FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Sensitivity analysis of models of cramped interior evacuation
Student:	Bc. Juraj Kmec
Supervisor:	Ing. Pavel Hrabák, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2023/2024

Instructions

A common tool supporting a fire-safety analysis is a multi-agent simulation of the evacuation from dedicated premises. An example of such a tool is a commonly used Pathfinder software. The main goal of the thesis is to perform a variance sensitivity analysis of key parameters and features of this tool with a focus on evacuation from cramped interiors with initial positions defined by seats, such as rail-car or lecture hall. The thesis should extend the results from Najmanová et al. (2022).

1. Survey suitable tools for sensitivity analysis and studies related to sensitivity analysis of evacuation simulations.
2. Create a script/pipeline enabling repetitive launch of Pathfinder simulation using the simulation input-file, while controlling selected parameters of the model, in the way that the outputs can be used as inputs for selected tools for sensitivity analysis.
3. Based on consultations with researchers from the Faculty of Civil Engineering select evacuation scenarios suitable for the sensitivity analysis, perform the analysis and compare the results.
4. By means of above-mentioned tools analyse various sources of randomness in the simulations (e.g. initial positions, agent parameters, inherent randomness) and investigate their impact to uncertainty in total evacuation time.
5. Using the simulations build a surrogate/regression model of the total evacuation time from selected scenarios. Compare the results of sensitivity analysis based on this model with the results based directly from simulations.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 21 November 2022 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master’s thesis

Sensitivity Analysis of Models of Cramped Interior Evacuation

Bc. Juraj Kmec

Department of Applied Mathematics
Supervisor: Ing. Pavel Hrabák, Ph.D.

May 4, 2023

Acknowledgements

I would like to express sincere gratitude to my supervisor, Ing. Pavel Hrabák, Ph.D., for actively pushing me in the right direction, guiding me through all parts of this thesis, and encouraging me to keep on trying. I would also like to thank the rest of the evacuation-dynamics research staff at CTU, namely Ing. Daniel Vašata, Ph.D., and Ing. Hana Najmanová, Ph.D., for continuously providing me with new thoughts and ideas. And last but not least, my family and friends for providing the much-needed emotional support.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 4, 2023 . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Kmec, Juraj. *Sensitivity Analysis of Models of Cramped Interior Evacuation.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Abstrakt

Hlavním cílem práce je identifikovat proměnné, které mají největší vlyv na celkový čas evakuace v rámci simulovaných evakuačních scénářů. Zkoumány jsou dvě kontrastní geometrie – dvoupatrový vlak a přednáškový sál. Pro každou z těchto geometrií jsou zohledněny dva různé scénáře – prostředí s nízkou hustotou, kde je objekt přibližně na 50 % naplněný, a prostředí s vysokou hustotou, kde je objekt naplněn na 90 % jeho kapacity. Tyto scénáře jsou analyzovány pomocí technik citlivostní analýzy se zvláštním důrazem na jednotlivé zdroje náhodnosti ve výstupu – parametry chodců, jejich počáteční pozice a inherentní náhodnost simulačního procesu. Davy v experimentech jsou tvořeny dvěma skupinami – skupinou fyzicky zdatných osob, která nemá žádné pohybové omezení, a skupinou s fyzickými znevýhodněními simulovanými pomalejší rychlostí a delším časem akcelerace. Výsledky ukazují, že při nižších hustotách hrají při určování celkové doby evakuace rozhodující roli počáteční pozice okupantů, ale se zvyšováním hustoty přebírají tuto roli makroskopické charakteristiky davu.

Klíčová slova Simulovaná evakuace, citlivostní analýza, heterogenní dav, náhradní modelování, Sobolovi citlivostní indexy, elementární efekty.

Abstract

The thesis' main goal is to identify the most influential variables in determining the total evacuation time of a simulated egress scenario. Two contrasting geometries are investigated – a double-decker train and a lecture hall. For each of these geometries, two distinct scenarios are considered – a low-density setting where the facility is at approximately 50 % capacity, and a high-density setting where the facility is at 90 % capacity. These scenarios are analyzed by means of sensitivity analysis with a special focus on the individual sources of randomness in the output – the occupant parameters, their initial positions, and the inherent uncertainty of the simulation process. The crowds in the experiments are made up of two groups of occupants – an able-bodied group with no movement limitations, and a movement-impaired group simulated by a slower maximum speed and a higher acceleration time. The results show that at lower densities, the initial positions of the occupants play a crucial role in determining the total evacuation time, but at higher densities, the macroscopic characteristics of the crowd take over as the most important factors.

Keywords Simulated evacuation, sensitivity analysis, heterogeneous crowd, surrogate modeling, Sobol' sensitivity indices, elementary effects.

Contents

List of Figures

List of Tables

Introduction

The study of evacuation scenarios and the process of safely evacuating a building or other premise has been a matter of active research for decades and remains a challenge among architects and safety engineers. Indeed, when every extra second could mean the difference between a successful evacuation and a disaster, every detail must be optimized. Of course, some factors such as the geometrical layout of the rooms (once they've been built) are fixed and out of our reach. What we can influence, for example, is the seating of the occupants where putting the slowest and/or most crucial evacuees too close to the exit or too far from the exit will have a drastic effect on the total evacuation time.

To determine which factors have the largest impact on making an evacuation scenario fast and safe, purely theoretical methods are insufficient. Therefore, it is necessary to explore various configurations and observe which factors have the most statistically significant effect. For practical as well as ethical reasons, running live experiments for every configuration we think of is infeasible. Instead, we should form our hypotheses and obtain preliminary results via simulation software. We should then design an actual experiment to validate our model and confirm or deny our hypotheses. We can then improve our model to match the true behavior more closely and refine our claims. This thesis will focus on the first part of this process and set up the grounds for a practical experiment to further verify its claims.

More specifically, the main aim of this thesis is to use the techniques of *sensitivity analysis* to numerically quantify the effects of various factors concerning the initial conditions of a simulated evacuation. Namely, I shall focus on the *maximum velocities* and *acceleration times* for two occupant groups – a group of *able-bodied* occupants with no movement impairment and a group of occupants *with limitations* whose movement is impaired by a lower free-movement velocity and higher acceleration time. Additionally, the *initial positions (seating)* of the occupants play a major role in determining the total evacuation time and the analysis will pay special attention to this phenomenon. Since there is no immediately apparent way to numerically describe a seating configuration, for this thesis I shall employ the use of a probabilistic way to generate the initial positions dictated by their *attraction to the exit* and *inter-pedestrian attraction*, for both the able-bodied group and the with-limitations group. Lastly, the effect that the *ratio of the number of occupants in each group* has on the evacuation time will be explored, for a total of 9 variables. The simulation software of choice for this thesis is Pathfinder – an industry-grade pedestrian simulation tool developed by Thunderhead Engineering.

Structure

This thesis is an open continuation of my Bachelor's thesis, which focused on the estimation of fundamental diagrams from evacuation experiments [2]. The thesis is structured as follows: The first three chapters focus on the theoretical framework for the thesis. The first chapter provides a brief overview of the relevant concepts of pedestrian dynamics as it is not a part of the faculty's standard curriculum. The second chapter follows with a discussion of Pathfinder's simulation model and how it is implemented. The third chapter presents the theoretical foundation of sensitivity analysis and all the methods employed. The fourth chapter opens up the practical part of the thesis with a discussion of the chosen evacuation scenarios and their implementation in Pathfinder. The fifth chapter describes the Python implementation of the simulation pipeline which was used to obtain the necessary data. The sixth chapter is the experimental chapter which contains a detailed description of the simulation scenarios and their results, as well the application of surrogate modeling to obtain more robust estimates. The thesis concludes with a summary of the results and a discussion of their contributions to the topic. Potential for further research is also highlighted.

Chapter 1

Pedestrian Dynamics

This chapter aims to introduce the relevant concepts for pedestrian dynamics. The main emphasis will be on the simulation model that is used by Pathfinder[1] as it is the main tool that will later be used to collect data for the subsequent analysis. Before diving deeper into the simulation model itself, I will define the basic measurable variables with special attention to their Pathfinder implementation and the concept of *fundamental diagram.* A more comprehensive study can be found in [3] as well as in my bachelor's thesis [2], which explored the concept of the fundamental diagram in more detail.

1.1 Measurable Variables

The three main variables that are used to describe the movement of a crowd of pedestrians on a macroscopic level are *velocity*, *density*, and *flow.* These are all quantities taken directly from the field of traffic engineering, which are in turn inspired by their counterparts in fluid dynamics. For this chapter, I shall adopt the standard notation from physics where a vector is denoted by an overhead arrow "$\rightarrow$", e.g., $\vec{v_i}(t)$ meaning the velocity of pedestrian i at time t. Temporal averages will be denoted by $\langle \cdot \rangle_{\Delta t}$ and spatial averages will be denoted by $\langle \cdot \rangle_{\Delta x}$ as is common in related literature such as [3].

