is designed to check that the occupants in the evacuation simulation use the route they are supposed to use according to the programs design.

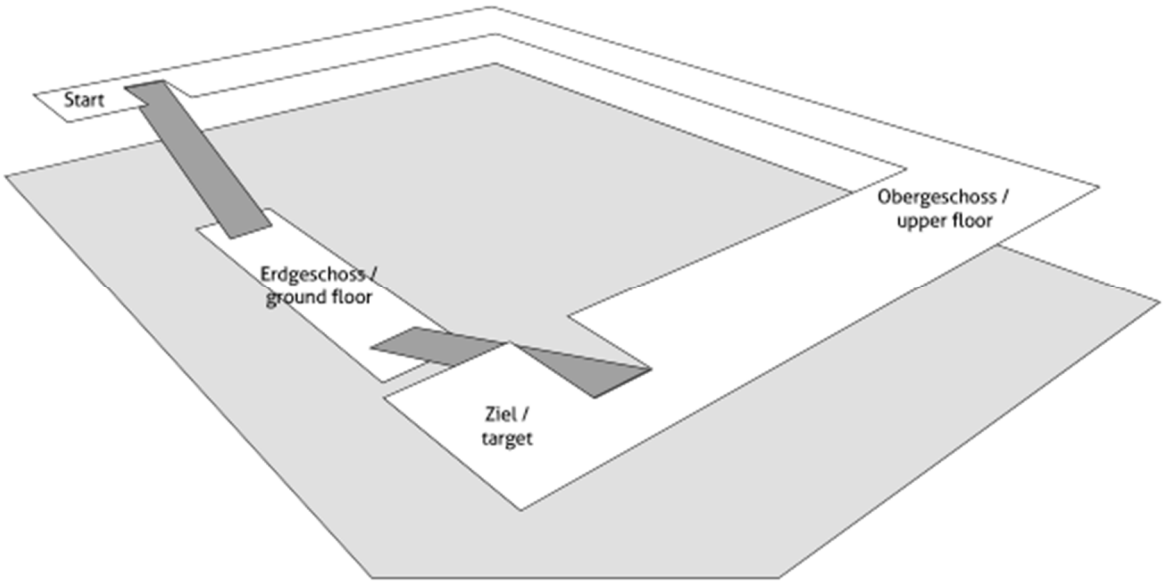

Figure 4. Test 14 for route choice (RiMEA e. V., 2016).

The International Maritime Organization have tests for verifying evacuation models. Those test however, are constructed for maritime purposes, like passenger ships, and not for buildings. Test 10 (IMO_10) is for verifying route choice, and the geometry and setup is shown in Figure 5. The agents in the test is assigned to be males between 30-50 years old, meaning they will ha a walking speed between 0,97-1,62 m/s. The agents in cabin 1,2,3,4,7,8,9, and 10 are assigned the main exit and cabin 11 and 12 are assigned the secondary exit. The expected result is that the agents move to their assigned exits (International Maritime Organization, 2007).

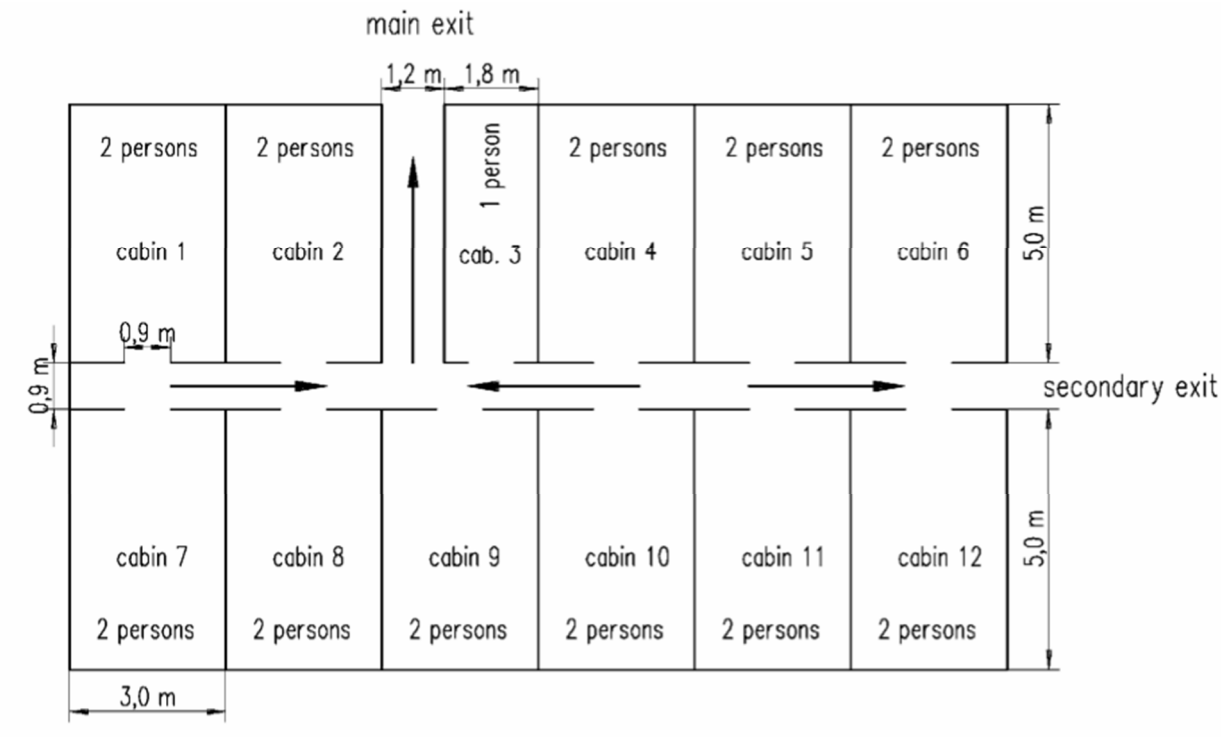

Figure 5. IMO test 10 for exit route allocation (International Maritime Organization, 2007).

Ronchi et al. (2013) have suggested a verification test for social influence, called “Verif 3.2”. The geometry is described in Figure 6 and Figure 7. In scenario 1 (Figure 6) the agent (agent 1) does not have a preferred exit. In scenario 2 (Figure 7) the second agent (agent 2) is assigned to Exit 2. Both scenarios are tested several times to get a stable distribution of chosen exits. The expected result is that agent 1 uses Exit 2 more in scenario 2, compared to scenario 1.

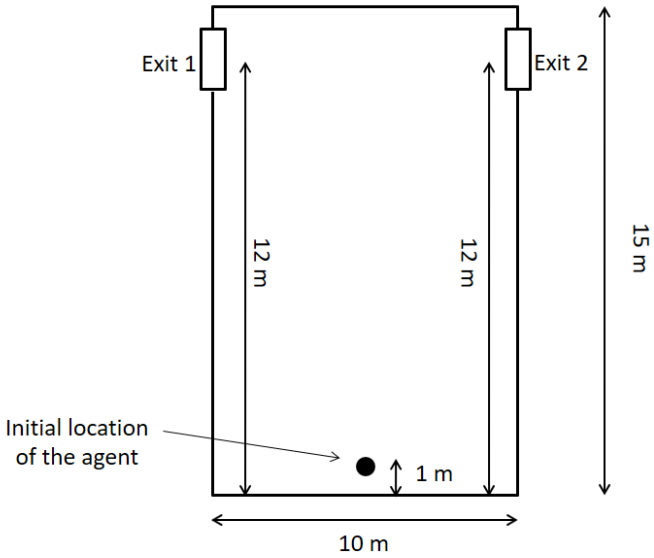

Figure 6. Verification test Verif 3.2, scenario 1 for social influence (Ronchi, Kuligowski, Reneke, Peacock, & Nilsson, 2013).

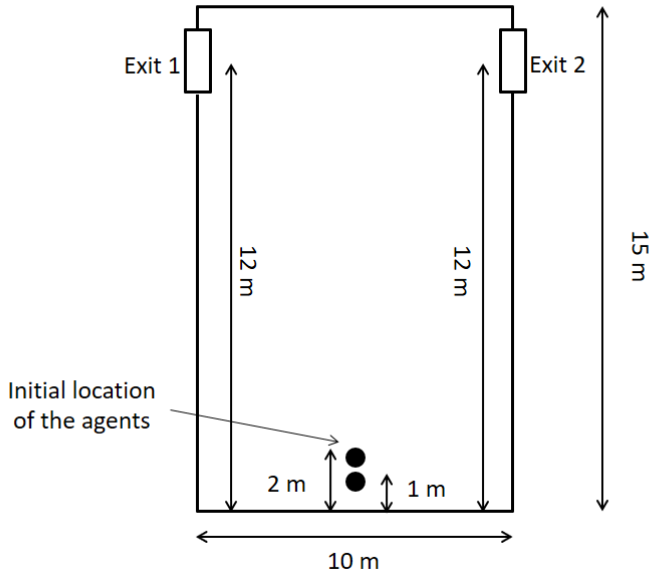

Figure 7. Verification test Verif 3.2, scenario 2 for social influence (Ronchi, Kuligowski, Reneke, Peacock, & Nilsson, 2013).

Ronchi et al. (2013) suggests a verification test for affiliation, called “Verif 3.3”. The geometry is described in Figure 8. In scenario 1 the agent does not have a preferred exit. In scenario 2 the agent is affiliated with Exit 2, but not Exit 1. Both scenarios are tested several times to get a stable distribution of chosen exits. The expected result is that the agent uses Exit 2 more in scenario 2, compared to scenario 1.

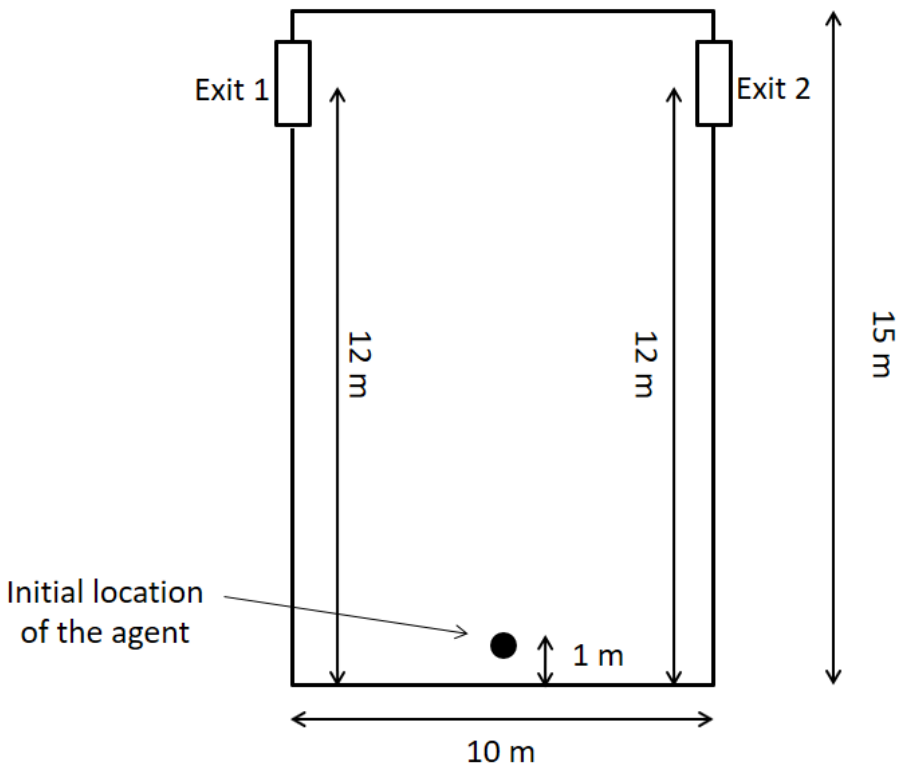

Figure 8. Verification test Verif 3.3 for affiliation (Ronchi, Kuligowski, Reneke, Peacock, & Nilsson, 2013).

Ronchi et al. (2013) suggests a verification test for dynamic availability of exits, called "Verif 4.1". The geometry is described in Figure 9. In this verification test Exit 1 becomes unusable one (1) second into the simulation. The expected result is that the agent only uses Exit 2.

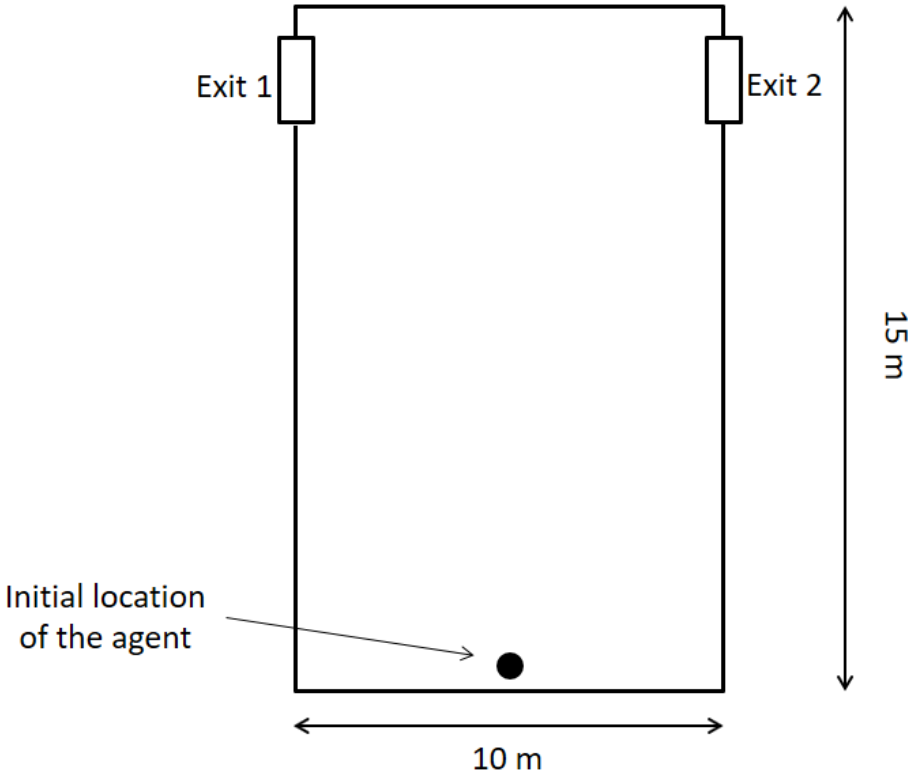

Figure 9. Verification test Verif 4.1 for dynamic availability of exits (Ronchi, Kuligowski, Reneke, Peacock, & Nilsson, 2013).

Ronchi et al. (2013) also suggests test for other factors like for example group behaviour. These tests are however not aimed at verifying route choice when the factors are implemented. They are only meant to verifying the general function of that factor.

However, different tests can be useful to explore the implementation of more complex route choice algorithms (i.e. which makes use of conditions, occupant's decision making, etc.) (Ronchi, Kuligowski, Reneke, Peacock, & Nilsson, 2013).

5.2 Suggested tests

These tests are mainly developed by the author of this thesis. Inspiration for the geometry of these tests were gathered from *The Process of Verification and Validation of Building Fire Evacuation Models* by Ronchi et al. (2013). The exact configuration of each test is however a product of the authors own judgement. The aim when designing the tests was to produce tests which are easy to construct in the evacuation models. But also tests that are intuitive to one without previous knowledge about evacuation models. Because of the limitations within this thesis, the tests have not been evaluated yet and should therefore be subjected to testing before used as they are intended.

Following section presents suggested test for verifying route choice in models that implement human behaviour. The tests will be presented with the same structure used by Ronchi et al. (2013), containing information about the following:

- *Geometry*: describes the configuration of the room(s) in the test.
- *Scenario*: describes the setup regarding agent characteristics.
- *Expected results*: describes what the result is expected to be after the test is done.
- *Test method*: describes if the test is quantitative (comparing test results with expected result as a percentage) or qualitative (comparing test results with expected behaviour based on current behavioural theory).
- *User actions*: describes what is required from the user to make use of the result.

Tests are suggested for verification of the factors affecting route choice from Table 2 in the category section of this thesis. However, not all the factors are relevant for testing at this time. This is because not all the factors are yet implemented in evacuation models. Helping other occupants, investigate fire, alarm awareness, staff giving directions, signage, available egress capacity and alarm type are the factors that might affect route choice, but that are not yet implemented explicitly. Those factors are marked as "not considered".