1.1.1 Velocity

The (immediate) velocity of occupant i is the first derivative of the position with respect to time, i.e., $\vec{v_i}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\vec{x_i}(t)$. In practice, this vector is approximated by a discretized derivative at time t and a small time step Δt, i.e.,

$$\vec{v_i}(t) = \frac{\vec{x_i}(t + \Delta t) - \vec{x_i}(t)}{\Delta t}, \tag{1.1}$$

[1] `https://www.thunderheadeng.com/pathfinder`

where Δt could correspond to, for example, a couple of frames in video footage. An alternative way to measure the (average) velocity of a pedestrian over a given measurement region is taken from traffic engineering and involves dividing the distance between the entry point and exit point by the time spent in the measurement region and interpolating, i.e.,

$$\vec{v_i}(t) = \frac{\vec{x_i}(t_{\text{out}}) - \vec{x_i}(t_{\text{in}})}{t_{\text{out}} - t_{\text{in}}}, \quad t \in [t_{\text{in}}, t_{\text{out}}]. \tag{1.2}$$

This method, however, will generally underestimate the true velocities because it assumes that the pedestrian has traveled in a straight line, which is often not the case. This problem could be solved by extracting their exact trajectory, but this can be a difficult problem by itself.

So far, these were *microscopic* definitions, which only concern a single pedestrian. It is important to distinguish them from *macroscopic* variables, which describe the group of pedestrians as a whole. The *spatial average velocity* over a given measurement area containing N pedestrians is simply the arithmetic mean of their velocities at time t:

$$v(t) = \langle v \rangle_{\Delta x}(t) = \frac{1}{N} \sum_{i=1}^{N} \|\vec{v_i}(t)\|. \tag{1.3}$$

Note that we are only interested in the norms of the individual velocities (speeds). The temporal average velocity over a given measurement line and a period T is defined analogously:

$$v(T) = \langle v \rangle_{\Delta t}(T) = \frac{1}{N} \sum_{i=1}^{N} \|\vec{v_i}(t_i)\|, \tag{1.4}$$

where $t_i \in T$ is the time at which pedestrian i crossed the measurement line.

Velocity is closely related to *acceleration*, which is its first derivative with respect to time. It can be measured via discretized derivatives similarly to velocity or via specialized gyroscopic sensors, but it is somewhat impractical due to the high costs and long setup time. However, Pathfinder explicitly uses an *acceleration time* parameter, which is the time it takes for a pedestrian to reach their maximum free-flow velocity, to compute an occupant's immediate velocity. I will describe this in more detail in a later section.

1.1.2 Density

The density ρ of a pedestrian group in a given area A at time t is the number of pedestrians N per unit area:

$$\rho(t) = \langle \rho \rangle_{\Delta x}(t) = \frac{N(t)}{|A|}, \tag{1.5}$$

where $|A|$ is the size of area A. The associated unit is $\mathrm{P/m^2}$ (persons per square meter). In three-dimensional cases, only the floor projection of the space is considered. This quantity is macroscopic by default as it depends on an entire area of interest. This also gives rise to several undesirable properties such as convergence to zero when considering smaller and smaller measurement regions and no clear way to look at per-pedestrian local densities.

For this reason, there exist several modifications of this definition such as *Voronoi density* [4] or various *local densities* based on the distances to k nearest neighbors. Pathfinder uses a modified 1-nearest neighbor local density for each occupant in order to adjust the velocity computation to also take into account the agent's immediate surroundings.

Pathfinder's technical reference [5] is not clear on the actual numerical values used, however, conceptually, the calculation works in the following way: The local density values are based on a study of average longitudinal and lateral spacing of pedestrians [6] whose results are shown in Figure 1.1 where each contour symbolizes a perceived level of local density. Each occupant's location is transformed into a local coordinate system with the origin at the occupant's center and their current facing direction is aligned with the x-axis in Figure 1.1. The density contours are approximated with ellipses whose major axis matches the occupant's velocity vector so that occupants that are directly in front of have less of an effect at the perceived local density. Then, the nearby occupants in a given radius are retrieved and for each of them, a local density value is computed by interpolating or extrapolating from the density ellipses. Additionally, all the nearby occupants with $x < 0$ in the local coordinate system are ignored so that the pedestrians directly behind do not affect the perceived density. Finally, the resulting density is taken as the maximum of the partial densities from each nearby occupant, effectively proportional to a 1-nearest neighbor distance in a modified coordinate system. This value is then used to lower each occupant's desired speed in situations with high local densities (crowds) as will be described in Section 2.3.1.

1.1.3 Flow

The final fundamental characteristic of pedestrian dynamics is the flow J. It is most often defined as the number of pedestrians N crossing a given measurement line per unit time. Due to the discrete nature of pedestrian flow, this quantity has to be estimated using average measurements over a time period T:

$$J(T) = \langle J \rangle_{\Delta t}(T) = \frac{N}{|T|}. \tag{1.6}$$

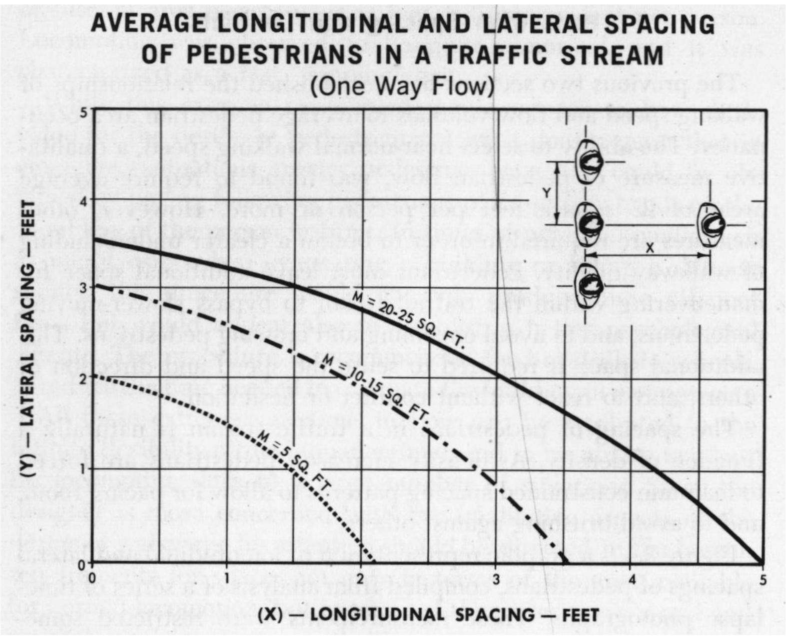


Figure 1.1: Density contours based on longitudinal and lateral spacing. (Figure 3.4 from [6])

An alternative estimate can be obtained as the inverse of the average time gap between consecutive pedestrians crossing the measurement line.

$$J(T) = \left(\frac{1}{N-1} \sum_{i=1}^{N-1} (t_{i+1} - t_i) \right)^{-1}, \tag{1.7}$$

where $t_i \in T$ is the time at which occupant i crossed the measurement line. The *specific flow* J_s is another derived quantity concerning a measurement line L which is defined as the flow per unit width,

$$J_s(L) = \frac{J(L)}{|L|}. \tag{1.8}$$

The associated units are either P/s (persons per second) or P/(m · s) (persons per meter per second). This characteristic is often used as a parameter in the simulation setup when we want to signalize that some doors are “harder to cross” than others and cause an increase or reduction in flow.

If we instead wanted to compute the instantaneous flow in an area A at time t, we can obtain this quantity via the hydrodynamic-inspired relation $J = \rho v$ (called the *fundamental equation of pedestrian flow*) using the spatial averages of both the velocity and density [3].

$$J(t) = \langle J \rangle_{\Delta x}(t) = v(t)\rho(t). \tag{1.9}$$

Note that the information about the direction of the flow gets lost this way.

In Pathfinder, doors are generally used as measurement lines and the flow rate through doors plays a major part in determining the total evacuation time.

1.2 Fundamental Diagram

The aforementioned fundamental equation of pedestrian flow is a characteristic that describes the link between macroscopic velocity, density, and flow via the equation $J = \rho v$ and is often called the *fundamental diagram of pedestrian flow*. It is usually expressed as a graph of the relation between either flow and density, velocity and density, or flow and velocity. All of these forms are equivalent and one can freely move between the different representations via the fundamental equation. An example of a ρ–J fundamental diagram is shown in Figure 1.2.