Even tough elevators are implemented in several of evacuation simulators today (Andrée, Nilsson, & Eriksson, 2016), no verification test will be suggested to verify its effect on route choice. Andrée et al. (2016) showed in experiments that even when the route via an elevator was shorter and faster, the occupants would not exclusively use it. The experiments showed that the choice to use an elevator to evacuate largely depended on wayfinding installations consisting of flashing green light. In the experiment with no flashing light, 59,6 % (31 out of 52) choose the elevators. With flashing light, 90 % (18 out of 20) choose the elevators. This result makes it difficult to correctly implement the effects of evacuation elevators on route choice today, seeing that more experiments are needed to get a better understanding of what affects the occupants to use an elevator to evacuate. To not risk verifying a behaviour that is not in line with the latest research, a verification test will therefore not be suggested for this factor.

The factors for which there already are verification test will not get new suggested tests. Instead, the focus is on the factors currently being implemented in evacuation simulators, but that lacks verification test. However, not all the evacuation simulation models are able to use every one of these tests since not all of them make use of every factor. The factors that will, or will not be subjected to verification test are summarized in Table 35.

Table 35. Factors that need verification tests.

Human behaviour	**Test number**
Social influence/Other occupant's behaviour	Already tested (Verif 3.2)
Affiliation	Already tested (Verif 3.3)
Visual perception	Test 1
Queuing time	Test 2
Walking time	Test 2
Group interactions/ Herding/ Family interaction	Test 3
Leadership	Test 4
Helping other occupants	Not considered
Investigate fire	Not considered
Alarm awareness	Not considered
Staff giving directions	Not considered
Fire conditions	
Radiation	Test 5
Smoke	Test 6
Temperature	Test 7
Building construction	
Wayfinding installations/Signage	Not considered
Exit block/obstacles	Test 8
Elevator	Not considered
Available egress capacity	Not considered
Alarm type	Not considered

As seen there are several of factors that will not be given verification tests. However, those that will be given verification tests are described in following sub-chapters.

5.2.1 Test 1

The described test is a verification test for visual perception when choosing exit route. The occupant should only use the exit that is in the line of sight (represented by the red lines in Figure 10).

Geometry: the geometry is described in Figure 10. The exits are 1 meter wide.

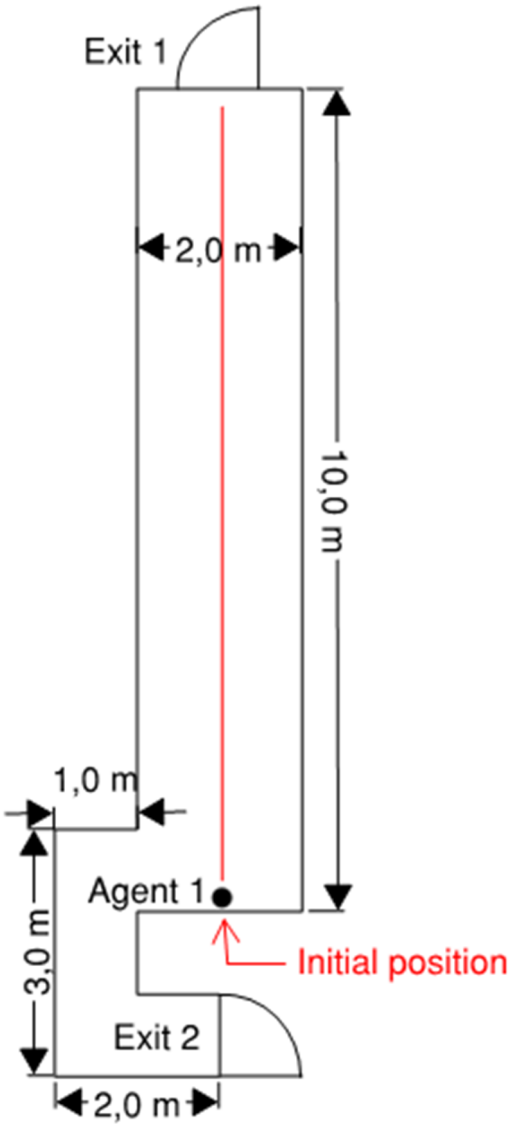

Figure 10. Verification test 1 for route choice.

Scenario: Agent 1 is placed in a corridor according to Figure 10 having the response time set to 0 seconds and a walking speed of 1 m/s. Agents 1 is not familiar with any of the exits.

Expected results: Agent 1 is expected to choose Exit 1 (the visible exit).

Test method: The verification is done in a quantitative way.

User actions: The test demands that the factor for visual perception overwrites the presence of a door reachable in a shorter distance or time. The user should document if this is not the case.

5.2.2 Test 2

The described test is a verification test for queuing time and walking time when choosing exit route. When testing the queuing time the occupant should use the exit with the shortest queuing time. When testing the walking time the occupant should use the exit that has the least walking time to reach.

Geometry: the geometry is described in Figure 11. The exits are 1 meter wide.

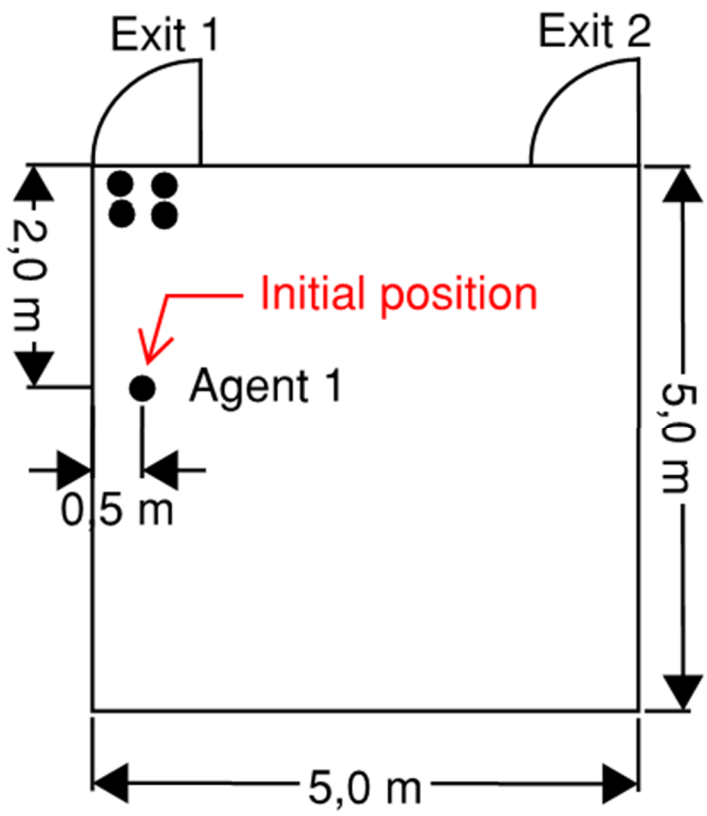

Figure 11. Verification test 2 for route choice.

Scenario: Agent 1 is placed in a room according to Figure 11, behind four other occupants queuing in front of a door. Agent 1 has a response time set to 0 seconds and a walking speed of 1 m/s. The four occupants in front of the door have a response time set to 10 seconds and a walking speed of 1 m/s.

Expected results: When testing an algorithm based on quickest time, Agent 1 is expected to use Exit 2 (it is at most 6 meters from the initial position to Exit 2, resulting in an evacuation time in 6 seconds in contrast to waiting in line for roughly 10 seconds). When testing an algorithm based on shortest distance, Agent 1 is expected to use Exit 1.

Test method: The verification is done in a quantitative way.

User actions: If the model uses an algorithm for shortest path the agent might wait in a queue at Exit 1. If the model uses an algorithm for quickest path the agent might use Exit 2. When testing walking time Agent 1 should stand still while queuing at Exit 1, otherwise the constant walking in one spot might count as walking time This should be documented.

5.2.3 Test 3

The described test is a verification test for group interactions/herding/family interactions when choosing exit route. Agent 1 should stick together with the group agents.

Geometry: the geometry is described in Figure 12. The exits are 1 meter wide.

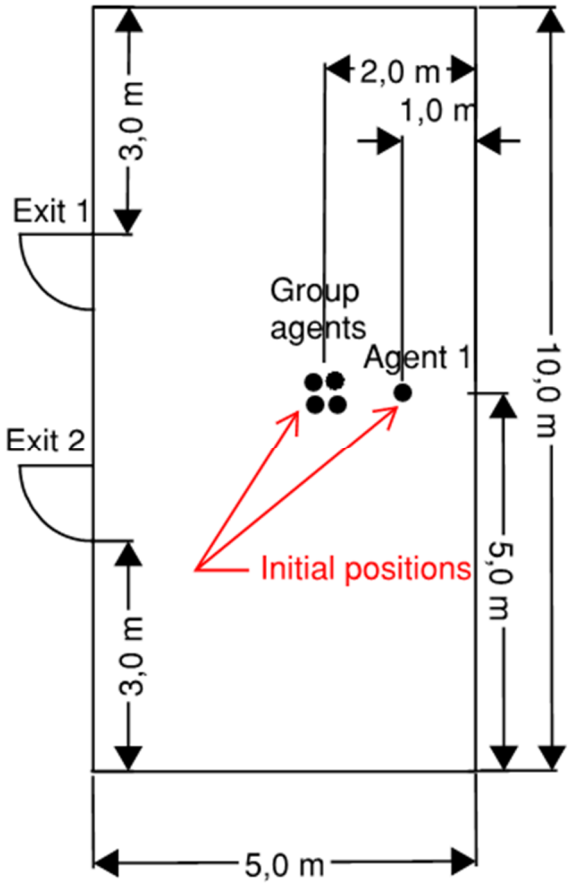

Figure 12. Verification test 3 for route choice.

Scenario: Agent 1 is placed behind a group of four occupants according to Figure 12. Agent 1 has a response time set to 0 seconds and a walking speed of 1 m/s. The group agents have a response time set to 5 seconds, a walking speed of 1 m/s and are deterministically assigned Exit 2. Agent 1 has no preferred exit, but is assigned to belong to the group agents.

Expected results: Agent 1 is expected to use Exit 2.

Test method: The verification is done in a quantitative way. However, it can also be done in a qualitative way, checking that Agent 1 follows the group agents using either the visualization tool of the model (if available) or the path of Agent 1.

User actions: If the verification is done in a quantitative way and Agent 1 chooses Exit 2, the user should control that Agent 1 follows the group agents and do not walk to Exit 2 by itself.

5.2.4 Test 4

The described test is a verification test for leadership when choosing exit route. The group occupants should follow the leader.

Geometry: the geometry is described in Figure 13. The exits are 1 meter wide.

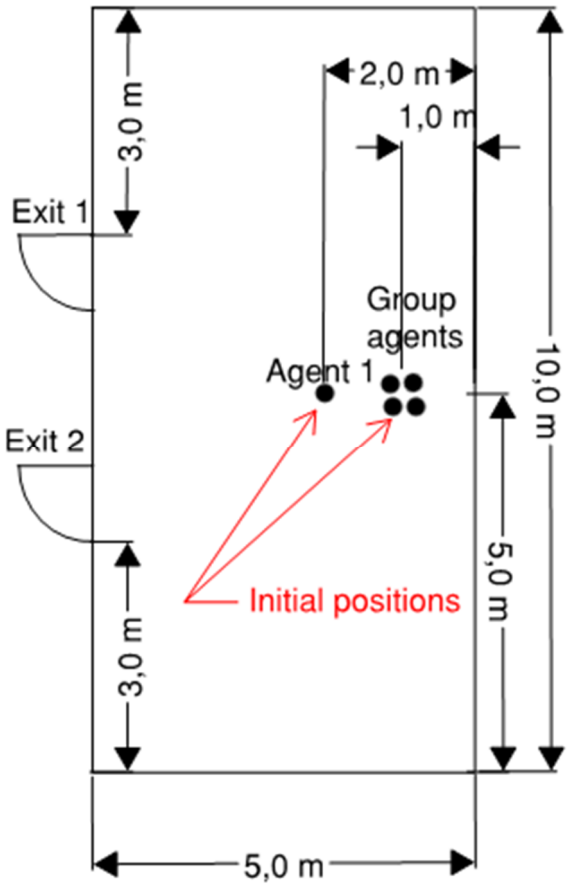

Figure 13. Verification test 4 for route choice.

Scenario: Agent 1 is placed in front of a group of four occupants according to Figure 13. The group agents have a response time set to 0 and a walking speed of 1 m/s. Agent 1 have a response time set to 5 seconds and a walking speed of 1 m/s. Agent 1 are deterministically assigned Exit 2. The group of agents has no preferred exit, but is assigned to follow Agent 1 (the leader).

Expected results: The group of agents is expected to use Exit 2.

Test method: The verification is done in a quantitative way. However, it can also be done in a qualitative way, checking that the group follows Agent 1 using either the visualization tool of the model (if available) or the path of Agent 1.

User actions: If the verification is done in a quantitative way and the group agents chooses Exit 2, the user should control that the group agents follows Agent 1 and do not walk to Exit 2 by their self.

5.2.5 Test 5

The described test is a verification test for radiation when choosing exit route. The occupants should not walk through the area with raised radiation.

Geometry: the geometry is described in Figure 14. The exits are 1 meter wide.

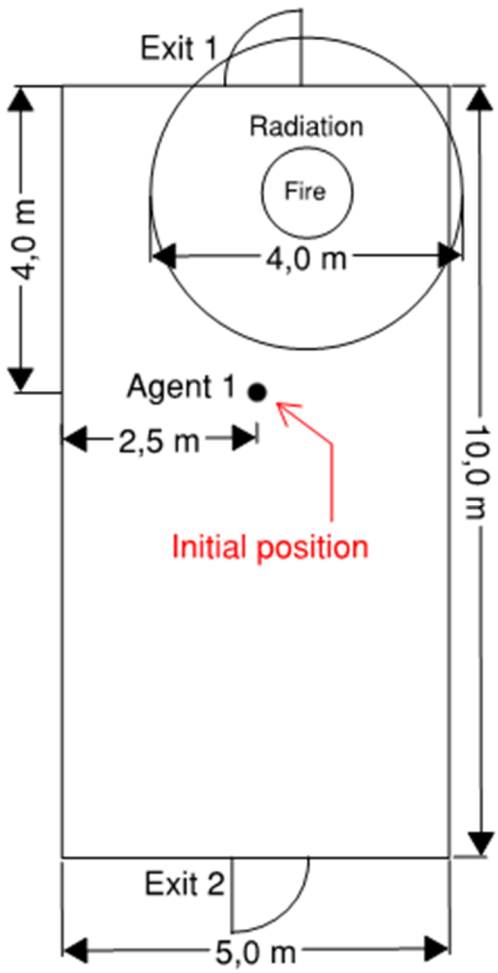

Figure 14. Verification test 5 for route choice.

Scenario: Agent 1 is placed closer to Exit 1 and has a response time set to 10 seconds and a walking speed of 1 m/s. In front of Exit 1 is an area with untenable radiation levels caused by the fire.

Expected results: Agent 1 is expected to use Exit 2.

Test method: The verification is done in a quantitative way. However, it can also be done in a qualitative way, controlling that Agent 1 does not try to walk through the area with raised radiation levels.

User actions: The user must set the effect of radiation at a level that makes the exit blocked and untenable conditions in that area. This should be done by making an initial simulation without occupants, where the radius of the radiation is measured. Also, the user must set the smoke and temperature to not be harmful.

5.2.6 Test 6

The described test is a verification test for smoke when choosing exit route. The occupants should not walk through the smoke.

Geometry: the geometry is described in Figure 15. The exits are 1 meter wide.