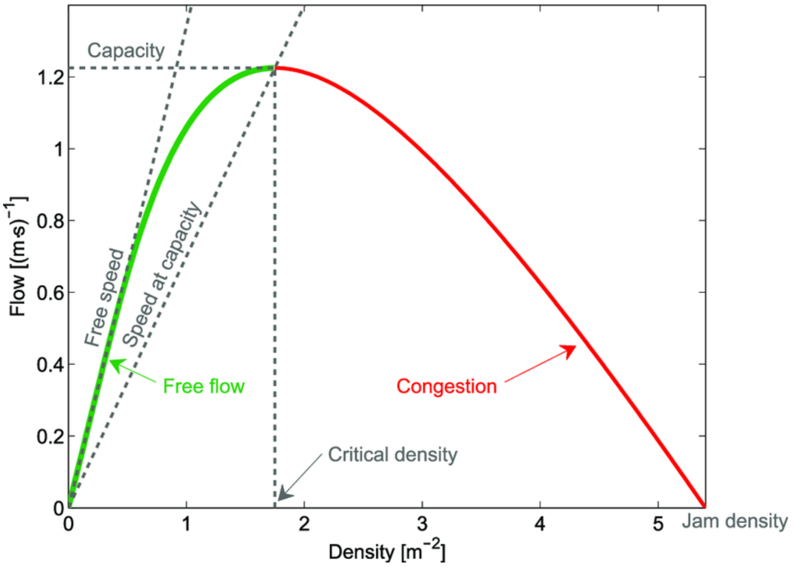


Figure 1.2: Fundamental diagram of flow as a function of density.

(Figure 1 from [7])

The diagram captures various natural expectations such as that if we keep increasing the density in a room then the flow will increase as well but only up to a certain point (critical density) at which congestion takes place and the flow will start decreasing until it completely stops (jam density). The fundamental diagram has many more interesting properties such as the fact that the slope of the graph with respect to the origin at a given density is, in fact, the spatial average velocity corresponding to said density. A more comprehensive study can be found in [8].

Although the qualitative shape of the diagram is generally agreed upon, there are still disagreements about the actual numerical values at which the important points in the diagram are located, which can be attributed, for example, to different facility types or even cultural differences [9, 10].

Perhaps the most commonly cited analytical approximation of pedestrian fundamental diagram is due to Weidmann [11], who expressed velocity as a function of density in the following way:

$$v(\rho) = v_0 \left[1 - \exp\left(-\rho_0 \left(\frac{1}{\rho} - \frac{1}{\rho_{\max}}\right)\right)\right], \tag{1.10}$$

where $v_0 = 1.34\,\mathrm{m/s}$ is the free-flow walking speed, $\rho_0 = 1.934\,\mathrm{P/m^2}$ is the base density, and $\rho_{\max} = 5.4\,\mathrm{P/m^2}$ is the jam density. The $J_s(\rho)$ form can then be obtained as $J_s(\rho) = \rho \cdot v(\rho)$. Other noteworthy forms are due to Predtechenskii and Milinskii [12] and Fruin [6]. A comparison along with experimental data is shown in Figure 1.3.

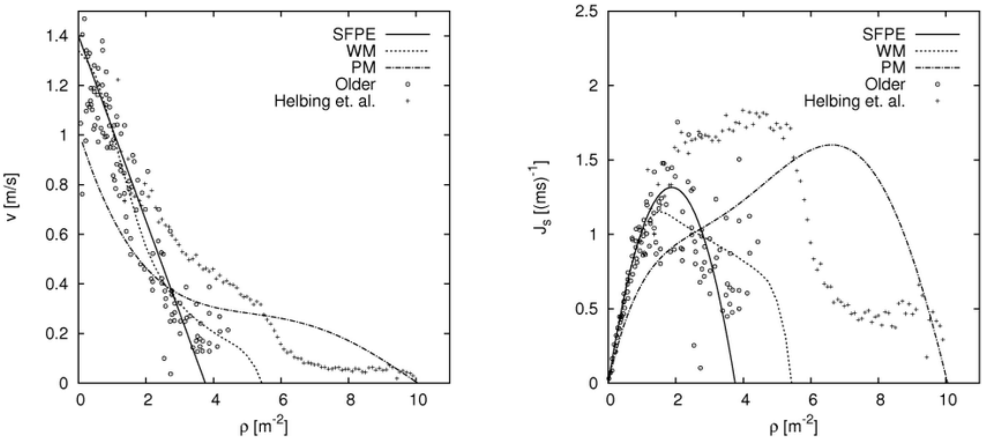


Figure 1.3: Comparison of different analytical and empirical FDs.

(Figure 2 from [13])

One that is of particular concern here is the fundamental diagram provided by the SFPE handbook of fire protection engineering [14]. SFPE stands for the Society of Fire Protection Engineers and their handbook contains a comprehensive overview of various topics surrounding fire protection engineering. The chapter "Egress Concepts and Design Approaches" concerns itself with evacuation dynamics and Pathfinder makes heavy use of its concepts and calculations in its implementation. The proposed $v(\rho)$ curve has the following form:

$$v(\rho) = \begin{cases} v_0 & \text{if } \rho \in [0, \rho_0], \\ v_{\max}(1 - \rho/\rho_{\max}) & \text{if } \rho \in [\rho_0, \rho_{\max}], \end{cases} \tag{1.11}$$

with the parameters maximum free-flow walking speed $v_0 = 1.19\,\mathrm{m/s}$, bounding velocity $v_{\max} = 1.40\,\mathrm{m/s}$, density $\rho_0 = 0.54\,\mathrm{P/m^2}$ at which the average speed starts to decrease, and density $\rho_{\max} = 3.76\,\mathrm{P/m^2}$ at which the flow completely stops. The graph is a piece-wise linear function that's constant at densities lower than ρ_0 (movement is not impaired by nearby occupants) and then starts to linearly decrease until it hits zero at $\rho = \rho_{\max}$. The corresponding $J_s(\rho)$ function hast the form

$$J_s(\rho) = \rho v_{\max} \left(1 - \rho/\rho_{\max}\right) \tag{1.12}$$

which is a parabola with the maximum flow at the critical density $\rho_c = 1.88\,\mathrm{P/m^2}$ (the constant part at low densities is discarded). Both of these graphs can be seen in Figure 1.3, although the first graph is missing the constant-velocity piece.

Chapter 2

Pathfinder Simulation Model

Now that we have defined the essential concepts of pedestrian dynamics, we can move to the more practical part of simulating pedestrian movement. There is a vast number of distinct competing approaches to modeling human locomotion, most of which can be classified as *macroscopic*, in which the individual pedestrians are modeled as particles guided by laws of fluid dynamics, or *microscopic* (including Pathfinder), in which each pedestrian is an individual agent. In microscopic models, each agent has to concern themselves with goals on three distinct levels: *strategic* level – activity planning, *tactical* level – scheduling and route choice, and *operational* level – walking behavior [3]. The next sections will briefly describe the methods of simulating each level and their implementations in Pathfinder. The implementation details that follow are based on the descriptions in Pathfinder technical reference [5].

Each occupant is assigned a *profile*. Profiles define the physical characteristics of an agent as well as agent metadata. This includes properties such as maximum velocity, shape, diameter, or height, but also more complex traits such as how much an occupant can "squeeze" themselves in order to fit in narrow geometry or how much their velocity reduces in high-density situations. Each property is defined on a profile level, but it can be overwritten on a per-occupant basis. The ones that will be of particular concern here are the following:

Maximum velocity – The agent's desired speed when their movement is not impaired by terrain or other agents.

Acceleration time – The time it takes for an agent to reach their maximum velocity from a complete stop.

Priority – A value that determines which agents are given higher priority when resolving movement conflicts.

Initial orientation – The direction which an occupant is facing at the beginning of the simulation.

In the context of this thesis, the former two are considered the agents' deterministic parameters used to group the occupants into an able-bodied and a with-limitations group, while the latter is considered the random parameters of the model and re-generating them will lead to different evacuation times despite the physical characteristics of the occupants being the same. There is also a random seed parameter `rseed` which is also treated as a strictly random parameter, but the documentation is not clear on its exact effect.

2.1 Strategic Level

Making decisions on a strategic level answers questions such as "What will I do next?", "Where should I go?", or "Should I stay in this queue or move to a different one?". The answer is simple when there the goal is evacuation and there is only a single exit, but becomes non-trivial when there are multiple points of interest such as queues in a shop or spaces in a parking lot. The classical methods for decision-making include *regression models*, *spatial interaction models*, or *Markovian models* [15]. Modern methods also make use of concepts from psychology [16].