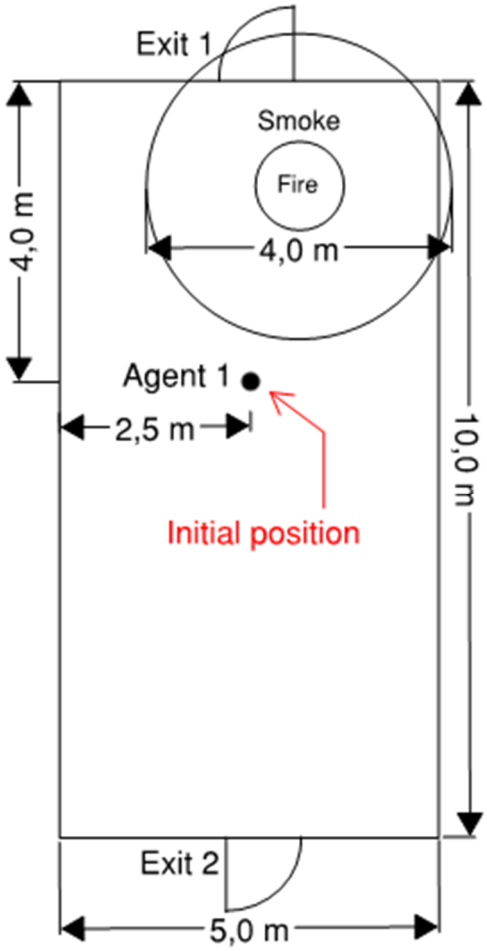

Figure 15. Verification test 6 for route choice.

Scenario: Agent 1 is placed closer to Exit 1 and has a response time set to 10 seconds and a walking speed of 1 m/s. In front of Exit 1 is heavy smoke caused by the fire.

Expected results: Agent 1 is expected to use Exit 2.

Test method: The verification is done in a quantitative way. However, it can also be done in a qualitative way, controlling that Agent 1 does not try to walk through the smoke.

User actions: The user must set the density of the smoke at a level that makes the exit blocked. This should be done by making an initial simulation without occupants, where the radius of the smoke is measured. Also, the user must set the radiation and temperature to not be harmful.

5.2.7 Test 7

The described test is a verification test for temperature when choosing exit route. The occupant should not walk through the high temperature area.

Geometry: the geometry is described in Figure 16. The exits are 1 meter wide.

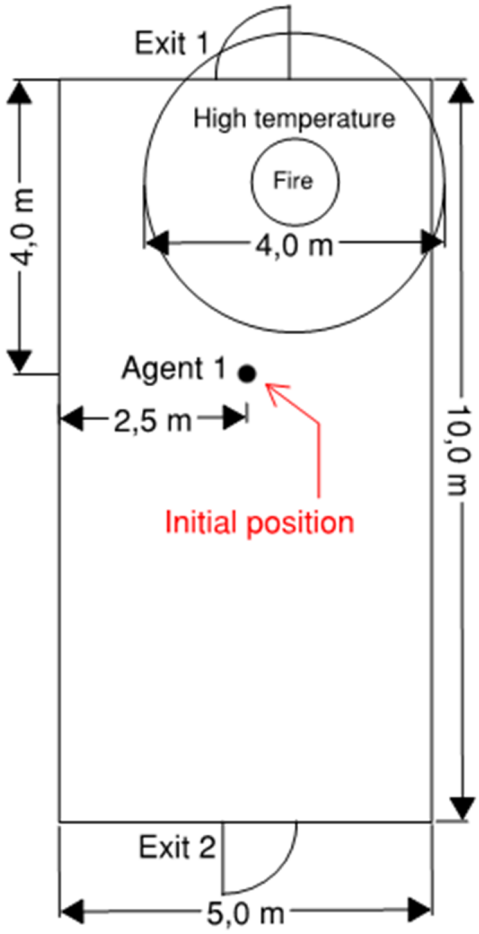

Figure 16. Verification test 7 for route choice.

Scenario: Agent 1 is placed closer to Exit 1 and has a response time set to 10 seconds and a walking speed of 1 m/s. In front of Exit 1 is an area with an untenable temperature caused by the fire.

Expected results: Agent 1 is expected to use Exit 2.

Test method: The verification is done in a quantitative way. However, it can also be done in a qualitative way, controlling that Agent 1 does not try to walk through the high temperature area.

User actions: The user must set the temperature at a level that makes the exit blocked. This should be done by making an initial simulation without occupants, where the radius of the high temperature is measured. Also, the user must set the smoke and radiation to not be harmful.

5.2.8 Test 8

The described test is a verification test for exit block/obstacles when choosing exit route. The occupant should not use the exit that is blocked.

Geometry: the geometry is described in Figure 17. The exits are 1 meter wide.

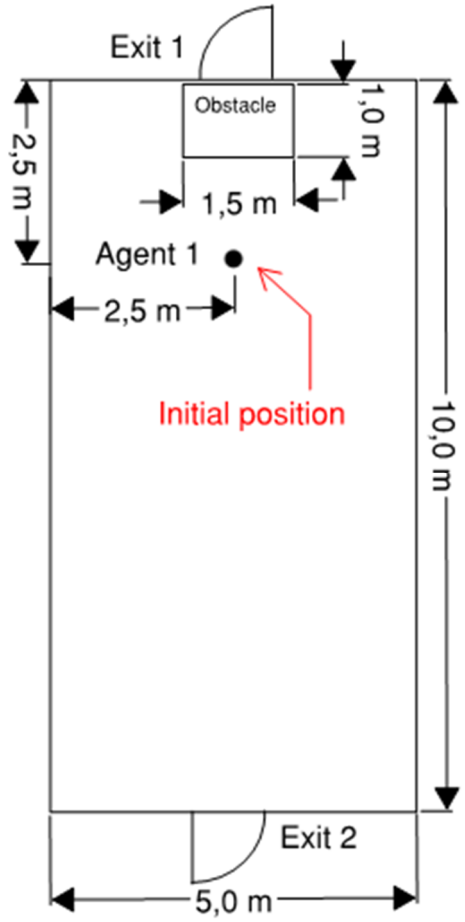

Figure 17. Verification test 8 for route choice.

Scenario: Agent 1 is placed close to Exit 1 and has a response time set to 0 seconds and a walking speed of 1 m/s. In front of Exit 1 is a 1,5 meter high obstacle placed so close to Exit 1 that it is not possible to pass.

Expected results: Agent 1 is expected to use Exit 2.

Test method: The verification is done in a quantitative way. However, it can also be done in a qualitative way, controlling that Agent 1 does not try to walk through or pass the obstacle.

User actions: The user should control that the obstacle is placed in such a way that it is not passable by the agent, but that the exit still can be seen behind it.

6. Discussion

At the start of this thesis the main objective was to look at algorithms for route choice. But as the literature study proceeded it became clear that a lot of the information was about methods or models as well. Most of the "pure" algorithms for route choice were based on A*. So, this thesis evolved into looking at methods and models too. However, sometimes it has been hard to make a distinction between different algorithms and methods. For example, the A* algorithm is implemented in other models or with other algorithms. Pathfinder 2016 uses the locally quickest algorithm for the exit choice, but the travelling between targets is done with the A* algorithm (Thunderhead, 2016). Furthermore, the A* algorithm is implemented in NavMeshes (Cui & Shi, 2012), which at first was seen as a separate model for route choice. The NavMesh is instead seen in this thesis as a way of representing the geometry in which algorithms (like the A*) for route choice can be implemented.

Also, some methods have been difficult to categorize. For example, the multiobjective optimization genetic algorithm that both searches for the quickest and shortest path. In the case of this example, the method got categorized as being conditional. However, the reader needs to keep in mind that the categorization has sometimes been made according to the authors judgement.

As stated in the thesis, only a brief explanation of each algorithms and method has been done. This in not only because of the time limitation, but there has also been difficult finding detailed information. Some algorithms and methods would need a more in-depth analysis, but in some cases, information is very limited. However, a lot of the algorithms and methods needs further investigation, to be able to draw conclusion about them.

The strengths and limitations of some algorithms and models have been difficult to find explicitly stated in literature. Furthermore, the author feels that it is problematic comparing all the algorithms and methods against each other based on the strengths and limitations found. For example, the multinomial logit model is stated to be computationally efficient while the cross-nested logit model is more computationally demanding. Seeing that these models both belong to the logit family comparison between the two is possible. But if you then look at the A* family, where the A* is computationally demanding and the fringe search is not. As in the logit family, comparison between the two versions of the A* algorithms are possible. But this is not the case if you for example want to compare which one of the A* algorithm and cross-nested logit model that is most computationally demanding.