Pathfinder's implementation of strategical-level decision-making is based on the concept of *goals* and *behaviors*. An occupant's behavior is defined by a sequence of goals that the occupant has to reach in order to exit the simulation. The behavior can be as simple as "Go to any exit", which is the one that will be used in the analyses in this thesis. An example of a more complex behavior is, for example, simulating a jump from a platform by instructing the occupant to wait in front of the platform for a given time delay before actually making the (instant) transition between the z coordinates as was shown in [1]. When an appropriate seek goal has been selected, path-finding can take place.

2.2 Tactical Level

Pedestrian behavior on a tactical level consists of constructing an appropriate path to the destination of choice. There, once again, exist a large number of different methods, but the ones most often used are either *field-based*, *graph-based*, or *hybrid*. The following is a short summary based on [17].

In field-based methods, each point in the geometry is assigned a scalar value that depends on the distance towards the point of interest or travel time therein. A path is then generated by minimizing the field value over the course of the trajectory, either in a deterministic or a probabilistic way. The path is periodically recomputed in order to account for other occupants in the area who may or may not influence the agent's trajectory.

In graph-based methods, which are also implemented in Pathfinder, a static graph is generated using "checkpoints" like doors or room boundaries, and the navigation mesh of the geometry is traversed using graph search algorithms such as Dijkstra's algorithm or A* algorithm.

In practice, most methods use a combination of both approaches. For example, in presence of multiple paths to target in a field-based method, a graph search using coarse checkpoints is done first in order to determine which general route to choose. Similarly, in a graph-based setting, scalar costs mimicking a field-based model can be assigned to each edge in the graph before starting a search.

In Pathfinder, the pathfinding algorithm first begins by *path planning*, that is, when there are multiple distinct paths to the seek goal, a specific one has to be chosen, as the shortest one is not necessarily the optimal one due to the presence of other occupants along the way. The method used in Pathfinder is called *locally quickest*, in which a route is planned hierarchically using local information about the current room and global information about the full geometry. The process works as follows:

1. A list of local targets is generated that includes the available doors in the current room and the seek goal itself if it is located in the current room.

2. The next local target is chosen by minimizing a weighted sum of different factors concerning the target.

3. The chosen target is followed using *path generation* and *path following* algorithms.

The weighted sum mentioned in the second step takes into account the following quantities:

Current room travel time t_{lt} – The time it would take to reach the target while traveling at full speed.

Current room queue time t_q – If the target is a door, this quantity is estimated using the number of occupants waiting in the queue and the flow rate of the door, otherwise it is set to 0.

Global travel time t_{gt} – The time it would take to get from the target to the current seek goal traveling at full speed, also taking into consideration other modifiers such as stairs or different terrain along the way.

Distance travelled in room d_t – The distance already travelled in the current room.

Each of these components has an associated cost factor k_{lt}, k_q, k_{gt}, and k_{dd}, along with a current door preference coefficient p. The total cost for the i-th target is computed as

$$C_i = C_{\text{local}} + C_{\text{global}} \tag{2.1}$$

with

$$\begin{aligned} C_{\text{local}} &= \max(p_d k_{lt} t_{lt}, k_{qh} k_q t_q), \\ C_{\text{global}} &= p_d k_{gt} t_{gt}, \\ p_d &= \exp(k_d d_t), \\ k_d &= \frac{\log 2}{k_{dd}}, \end{aligned}$$

and $k_{qh} = 1$ for the most recently chosen target and $1 - p$ otherwise. A target door is chosen once an occupant enters a room and then recomputed periodically every second. A backtrack prevention mechanism is also present so that agents do not circle back and forth between nearby rooms.

Once the next target is chosen, a concrete path can be generated. Pathfinder employs a graph search procedure over the navigational mesh of the geometry using the A* algorithm. Since a path obtained this way can be needlessly jagged, it is additionally smoothed so that only the edges at which an actual change in direction happens remain. An example of a generated path is shown in Figure 2.1. The nodes along the path are called “waypoints”.

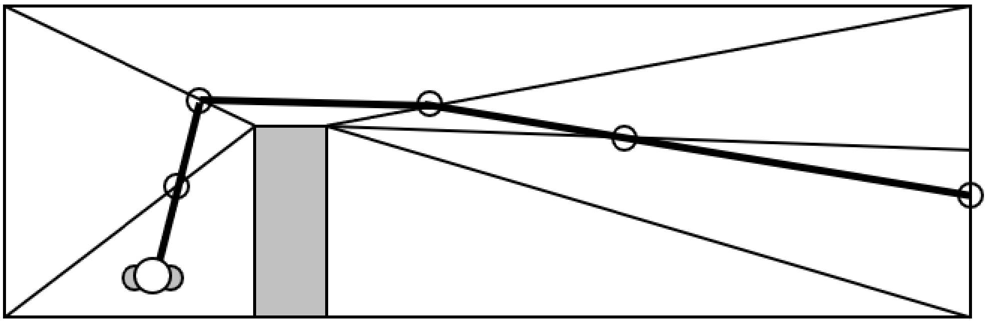


Figure 2.1: Example occupant route.

(Figure 9 from [5])

When a path is built, the occupant proceeds with *path following*. The following algorithm is used:

1. The current waypoint and the next waypoint are tracked.

2. If the current waypoint is reached, it is swapped with the next one.

3. The path is reevaluated if the occupant cannot see a straight line to the current waypoint or if the current target has been changed by the path planning algorithm.

4. A *seek curve* is generated using the algorithm on the operational level.

5. A move along the seek curve is attempted.

Interaction with other occupants and the movement itself is then computed on the operational level.

2.3 Operational Level

The operational level of pedestrian simulation takes care of the movement itself, i.e., how hard to accelerate, what steps to take, or how to avoid collisions. It contains perhaps the most diverse set of models, of which the most well-known include *cellular automata* models [18], *social-force* models[19], *optimal velocity* models [20], or the *optimal steps* model [21]. A nice overview can be found in the introductory chapter of [22] and a more comprehensive study in [9] and [23].

Pathfinder implements two distinct movement models: a simpler and faster *SFPE mode* based on the guides and calculations in the SFPE Handbook of Fire Protection Engineering [14] and a more complex *steering mode* based on the works in [24] and [25].

2.3.1 SFPE Mode

In SFPE mode, the movement is modeled purely in a flow-based fashion where no explicit occupant-to-occupant interactions take place. Each agent moves towards their target independently of others which means that multiple occupants can occupy the same space. The dynamics of the simulation are dictated solely by the densities in the rooms, flow rates of the doors, and the SFPE fundamental diagram presented in Section 1.2. Each room can additionally have a speed modifier attached which makes the occupants move slower or faster in order to simulate phenomena such as smoke. Stairs are a special type of room with a speed modifier to reflect the effect of the slope on the velocity.

The density in SFPE mode is considered uniform throughout the room and is computed as $\rho = N/A$ as in (1.5). It is used in the desired ("base") velocity v_b computation in the following way:

$$v_b = v_{\max} \cdot \delta(\rho) \cdot \tau, \tag{2.2}$$

where $v_{\max}$ is the free-flow speed, $\delta(\rho) \in [0, 1]$ is a scaling coefficient based on the density in the room and $\tau > 0$ is a scaling coefficient based on the terrain in the room. This equation is also used in steering mode but uses a modified form of density.

The $\delta(\rho)$ coefficient copies (1.11) from the fundamental diagram and is normalized to $[0, 1]$ interval:

$$\delta(\rho) = \begin{cases} 1 & \text{if } \rho < 0.55\,\mathrm{P/m^2}, \\ \max\left(\delta_{\min}, \frac{1}{0.85}\,(1 - \rho/3.76)\right) & \text{if } \rho \geq 0.55\,\mathrm{P/m^2}, \end{cases} \tag{2.3}$$

with $\delta_{\min} = 0.15$ used to prevent a complete stop and $1/0.85$ being a normalization constant. The τ coefficient is equal to 1 for level terrain and ramps, and for stairs, it decreases almost linearly as the slope of the stairs increases.

The other defining factor of the SFPE mode is the door flow rate. Once multiple occupants queue in front of a door, the first one crosses immediately, and the others have to wait for a time period that depends on the door flow rate. After that, another occupant is let through and the process repeats. The flow rate calculation is based on the fundamental diagram in (1.12) as follows:

$$J_s = k\rho(1 - \rho/3.76), \tag{2.4}$$

where k is a speed constant that depends on the terrain of the previous room and ρ is the maximum of the densities of the adjacent rooms. The resulting flow is adjusted for the width of the door w as $J = wJ_s$ and clamped so that low densities do not slow the flow too much and high densities do not cause it to degrade to zero. The time it takes for N occupants to pass through a door is $\frac{N-1}{J}$ as the first occupant does not have to wait for the door to clear.

2.3.2 Steering Mode

Steering mode provides a more complex way of following the seek curve where occupants do not travel in a straight line but follow a more natural path. Occupant-to-occupant interactions also take place and collision avoidance mechanisms are employed.

The desired velocity computation is the same as in (2.2), that is, it depends on the maximum velocity of the occupant and the density and terrain of the room. However, the density calculation is not uniform throughout the room anymore as it is calculated in the way described in Section 1.1.2, i.e., it is dependent on the distance to the closest occupant in front of the one in question.

Additionally, a maximum tangential forward acceleration component is computed as $a_f = \frac{v_{\max}}{t_{\text{accel}}}$, where t_{accel} is the defined acceleration time. A tangential backward acceleration component is obtained as $a_b = 2a_f$ and a radial component as $a_r = 1.5a_f$, which are combined to determine the final acceleration value $a_{\max}$. Note that these quantities, as well as the velocity computed in the previous paragraph, are only the magnitudes of their respective vectors, their direction is yet to be determined.

The direction choice itself works by first generating several discrete movement directions around the direction given by the seek curve. Then, for each direction, several steering behaviors are evaluated and assigned a cost. The number of directions and behaviors depends on the state that the occupant is in (idle or seeking). The following behaviors are possible:

Seek – Follow the seek curve as closely as possible.

Idle separate – In an idle state, maintain distance from other occupants.

Avoid walls – Detect walls and move to avoid collisions.

Avoid occupants – Detect occupants and move to avoid collisions.

Seek separate – Steer to lessen the perceived density.

Seek wall separate – Steer to maintain distance from walls.

Lanes – Move to a nearby lane.

Pass – Overtake slower occupants.

Cornering – Join a group of occupants that's about to move along a corner.

The total cost for each direction is obtained as a weighted sum of the costs of the individual behaviors and the one that minimizes the cost is chosen, i.e., the direction that provides the most utility based on the behaviors above.

Once the desired direction d_{des} is decided, it is scaled to the magnitude v_b computed above to obtain the desired velocity vector $v_{\vec{\text{des}}}$ and the acceleration vector is computed as

$$\vec{a} = a_{\text{max}} \frac{\vec{v}_{\text{des}} - \vec{v}_{\text{curr}}}{\|\vec{v}_{\text{des}} - \vec{v}_{\text{curr}}\|} \tag{2.5}$$

with $v_{\vec{\text{curr}}}$ being the current velocity vector. A stopping mechanism that prevents taking too large steps is also employed.

The velocity vector $\vec{v}_{\text{next}}$ and position $\vec{p}_{\text{next}}$ in the next time frame is obtained using explicit Euler integration as

$$\vec{v}_{\text{next}} = \vec{v}_{\text{curr}} + \vec{a}\Delta t \tag{2.6}$$

$$\vec{p}_{\text{next}} = \vec{v}_{\text{curr}} + \vec{a}\Delta t, \tag{2.7}$$

where Δt is the time step.

The final component of the steering behavior is the priorities mentioned in Section 2 which are used in conflict resolution. Each occupant is assigned a priority level and when they meet an occupant with the same priority they act as described above. When they, however, meet an occupant with a different priority level (which will always be the case in the coming experiments), they adapt their behavior. If the other occupant is of lower priority, no explicit action is made. If the other occupant is of higher priority, the occupant will temporarily switch to an idle state and essentially make way for the higher-priority occupant.

In situations where there is a movement conflict, for example, when grouping in front of a bottleneck, the highest-priority occupants are let through first. When the occupants are of equal priority and none of them has a clear path, their priorities are locally elevated and the conflict is resolved with the new priorities. The exact way is not described in the reference.

The steering system also implements other advanced techniques such as further collision avoidance or door flow rate limiting, but for this thesis, they are not as important.

Chapter **3**

Sensitivity Analysis

The second major theoretical prerequisite for this thesis is sensitivity analysis (SA). A possible definition was given by Saltelli [26]: "*The study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input.*". It is essentially a way of understanding how the system or model in question behaves with respect to its input parameters. Which parameters are the most important? Which ones are irrelevant? What combinations of input values correspond to "interesting" outputs? Methods of sensitivity analysis can answer all of these questions and more. For a detailed introduction to the topic, I would point the readers to [26] and [27]. This chapter will provide an overview of the basic concepts relevant to this thesis and present the available software solutions.

3.1 Model

Firstly, let us define the model setting. Most texts concerning SA work with the model $Y = f(\boldsymbol{X})$, where $\boldsymbol{X} = X_1, \dots, X_k$ are the input variables (*factors*) and Y is a scalar output. The model f is considered a black box, i.e., we have no analytical knowledge about its inner structure. The model presented in this fashion is deterministic – the uncertainty is located in the parameters. An example provided in Chapter 3.1 of [26] is that of a bungee jumper preparing for a jump from a bridge whose exact height he doesn't know and wants to choose a jumping cord to maximize his enjoyment but minimize the risk of hitting the ground. The simplified physical model is deterministic – the uncertainty is in the height of the bridge and the thickness of the cord.

This is different from our setting of measuring the total evacuation time of simulated evacuations. This model is inherently nondeterministic because of the dependence on random seeds which reflect the uncertainty in the behavior of a person's decision-making process. In addition to the uncertainty in the parameters, the output itself is a random variable. In order to suppress the randomness of the model function, we will instead be working with the expected values at given points in the parameter space. Denoting the total evacuation time with respect to the parameters $\boldsymbol{X}$ as $TET_{\boldsymbol{X}}$, the model takes on the following form:

$$Y = f(\boldsymbol{X}) = \mathbb{E}\left[TET_{\boldsymbol{X}}\right]. \tag{3.1}$$

In practice, this means that for each combination of input parameters, the model will be run several times with different random seeds and the outputs averaged.

A distinctive property of the model is that it is often very computationally expensive – a single evaluation of the model takes a long time. The simulations in this thesis take approximately 5 seconds per run, but more complex models may take up to hours or days at a time. Even with the relatively short run time of 5 seconds, if we were to sample the k-dimensional parameter space with 1000 points for each variable and estimate each conditional mean with the average of 50 runs, we would need to run the model $k \times 1000 \times 50 = 50\,000k$ evaluations. For $k = 5$ this equals approximately 14 days of computation time. For this reason, the model is often replaced with a *surrogate model* using statistical or machine-learning techniques. The surrogate model is less precise but fast to evaluate which can be used to obtain preliminary results, assuming that the surrogate model is a reasonably good approximation. Popular methods include polynomial regression, Gaussian processes (kriging), random forests, SVM regression, or artificial neural networks [28].

One more matter that is worth addressing are the individual probability distributions of the factors. Many methods by-default work with a uniform distribution on the interval $[0, 1]$ because it is easy to evenly partition and sample. Real-world variables of interest will rarely have this distribution though, so when working with different distributions we will instead sample the *quantiles* uniformly and map them to the actual values via the *inverse cumulative distribution function* (quantile function). This works because when a continuous random variable X has the cumulative distribution function F_X, we can define the transformed variable $U = F_X(X)$ and if for a given $u \in [0, 1]$ the value $F_X^{-1}(u)$ exists, then

$$\begin{aligned} F_U(u) &= \Pr(U \leq u) \\ &= \Pr(F_X(X) \leq u) \\ &= \Pr\left(X \leq F_X^{-1}(u)\right) \\ &= F_X\left(F_X^{-1}(u)\right) \\ &= u, \end{aligned} \tag{3.2}$$

which is the CDF of a standard uniform random variable, meaning that we can freely move between the standard uniform distribution and other “well-behaved” distributions. When working with discrete variables it is customary to also employ this continuous approach and then simply round to the nearest integer [29].

3.2 Settings

There are various distinct use cases for sensitivity analysis as implied by the questions in the first paragraph of this chapter. The most common settings as identified by Saltelli in [26] are the following:

Factor prioritization (FP) The FP setting is concerned with identifying the most important factors. The "importance" of a factor in this context is understood in terms of variance – the most important factors contribute to the total variance of the output the most or, equivalently, fixing the factor to its conditional mean given all the other factors will lead to the largest reduction in the total variance. The methods used in this setting are often expensive in terms of the number of model evaluations and techniques such as surrogate modeling or factor screening are employed. This is the main setting this thesis focuses on with the method of choice being *variance-based sensitivity analysis* [30].