Queuing time is the most commonly implemented factor affecting route choice. The author assumes this is because it is probably easy to implement just by adding a function for the calculation of time. The same argument is valid for walking time which is the second most implemented factor. The third most implemented factors are visual perception and affiliation. The reason for visual perception standing out above several other factors is likely because it is the main function for path-finding in Real time A*. Otherwise it would be equally common as herding, exit block/obstacles and smoke.

It should also be stated that the factors for human behaviour is not likely equally important when evacuating. However, when comparing algorithms and evacuation simulators in this thesis, the comparison has been based on the number of factors implemented. This is not realistic, since some factors are presumably more important than others. This means that when

comparing algorithms and evacuation simulators, one must consider the importance of each factors that are implemented, and not only the sheer number of factors implemented. However, to evaluate the importance of each factor have not been within the scope of this thesis.

In this thesis, it has been assumed that an algorithm implementing human behaviour is more advanced and seen as better than one that do not implement human behaviour. This is not necessarily true. For an algorithm with human behaviour implemented to give a better or more realistic result, every factor should be implemented correctly, and the user input should be in accordance with the latest research. Furthermore, the factors can not be in conflict with each other. A hierarchy where the factors are sorted by their relevance or in an order by their importance to the result, could be one way around this problem.

Amongst the most common evacuation models only one do not explicitly implement any behavioural factors in route choice. This is Simulex which uses the surplus distance method. The latest literature source found on this method is from 1997 which might be the explanation behind this. No public documents have been found on further developments of the model. Seeing that the other models implementing human behaviour have been developed between 2004-2016, it can be a reasonable explanation.

A great limitation seems to be that when implementing more factors and making the algorithm or method more advanced, the computationally cost increases. However, some algorithms and methods have started implementing human behaviour, and a lot of them can find the best path according to their algorithm. How realistic it is for occupants to always find the best route is a question for future research.

There are some algorithms and methods explicitly used in traffic assignments that could possibly be implemented in evacuation models. The labelling approach is the only one implementing factors for human behaviour on route choice. The logit models have been used in traffic assignment models, and even though they do not utilise human behaviour they are more advanced models since they calculate probabilities. The author assumes that it is possible to develop these so that they are able to implement human behaviour represented by probabilities for different route choices. Furthermore, information about some of the logit models are very in-depth, more so than is reflected by this thesis. Therefore, a more in-depth research about these models could be beneficial. This applies also to the labelling approach.

Most verification tests have been straight forward constructing. However, regarding testing elevator usage there have been need for some extra considerations. A test was initially constructed where the agent was supposed to choose the shortest or quickest path, which was using the elevator. But additional findings in studies later indicated that this was not a realistic behaviour. Occupants would not choose to use the elevator based on time or distance. Instead wayfinding installation consisting of flashing green light had the largest impact. Based on this result, the previously constructed test for verifying elevator usage in route choice would not be applicable. Since wayfinding installation as those in the experiments are not yet explicitly implemented in any route choice algorithm, the author felt it was not possible to construct a realistic verification test for elevator usage at this point.

Since the verification tests suggested in this thesis are not yet tested in any way the next step should probably be to test and evaluate them further to see if they are constructed correctly and that they verify what is intended, or if they need improvements.

7. CONCLUSION

It is hard to say which algorithms or method for route choice that is best. Since there has been a lack of information regarding some of the criteria searched for (computational cost, ability to handle dynamic environments, fast searches) it is difficult to directly compare the algorithms and methods amongst each other. The factors concerning human behaviour that are implemented in each algorithm or method are however known. The conclusion will therefore focus on this.

Among the selected reviewed models, the factors implemented in evacuation models today are queuing time (6 times), walking time (4 times), visual perception (3 times), affiliation (3 times), herding (2 times), exit block/obstacles (2 times), smoke (2 times), elevators (1 time) and temperature (1 time).

The models implementing the most factors for human behaviour are EXODUS and FDS+Evac. EXODUS implements five factors and FDS+Evac implements seven factors. Being a continuous model, FDS+Evac is deemed to have a higher computational cost but it can handle dynamic environments. Information about these characteristics of the models have not been found for EXODUS which makes it hard to directly compare the two models against each other. But if a comparison was to be made, FDS+Evac would be slightly preferred based on the number of factors for human behaviour implemented (this assumes that more factors equals better algorithms, see the Discussion). In addition to the algorithms used in these two models, there is the locally quickest algorithm used in Pathfinder and STEPS method. This algorithm and method implements three factors each for human behaviour. Based on the number of factors implemented they are here considered the third best (with one of their main limitations being lack of direct impact of smoke on route choice). However, their capabilities seem reasonably more complete than most algorithms or methods for route choice reviewed in this thesis when looking at factors for human behaviour (there are only three additional algorithms directly implementing human behaviour).

Affiliation is only implemented in three algorithms. Affiliation (which depends on familiarity) however is one of the most common mentioned factors affecting route choice in the literature and the aim should therefore be to directly implement it in any algorithm representing route choice.

Factors affecting route choice are mentioned throughout this thesis. All of them have been found in the literature saying that they can affect route choice. However, there is a lack of information about *how* some of them affect route choice. Some might not need it, for example "following a leader" is self-explanatory. But factors like for example "alarm type" is not a given one. Those factors need further research to be able to implement them in evacuation models, verify and validate them. The tests suggested for some factors in this thesis is a first step towards identifying and verifying human behaviour factors in route choice.

The information summarized about traffic assignment model in this thesis is only a scratch on the surface. To gain a better understanding of those algorithms and methods a dedicated research project on traffic assignment models is needed. This applies especially for the logit family which could potentially be used in evacuation simulations.

As mentioned in the discussion section, there is a problem when comparing strengths and limitations amongst different types of algorithms or methods. The example mentioned

comparison of computational cost between the A* algorithms and the cross-nested logit model. The author suggests that tests for comparison need to be developed if a fair assessment between the different algorithms and methods is to be done. In the case of the used example it could be a complex building with only one functional route, where computationally cost and speed of the search could easily be calculated by the computer.

Route choice in computer games is often done with the A* algorithm (Cui & Shi, 2011; Botea, Müller, & Schaeffer, 2004). However, it seems as if the algorithm has a hard time calculating movement among bigger crowds, as is seen in the computer game Age of Empires (Cui & Shi, 2011). A version of the A* algorithm often used in computer games is the iterative deepening A* algorithm. This is because it demands less memory. However, it is not as fast as the A* algorithm. FDS+Evac uses a game theoretic model in their route choice model. This model is based on a concept used in computer games (Korhonen, 2015). FDS+Evac therefore seems to be the only evacuation model implementing techniques from the computer game world.

One of the found algorithms, the dynamic A* lite/D* lite, has been implemented in route choice for robot navigation. This is because it has the possibility to change the route of moving occupants, or robots in this case, at the start. No additional algorithms for route choice used by robots or artificial intelligence have been found in the present review.

This thesis is focusing on the factors that route choice models represent explicitly. However, some models might be able to represent some of the not yet implemented factors implicitly.

The suggested verification tests for factors affecting route choice is a first step towards better route choice algorithms. By being able to verify the factors affecting route choice, the algorithms for route choice can in the future also be used to validate behaviour when evacuating.

REFERENCES

Andrée, K., Nilsson, D., & Eriksson, J. (2016). Evacuation Experiments in a Virutal Reality High-Rise Building: Exit Choice and Waiting Time for Evacuation Elevators. *Fire and Materials*, 554-567.

Anguelov, B. (2011). *Video Game Pathfinding and Improvements to Discrete Search on Grid-based Maps.* Pretoria: University of Pretoria.

Antonini, G., Bierlaire, M., & Weber, M. (2006). Discrete Choice Models of Pedestrian Walking Behaviour. *Transportation Research Part B 40*, 667-687.

Bae, S., Choi, J.-H., Kim, C., Hong, W., & Ryou, H. (2016). Development of New Evacuation Model (BR-Radiation Model) Through an Experiment. *Journal of Mechanical Science and Technology 30*, 3379-3391.

Bast, H., Funke, S., & Matijevic, D. (2009). *Ultrafast Shortest-Path Queries via Transit Nodes.* The Shortest Path Problem: Ninth DIMACS Implementation Challenge.