Factor fixing (FF) The FF setting (also called *screening*) has the opposite objective – it aims to identify the non-influential factors which can then be fixed at an arbitrary value without losing too much information in the output. The resulting model will then be simpler due to the presence of fewer variables, enabling us to employ more computationally-demanding techniques of factor prioritization. The method of *elementary effects* [31] (also called *Morris' method*) will be used in this thesis.

Variance cutting (VC) The goal of this setting is to get the unconditional output variance below a given threshold by fixing the values of as few factors as possible. This can be useful in risk analysis where we, for example, want to eliminate various sources of uncertainty in order to keep the risk at an acceptable value. The methods used in factor prioritization can also be employed in this setting, but the setting itself will not be explored in this thesis.

Factor mapping (FM) The final basic setting is that of factor mapping which aims to classify the output usually into two groups, e.g., interesting and non-interesting or acceptable and unacceptable, and asks the question "Which values of the input factors are the most likely to produce outputs in the region of interest?". An application is in reliability engineering where we are interested in what combinations of features will cause a component to fail. The method of choice is *Monte Carlo filtering* [32] and *Smirnov test* [33]. This setting will likewise not be explored further in this thesis.

As mentioned in their respective paragraphs, the main focus of the coming chapters will be on factor prioritization and factor fixing. What follows is a description of the used methods.

3.3 Methods

Over the course of previous decades, numerous distinct approaches to sensitivity analysis have been developed. The simplest method that most people use without realizing it is a basic *scatter plot*. Figure 3.1 shows an example scatter plot of the function $f(X_1, X_2) = 2X_1 - X_2$ with $X_1, X_2 \sim \mathcal{N}(0, 1)$ against each of the variables. Clearly, X_1 has a larger effect on the output which is evident by a clearer "shape" of the plot. Scatter plots are an effective way of doing a preliminary analysis of the problem but when the number of factors is in the tens or more, their usage becomes impractical, and more sophisticated methods have to be employed.

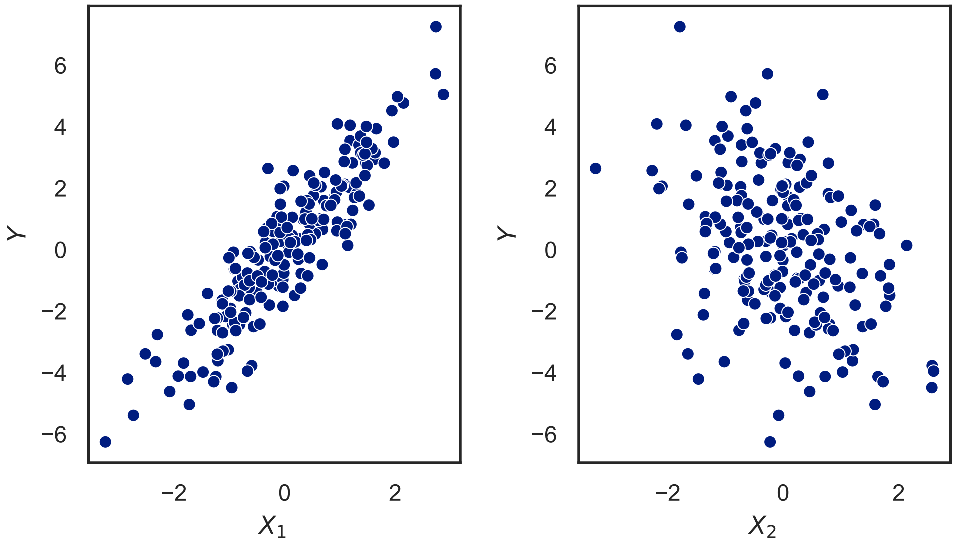


Figure 3.1: Example scatter plots of $Y = 2X_1 - X_2$ with $X_1, X_2 \sim \mathcal{N}(0, 1)$.

Originally, SA methods were often based on partial derivatives $\partial Y / \partial X_i$ [34] or their discrete approximations. These are the so-called *local methods*, also called *one-at-a-time (OAT)* methods, in which a single factor X_i is being varied while all the other $X_j \neq X_i$ are fixed. The change in the output is then attributed to the change in X_i because the value of all the others stayed the same. This method is still one of the most popular methods used in practice [35]. Despite its popularity, it has two critical drawbacks and A. Saltelli advises strongly against it [36]:

- It does not explore the parameter space properly. If the model is nonlinear, the local behavior is unlikely to represent the overall behavior of the model adequately.

- It fails to detect the interaction between variables. Using an OAT design when variable interactions are present, i.e., when varying multiple variables at once has a specific effect on the output, can lead to underestimating the sensitivities of such factors [37].

Due to these limitations, I will not focus on OAT design in this thesis further.

On the other side of the spectrum are the *global methods* which aim to explore the entire parameter space as efficiently as possible. The most popular methods [38] are

- regression-based methods,
- variance-based methods, and
- elementary effects,

which are all used in this thesis.

3.3.1 Regression-Based Methods

A simple global sensitivity measure can be obtained by performing a standard ordinary-least-squares linear regression on the model output using the input factors, obtaining the proxy model

$$Y = b_0 + \sum_{i=1}^{k} b_i X_i + \varepsilon, \tag{3.3}$$

where $b_1, \dots, b_k$ are the regression coefficients and b_0 is the intercept term, and ε is the noise term with $\mathbb{E}[\varepsilon] = 0$. We could then use the regression coefficients as the "importance weights" of each variable. This approach, however, fails when the individual factors have different scales. For example, if X_1 was in the order of hundreds and X_2 in the order of thousands, but their relative importance was equal, then X_1's importance would be underestimated while X_2's importance would be overestimated.

This problem can be avoided by standardizing the regression coefficients by the input–output standard deviations σ_{X_i}/σ_Y to obtain the *standardized regression coefficients (SRC)*

$$\beta_i = \frac{\sigma_{X_i}}{\sigma_Y} b_i. \tag{3.4}$$

Technically, the division by σ_Y is not necessary to determine the relative importance of the factors, but it provides the model with certain desirable properties, mainly that

- the coefficients are unitless, i.e., one standard deviation change in X_i will lead to β_i standard deviations change in Y, and
- $\sum_{i=1}^{k} \beta_i^2 = R^2$, where R^2 is the coefficient of determination which, in this case, measures the proportion of the output variance explained by the linear model [27, Ch. 1.2].

Additionally, the sign of the coefficient signals whether the variable has a positive or negative effect on the output. The regression-based approach is fast and simple and can also be used as a surrogate model.

However, it also has several shortcomings. First of all, it is *model-dependent* – it assumes that the linear model is a good fit. If the true model is nonlinear, this assumption may not hold and the results may be misleading. For this reason, it is necessary to check the goodness of fit before making any claims about the variables' effects – in [26, Ch. 2.2], Saltelli advises an R^2 coefficient of at least 0.7. Secondly, it fails to detect interactions between factors. The β coefficients only measure the first-order interactions, i.e., the effects of the variables "in a vacuum". This is fine if the true model is linear because there are no interactions in a linear model but as the nonlinearity of the model increases, so does the strength of the variable interactions. This can be partially improved upon by adding polynomial features to the linear model but this also causes the complexity of the model to rapidly increase. For these reasons, more complex techniques were developed that are *model-free* and consider variable interactions by default.

3.3.2 Variance-Based Methods

A more sophisticated approach, and the de facto standard today, is the variance-based approach. As a motivational example, I will explain the intuition behind the *first-order sensitivity index* S_i (the formal definitions will come later):

$$S_i := \frac{\operatorname{Var}_{X_i}\left(\mathbb{E}_{\boldsymbol{X}_{\sim i}}\left[Y \mid X_i\right]\right)}{\operatorname{Var}\left(Y\right)}, \tag{3.5}$$

where the notation $\boldsymbol{X}_{\sim i} = X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_k$ means all the variables except for X_i. The idea can be illustrated well via X_i–Y scatter plots. First, we divide the x-axis into N bins $B_1, \dots, B_N$ with an approximately equal number of points in each bin. Then, for each bin B_n, we compute the average value of Y w.r.t. X_i for all the points in said bin, essentially averaging out the effect of all the other variables to obtain $\mathbb{E}_{\boldsymbol{X}_{\sim i}}\left[Y \mid X_i \in B_n\right]$. Finally, we look at how "spread out" these average values are w.r.t. the binning by looking at the variance $\operatorname{Var}_{B_n}\left(\mathbb{E}_{\boldsymbol{X}_{\sim i}}\left[Y \mid X_i \in B_n\right]\right)$. If they have a low variance, it is likely that X_i only has a small effect on the output, and if their variance is high, then X_i influences the output more. This, of course, depends on how we construct the bins, but if we make them infinitely small and additionally divide them by the output variance, we will arrive at exactly the first-order sensitivity index S_i:

$$\lim_{|B_n|\to 0} \frac{\mathrm{Var}_{B_n}\left(\mathbb{E}_{\boldsymbol{X}_{\sim i}}\left[Y \mid X_i \in B_n\right]\right)}{\mathrm{Var}\left(Y\right)} = \frac{\mathrm{Var}_{X_i}\left(\mathbb{E}_{\boldsymbol{X}_{\sim i}}\left[Y \mid X_i\right]\right)}{\mathrm{Var}\left(Y\right)} = S_i. \tag{3.6}$$

The idea is illustrated in Figure 3.2 using the function

$$Y = \sin(X_1) + 0.1X_2 \tag{3.7}$$

with $X_1, X_2 \sim \mathcal{U}(-\pi, \pi)$, where most of the variance can be attributed to X_1, and the sensitivity indices obtained by binning are $S_1 \approx 0.9$ and $S_1 \approx 0.1$.