Ben-Akiva, M. (1973). *Structure of Passenger Travel Demand Models.* Department of Civil Engineering, MIT, Cambridge, Ma.

Ben-Akiva, M., & Bierlaire, M. (1999). *Discrete Choice Methods and their Applications to Short Term Travel Decisions.*

Ben-Akiva, M., & Bierlaire, M. (2003). Discrete Choice Models with Application to Departure Time and Route Choice. *Hanbook of Transportation Science, Vol. 56*, 7-37.

Ben-Akiva, M., & Bolduc, D. (1996). *Multinomial Probit with a Logit Kernel and a General Parametric Specification of the Covariance Structure.* Colombia University.

Ben-Akiva, M., Bergman, M., Daly, A., & Ramaswamy, R. (1984). Modelling Inter Urban Route Choice Behaviour. *Ninth International Symposium on Transportation and Traffic Theory. VNU Science Press*, 299-330.

Bierlaire, M. (2001). A General Formulation of the Cross-Nested Logit Model. *1st Swiss Transport Research Conference*.

Bierlaire, M. (2006). A Theoretical Analysis of the Cross-Nested Logit Model. *Ann Oper Res*, 287-300.

Björnsson, Y., Enzenberger, M., Holte, R., & Schaeffer, J. (2005). *Fringe Search: Beating A* at Pathfinding on Game Maps.*

Botea, A., Müller, M., & Schaeffer, J. (2004). *Near Optimal Hierarchical Path-Finding.* Edmonton: Department of Computing Science, Univeristy of Alberta.

Bovy, P., Bekhor, S., & Prato, C. (2008). The Factor of Revised Path Size: An Alternative Derivation. *Transportation Research Record, 2076*, 132-140.

Canter, D. (1990). *Fires and Human Behaviour.* London.

Cascetta, E., Nuzzolo, A., Russo, F., & Vitetta, A. (1996). A Modified Logit Route Choice Model Overcoming Path Overlapping Problems. *Transportation and Traffic Theory*, 697-711.

Chooramun, N. (2011). *Implementing a Hybrid Spatial Discretisation within an Agent Based Evacuation Model.* Greenwich: Univerisyt of Greenwich.

Chu, C. (1989). A Paired Combinatorial Logit Model for Travel Demand Analysis. *Proceedings of the 5th World Conference on Transportation Research*, 295-309.

Chu, M., Parigi, P., Law, K., & Latombe, J.-C. (2015). *Simulating Individual, Group and Crowd Behaviours in Building Egress.* Simulation: Transactions of the Society for Modeling and Simulation International.

Cui, X., & Shi, H. (2011). *A*-based Pathfinding in Modern Computer Games.* International Journal of Computer Science and Network Security.

Cui, X., & Shi, H. (2012). An Overview of Pathfinding in Navigation Mesh. *IJCSNS International Journal of Computer Science and Network Security, Vol.12 No.12*, 48-51.

Cutnell, J., & Johnson, K. (2013). *Introduction to Physics, 9th edition.* Singapore: John Wiley & Sons, Inc.

Deutsch, M., & Gerard, H. (1955). A Study of Normative and Informational Social Influnences upon Individual Judgment. *The Journal of Abnormal and Social Psychology, Vol. 51*, 629-636.

Dia, H. (2002). An Agent-Based Approach to Modelling Driver Route Choice Behaviour Under the Influence of Real-Time Information. *Transportation Research Part C, 10*, 331-349.

Drury, J., Cocking, C., Reicher, S., Burton, A., Schofield, D., Hardwick, A., & Langston, P. (2009). Cooperation Versus Competition in a Mass Emergency Evacuation: A new Laboratory Simulation and a new Theoretical Model. *Behavior Research Methods, 41(3)*, 957-970.

Ehtamo, H., Heliövaara, S., Hostikka, S., & Korhonen, T. (2008). *Modelling Evacuees' Exit Selection with Best Response Dynamics.* Springer.

Ehtamo, H., Heliövaara, S., Korhonen, T., & Hostikka, S. (2010). Game Theoretic Best-Response Dynamics for Evacuees' Exit Selection. *Advances in Complex Systems, Vol. 13, No. 1*, 113-134.

Firaxis Games. (2016, October 7). *Information, Civilization V*. Retrieved from Civilization V: www.civilization5.com

Fire Safety Engineering Group. (2016, November 16). *Exodus Introduction*. Retrieved from Fire Safety Engineering Group - The Faculty of Architecture, Computing & Humanities, University of Greenwich: http://fseg.gre.ac.uk/exodus/

Frejinger, E., & Bierlaire, M. (2007). *Random Sampling of Alternative for Route Choice Modeling.* Swiss Transport Research Conference.

Gales, E., Lawrence, p., Gwynne, S., Filippidis, L., Blackshields, D., & Cooney, D. (2006). *BuildingEXODUS v4.06 User Guide and Technical Manual.* Fire Safety Engineering Group, London U.K: University of Greenwich.

Georgoudas, I., Sirakoulis, G., & Andreadis, I. (2008). Hardware Implementation of a Crowd Evacuation Model Based on Cellular Automata. In W. W. Klingsch, C. Rogsch, A. Schadschneide, & M. Schreckenberg, *Pedestrian and Evacuation Dynamics 2008* (pp. 451-463). Department of Electrical and Computer Engineering, Democritus Univeristy of Thrace, 67100 Xanthi, Greece.

Gwynne, S., & Rosenbaum, E. (2016). Employing the Hydraulic Model in Assessing Emergency Movement. In M. J. Hurley, D. T. Gottuk, J. R. Hall Jr, K. Harada, E. D. Kuligowski, M. Puchovsky, . . . C. J. Wieczorek, *SFPE Handbook of Fire Protection Engineering* (pp. 2115-2151). Springer New York.

Gwynne, S., Galea, E., Lawrence, P., Owen, M., & Filippidis, L. (2000). Adaptive Decision-Making in BuildingEXODUS in Response to Exit Congestion. *Fire Safety Science - Proceedings of the Sixth International Symposium*, 1041-1052.

Gwynne, S., Galea, E., Owen, M., Lawrence, P., & Filippidis, L. (2004). *A Systematic Comparison of buildingEXODUS Predictions with Experimental Data from the Staplefeldt Trials and the Milburn House Evacuation.* Greenwich: University of Greenwich.

Hamacher, H., & Tjandra, S. (2001). *Mathematical Modelling of Evacuation Problems: A State of Art.* Fraunhofer: Fraunhofer Institute for Industrial Mathematics ITWM.

Hart, P., Nilsson, N., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In *IEEE Transactions on Systems Science and Cybernetics* (pp. 100-107).

Henn, V. (1998). *Fuzzy Route Choice Model for Traffic Assignment.* Fuzzy Sets and Systems.

Horiuchi, S., Murozaki, Y., & Hukugo, A. (1986). A Case Study of Fire and Evacuation In a Multi-purpose Office Building, Osaka, Japan. In *Fire Safety Science 1* (pp. 523-532).

IES. (2000). *Simulex Technical Reference; Evacuation Modelling Software.* Integrated Environmental Solutions, Inc. Generic.

International Maritime Organization. (2007). *Guidelines for Evacuation Analysis for New and Existing Passenger Ships.*

International Standards Organization. (2015). *Safety Engineering - Assessment, verification and validation of calculation methods. ISO 16730-1.*

Kinateder, M. (2013). *Social Influence in Emergency Situatuib-Studies in Virtual Reality. Phd Dissertation.* University of Würzburg.

Kinsey, M. (2011). *Vertical Transport Evacuation Modelling.* Greenwich, London, UK.

Kisko, T., Francis, R., & Nobel, C. (1998). *EVACNET4 USER'S GUIDE.* University of Florida.

Koenig, S., & Likhachev, M. (2002). *D* Lite.*

Koenig, S., Likhachev, M., & Furcy, D. (2005). *Lifelong Planning A*.*

Koppelman, F., & Bhat, C. (2006). *A Self Instructing Course in Mode Choice Modeling: Multinomial and Nested Logit Models.*

Koppelman, F., & Wen, C. (2000). The Paired Combinatorial Logit Model: Propertiesm Estimation and Application. *Transportation Reserach Part B: Methodological, Vol. 34 (2)*, 75-89.