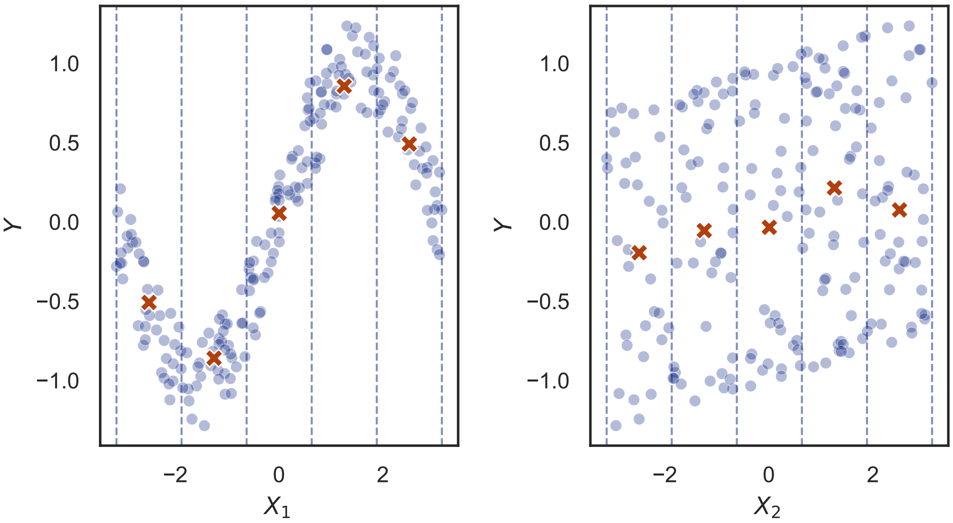


Figure 3.2: Conditional variances as a basis for SA.

This process can easily be generalized to groups of factors where the bins will become multidimensional, although the definition of the indices needs a slight adjustment so as to not count any effect twice, as will be seen shortly.

Sobol' sensitivity indices The method was originally proposed by Sobol' in [30], where he proved that if the model function $Y = f(\boldsymbol{X})$ of k variables is square-integrable, the input factors are uncorrelated (orthogonal), and defined over the unit hypercube[2]

$$\Omega^k = \{x \mid 0 \leq x_i \leq 1; i = 1, \ldots, k\}\,, \tag{3.8}$$

then f can be decomposed into a sum of functions with increasing dimensionality

$$f = f_0 + \sum_{i=1}^{k} f_i + \sum_{i=1}^{k}\sum_{j>i}^{k} f_{ij} + \cdots + f_{1\ldots k}, \tag{3.9}$$

[2] For practical scenarios we can assume that these are the quantiles of different continuous distributions.

where f_0 is a constant and the others are functions only of the factors in its index, e.g., $f_i = f_i\left(X_i\right)$ or $f_{ij} = f_{ij}\left(X_i, X_j\right)$. The f_i are called the first-order terms, f_{ij} are called the second-order terms, and so on. If all the functional terms in (3.9) have zero means[3], i.e., their integrals with respect to their variables are zero, then the decomposition is unique and can be computed in terms of conditional means of the variables:

$$\begin{aligned} f_0 &= \mathbb{E}\left[Y\right], \\ f_i &= \mathbb{E}\left[Y \mid X_i\right] - f_0, \\ f_{ij} &= \mathbb{E}\left[Y \mid X_i, X_j\right] - f_i - f_j - f_0, \end{aligned} \tag{3.10}$$

and so on. This decomposition is called the *high-dimensional model representation (HDMR)*. To translate this into the language of variances, one can square and integrate each term over the unit hypercube Ω^k to obtain

$$\int_{\Omega^k} f^2\left(\boldsymbol{X}\right) \mathrm{d}\boldsymbol{X} - f_0^2 = \sum_{n=1}^{k} \sum_{i_1 < \cdots < i_n}^{k} \int \cdots \int_{\Omega^k} f_{i_1 \dots i_n}^2 \, \mathrm{d}X_{i_1} \dots \mathrm{d}X_{i_k}, \tag{3.11}$$

where the term on the left is, in fact, the unconditional variance $\mathrm{Var}(Y)$ and the terms on the right are also variance terms decomposed with respect to the variables in their index. In a more readable way, the decomposition is analogical to the functional decomposition in (3.9):

$$\mathrm{Var}(Y) = \sum_{i=1}^{k} V_i + \sum_{i=1}^{k} \sum_{j>i}^{k} V_{ij} + \cdots + V_{1 \dots k}, \tag{3.12}$$

with

$$\begin{aligned} V_i &= \mathrm{Var}_{X_i}\left(\mathbb{E}_{\boldsymbol{X}_{\sim i}}\left[Y \mid X_i\right]\right), \\ V_{ij} &= \mathrm{Var}_{X_i, X_j}\Big(\mathbb{E}_{\boldsymbol{X}_{\sim i,j}}\left[Y \mid X_i, X_j\right]\Big) - V_i - V_j, \end{aligned} \tag{3.13}$$

and so on. Finally, by dividing each side by $\mathrm{Var}(Y)$ we obtain the desired *Sobol' sensitivity indices (SSI)*

$$1 = \sum_{i=1}^{k} S_i + \sum_{i=1}^{k} \sum_{j>i}^{k} S_{ij} + \cdots + S_{1 \dots k}. \tag{3.14}$$

The number of variables in the index of each term is called the *order* of the interaction. The terms in the equation above are all in the range $[0, 1]$ and correspond to the portion of the total variance that can be attributed directly to the combination of variables in its argument, as all the lower-order interaction terms were filtered out. The decomposition is also possible for correlated inputs, but it will not be explored in this thesis.

[3] Which will be true for uncorrelated inputs.

The first-order indices S_i also have a connection with the standardized regression coefficients β_i from Section 3.3.1 in that $S_i = \beta_i^2$ if the model is linear [27, Ch. 1.2]. The major advantage of the sensitivity indices, however, is that they are model-agnostic. While the sum of the squared SRCs measures the proportion of variance explained by a linear model, the sum of the first-order SSIs measures the proportion of variance explained by an additive model, i.e., a model in which the effects of the variables can be partitioned into individual summands such as the one in (3.7).

Total effects Since the number of terms in (3.14) is $2^k - 1$, computing all of them quickly becomes infeasible. Is there a way to aggregate all the effects of X_i into a single number? We can look at the term

$$\mathrm{Var}_{\boldsymbol{X}_{\sim i}}\left(\mathbb{E}_{X_i}\left[Y \mid \boldsymbol{X}_{\sim i}\right]\right) = \sum_{j \in \mathcal{P}^+(\sim i)} V_j, \tag{3.15}$$

where $\mathcal{P}^+(\sim i)$ is the set of all non-empty subsets of $\{1, \ldots, i-1, i+1, \ldots, k\}$ that don't contain i, meaning that (3.15) contains the effect of all the terms of any order which *don't* include X_i. Because the sum of all V terms is $\mathrm{Var}(Y)$, then the terms absent in (3.15) must make up all the terms of any order that *do* contain X_i. This leads us to the definition of the *total effect term (TE)* T_i:

$$T_i = 1 - \frac{\mathrm{Var}_{\boldsymbol{X}_{\sim i}}\left(\mathbb{E}_{X_i}\left[Y \mid \boldsymbol{X}_{\sim i}\right]\right)}{\mathrm{Var}(Y)}. \tag{3.16}$$

The total effect statistic gives us a summary statistic of how much each variable affects the output. It does not suffer from false negatives, where an influential variable is not detected, as SRCs do. As an example, for a model with three factors X_1, X_2, X_3, the total effect terms have the following form:

$$\begin{aligned} T_1 &= S_1 + S_{12} + S_{13} + S_{123}, \\ T_2 &= S_2 + S_{12} + S_{23} + S_{123}, \\ T_3 &= S_3 + S_{13} + S_{23} + S_{123}. \end{aligned} \tag{3.17}$$

It is worth noting that $\sum_i T_i \neq 1$ unless the model is purely additive because the higher-order terms will get counted more than once as is apparent from (3.17).