Korf, R. (1985). Depth-First Iterative-Deepening: An Optimal Admissible Tree Search. In *Artificial Intelligence 27* (pp. 97-109). Elsevier Science Publishers B.V.

Korf, R. (1990). Real-Time Heuristic Search. In *Artificial Intelligence, vol. 42* (pp. 189-211).

Korhonen, T. (2015). *Fire Dynamics Simulator with Evacuation: FDS+Evac.* VTT Technical Research Centre of Finland.

Kuligowski, E. (2016). Computer Evacuation Models for Buildings. In M. J. Hurley, D. T. Gottuk, J. R. Hall Jr, K. Harada, E. D. Kuligowski, M. Puchovsky, . . . C. J. Wieczorek, *SFPE Handbook of Fire Protection Engineering* (pp. 2152-2180). New York: Springer New York.

Latané, B., & Darley, J. (1970). *The Unresponsive Bystander: why doesn't he help?* Appleton-Century Crofts, New York.

Li, Q., Fang, Z., Li, Q., & Zong, X. (2010). *Multiobjective Evacuatinon Route Assignment Model Based on Genetic Algorithm.*

Li, X., & Wu, Y. (2016). Numerical Simulation of the Propagation of Hydraulic and Natural Fracture Using Dijkstra's Algorithm. *Energies 9 (7), 519*.

Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., & Thrun, S. (2005). *Anytime Dynamic A*: An Anytime, Replanning Algorithm.*

Likhachev, M., Gordon, G., & Thrun, S. (2003). *ARA*: Anytime A* with Provable Bounds on Sub-Optimality.* Pittsburgh: Scool of Computer Sciende, Carnegie Mellon University.

Lo, S., Huang, H., Wang, P., & Yuen, K. (2006). A Game Theory Based Exit Selection Model for Evacuation. *Fire Safety Journal 41*, 364-369.

Luce, R. (1959). *Induvidual Choice Behaviour: A Theoretical Analysis.* New York: Wiley.

McFadden, D. (1976). Quantal Choice Analysis: A Survey. *Annals of Economic and Social Measurement, 5*, 363-390.

Microsoft Game Studios. (2016, October 7). *Games, Age of Empires*. Retrieved from Age of Empires: https://www.ageofempires.com/

Misa, T. (2010, August). An Interview with Edsger W. Dijkstra. *Viewpoints*, pp. 41-47.

Mott MacDonald Group Limited. (2016, November 16). *About Steps*. Retrieved from Mott MacDonald: http://www.steps.mottmac.com/aboutsteps2/

Moussaïd, M., Helbing, D., & Theraulaz, G. (2011). How Simple Rules Determine Pedestrian Behavior and Crowd Disasters. *Proceedings of the National Academy of Sciences, 108(17)*, 6884-6888.

Nilsson, D., & Johansson, A. (2009). Social Influence During the Initial Phase of a Fire Evacuation - Analysis of Evacuation Experiments in a Cinema Theatre. *Fire Safety Journal, 44(1)*, 71-79.

Novák, V., Perfilieva, I., & Močkoř, J. (1999). *Mathematical Principles of Fuzzy Logic* . Kluwer Academic Publishers.

Olander, J., Ronchi, E., Lovreglio, R., & Nilsson, D. (2017). Dissuasive Exit Signage for Building Fire Evacuation. *Applied Ergonomics, 59*, 84-93.

Oxford Dictionaries. (2016, October 10). Retrieved from Oxford Dictionary: https://en.oxforddictionaries.com/definition/line_of_sight

Prato, C. (2009). *Route choice modelling: past, present and future research directions.* Haifa: Journal of Choice Modelling.

Prato, C. (2009). Route Choice Modelling: Past, Present and Future Research Directions. *Journal of Choice Modelling 2(1)*, 65-100.

Prato, C. (2009). Route Choice Modelling: Past, Present and Future Research Directions. *Journal of Choice Modelling 2(1)*, 65-100.

Purser, D. (2003). ASET and RSET: Addressing Some Issues in Relation to Occupant Behaviour and Tenability. *FIre Safety Science - Proceedings of the Seventh International Symposium*, 91-102.

Ramming, M. (2002). *Network Knowledge and Route Choice.* Massachusetts: Department of Civil and Environmental Engineering, Massachesetts Institute of Technology.

RiMEA e. V. (2016). *Guideline for Microscopic Evacuation Analysis.*

Ronchi, E. (2014). *The Need for a Verification and Validation Protocol for Evacuation Models.* Lund: Department of Fire Safety Engineering.

Ronchi, E., & Kinsey, M. (2011). *Evacuation Models of the Future: Insight Fram an Online Survey of User's Experiences and Needs.* Santander, Spain: Advanced Research Workshop - Evacuation and Human Behaviour in Emergency Situations.

Ronchi, E., & Nilsson, D. (2016). Basic Concepts and Modelling Methods. In A. Cuesta, O. Abreu, & D. Alvear, *Evacuation Modelling Trends* (pp. 1-23). Springer International Publishing.

Ronchi, E., Kuligowski, E., Reneke, P., Peacock, R., & Nilsson, D. (2013). *The Process of Verification and Validation of Building Fire Evacuation Models.* National Institute of Standards and Technology.

Ronchi, E., Nilsson, D., Modig, H., & Walter, A. (2016). Variable Message Signs for Road Tunnel Emergency Evacuations. *Applied Ergonomics, 52*, 253-264.

Santos, G., & Aguirre, B. (2004). *A Critical Review of Emergency Evacuation Simulation Models.* Delaware: University of Delaware.

SFPE. (2007). *SFPE Engineering Guide to Performance-Based Fire Protection.* Quincy, MA: National Fire Protection Association.

Sheffi, Y., & Powell, W. (1982). An Algorithm for the Equilibrium Assignment Problem with Random Link Times. *Networks, 12*, 191-207.

Sime, J. (1985). Movement Toward the Familiar: Person and Place Affiliation in a FIre Entrapment Setting. *Environment and Behavior 17*, 697-724.

Stout, B. (1997). *Smart Moves: Intelligent Pathfinding.* Game Developer Magazine.

Thompson, P. (1994). *Developing New Techniques for Modelling Crowd Movement.* Edinburgh: University of Edinburgh.

Thompson, P., & Marchant, E. (1994). Simulex; Developing New Computer Modelling Techniques for Evaluation. *Fire Safety Science - Proceedings of the Fourth International Symposium*, 613-624.

Thompson, P., Wu, J., & Marchant, E. (1997). Simulex 3.0: Modelling Evacuation in Multi-Storey Buildings. *Fire Safety Science 5*, 725-736.

Thornton, C., O'Konski, R., Klein, B., Hardeman, B., & Swenson, D. (2012). *New Wayfinding Techniques in Pathfinder and Supporting Research.* Manhattan, USA: Thunderhead Engineering.

Thunderhead. (2016). *Pathfinder - Technical Reference.* Manhattan: Thunderhead engineering.

Thunderhead. (2016). *Pathfinder - Verification and Validation.* Manhattan: Thunderhead Engineering.

Tong, D., & Canter, D. (1985). The Decision to Evacuate: a study of the Movement which Contribute to Evacuation in the Event of Fire. *Fire Safety Journal 9*, 257-265.

Tozour, P. (2002). Building a Near-Optimal Navigation Mesh. In S. Rabin, *AI Game Programming Wisdom* (pp. 171-185). Charles River Media, Inc.

Waterson, N., & Pellissier, E. (2010). *The STEPS Pedestrian Micorsimulation Tool - A Technical Summary.* Croydon, UK.: Mott MacDonald Limited.

Wen, C., & Koppelman, F. (2001). The Generalized Nested Logit Model. *Transportation Research Part B, 35(7)*, 627-641.

Winston, W. (2004). *Operations Research: Applications and Algorithms.* Indiana University, Kelley School of Business: Cengage Learning India Pvt Ltd.

Vovsha, P. (1997). *Application of Cross-Nested Logit Model to Mode Choice in Tel-Aviv, Israel, Metropolitan Area.*

Xie, H. (2011). *Investigation into the Interaction of People with Signage Systems and its Implementation within Evacuation Models.* Greenwich: University of Greenwich.

Zadeh, L. (1965). Fuzzy Sets. In *Information and Control 8* (pp. 338-353).