Practical use and computation In order to compute the SSIs and TEs in a naive way, one would need to first estimate the inner expectations with M points and then the outer variance with N points, leading to $M \times N$ model evaluations for each of the $2^k - 1$ indices. However, it is a standard practice to only compute the S_i and T_i [26, Ch. 5], cutting the cost to $2k \times M \times N$. These generally give us enough information to make informed claims about the model:

- The S_i values give us an idea of how much X_i participates in simple additive interactions.
- The T_i values measure the overall importance of the variable.
- The differences $T_i - S_i$ indicate how much X_i participates in composite interactions.

Furthermore, it is also possible to cut the $M \times N$ cost. A Monte Carlo approach was suggested by Sobol' in [39] and refined by Saltelli in [40], in which both the S_i and T_i can be computed using only $N(k+2)$ model evaluations, or $N(2k+2)$ model evaluations if S_{ij} are also required. I will not be going too deeply into the algorithm as it is not the primary objective of this thesis, but in short:

1. Two matrices $\mathbf{A}$ and $\mathbf{B}$ of size $N \times k$ are generated. These contain the model inputs generated from the marginal distribution of each factor.

$$\mathbf{A} = \begin{pmatrix} x_1^{(1)} & \dots & x_k^{(1)} \\ x_1^{(2)} & \dots & x_k^{(2)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \dots & x_k^{(N)} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \tilde{x}_1^{(1)} & \dots & \tilde{x}_k^{(1)} \\ \tilde{x}_1^{(2)} & \dots & \tilde{x}_k^{(2)} \\ \vdots & \ddots & \vdots \\ \tilde{x}_1^{(N)} & \dots & \tilde{x}_k^{(N)} \end{pmatrix}. \tag{3.18}$$

2. The model is evaluated for each point in $\mathbf{A}$ and $\mathbf{B}$ to obtain $\boldsymbol{y}_{\mathbf{A}}$ and $\boldsymbol{y}_{\mathbf{B}}$.

$$\boldsymbol{y}_{\mathbf{A}} = f(\mathbf{A}), \quad \boldsymbol{y}_{\mathbf{B}} = f(\mathbf{B}). \tag{3.19}$$

3. For each X_i, the matrix $\mathbf{C}_i$ is constructed which contains all the columns from $\mathbf{A}$ except for the i-th column, which is taken from $\mathbf{B}$.

$$\mathbf{C}_i = \begin{pmatrix} x_1^{(1)} & \dots & \tilde{x}_i^{(1)} & \dots & x_k^{(1)} \\ x_1^{(2)} & \dots & \tilde{x}_i^{(2)} & \dots & x_k^{(2)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^{(N)} & \dots & \tilde{x}_i^{(N)} & \dots & x_k^{(N)} \end{pmatrix}. \tag{3.20}$$

4. The model is evaluated for each point in each $\mathbf{C}_i$ to obtain $\boldsymbol{y}_{\mathbf{C}_i}$.

$$\boldsymbol{y}_{\mathbf{C}_i} = f(\mathbf{C}_i). \tag{3.21}$$

5. The unconditional mean and variance are estimated in the standard way.

$$\begin{aligned} \hat{\mathbb{E}}[Y] &= \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{y}_{\mathbf{A}}^{(j)}, \\ \widehat{\mathrm{Var}}(Y) &= \frac{1}{N}\sum_{j=1}^{N} \left(\boldsymbol{y}_{\mathbf{A}}^{(j)}\right)^2 - \hat{\mathbb{E}}[Y]^2. \end{aligned} \tag{3.22}$$

6. Sensitivity indices and total effects are estimated for each X_i.

$$\begin{aligned}\hat{S}_i &= \frac{(1/N)\sum_{j=1}^{N} \boldsymbol{y}_{\mathbf{B}}^{(j)} \boldsymbol{y}_{\mathbf{C}_j}^{(j)} - \hat{\mathbb{E}}[Y]^2}{\widehat{\mathrm{Var}}(Y)},\\ \hat{T}_i &= 1 - \frac{(1/N)\sum_{j=1}^{N} \boldsymbol{y}_{\mathbf{A}}^{(j)} \boldsymbol{y}_{\mathbf{C}_j}^{(j)} - \hat{\mathbb{E}}[Y]^2}{\widehat{\mathrm{Var}}(Y)}.\end{aligned} \tag{3.23}$$

A discussion of why this works and possible improvements is provided in [40]. One notable technique that speeds up the convergence is to not generate the matrices **A** and **B** randomly but to instead use low-discrepancy sequences such as Sobol' sequences [41].

Law of total variance One additional form of variance decomposition that is connected to the variance-based methods and is of particular interest in our setting is the so-called *Law of total variance* or *Eve's law* which states that if $X_1, \ldots, X_k$ and Y are random variables, then

$$\mathrm{Var}(Y) = \mathbb{E}[\mathrm{Var}(Y \mid X_1, \ldots, X_k)] + \mathrm{Var}(\mathbb{E}[Y \mid X_1, \ldots, X_k]). \tag{3.24}$$

A proof can be found, for example, in [42]. If $X_1 \ldots X_n$ are the parameters of the model (inputs), the term on the right can be interpreted as the variance explained by the model parameters. This is because the inner expectation eliminates the inherent stochasticity of the model and the resulting variance is due to the deterministic part. The term on the left has the opposite meaning – it is the average variance that is induced by the randomness of the output. This formula allows us to quantify how much of the uncertainty in the evacuation time can be attributed to purely random factors, as will be shown in Section 6.2.

3.3.3 Elementary Effects

The final method used in this thesis is the so-called elementary effects method developed by Morris [31] and later improved by Campolongo [43]. It is the standard method used for screening – identifying the variables whose effect on the output is negligible and fixing them to a baseline value. The goal is mainly to reduce the dimensionality of the problem and enable the use of more powerful but also more computationally expensive techniques such as the variance-based methods.

For this to work, the method must provide reliable sensitivity estimates with only a few model evaluations. The elementary effects method approximates the partial derivatives with respect to each input variable in different sample points in the parameter space and aggregates them to obtain a global measure. There are visible similarities with the one-at-a-time method, which was discouraged earlier, however, the usage of multiple sample points serves to overcome the shortcomings of the OAT approach. This method has been shown to produce reasonable rankings of the factors in as few as 10 to 100 samples per factor [43].

EE and derived measures As before, the input factors $X_1, \dots, X_k$ are assumed to be defined over the unit hypercube Ω^k, and translation to their actual values can be done via the quantile function. Each factor's input range is evenly discretized into a p-level grid with a spacing of $\delta = 1/(p-1)$ to obtain Ω_p^k. For example, if $p = 4$ (a common choice), then each X_i can take on the values $\{0, 1/3, 2/3, 1\}$. Finally, a step size Δ is chosen as $n \cdot \delta$ for an $n \in \{1, \dots, p-2\}$ so that for an $x \in X_i$, at least one of $x + \Delta$ or $x - \Delta$ is still in X_i. The recommended choice is $p/2 \cdot \delta$ as it has some desirable sampling properties, namely that it does not inherently favor any points in the sample space [27, Ch. 2.4]. This also forces the p to be an even integer. To finish the example, for $p = 4$, the corresponding Δ would be chosen as $2/3$. It is also worth noting that the analogy with partial derivatives is not entirely accurate as the step size Δ is not necessarily a small one – the most commonly used value makes for steps larger than half of the variable's range.

We can now define the elementary effect associated with X_i as

$$EE_i(\boldsymbol{X}) = \frac{f(X_1, \dots, X_i + \Delta, \dots, X_k) - f(X_1, \dots, X_i, \dots, X_k}{\Delta}, \tag{3.25}$$

or, in short,

$$EE_i(\boldsymbol{X}) = \frac{f(\boldsymbol{X} + \Delta \boldsymbol{e}_i) - f(\boldsymbol{X})}{\Delta}, \tag{3.26}$$

where $\boldsymbol{e}_i$ is a vector of zeros with a one in position i and $\boldsymbol{X} \in \Omega_p^k$ such that $\boldsymbol{X} + \Delta \boldsymbol{e}_i$ is still in Ω_p^k.

The number of elementary effects associated with X_i is finite due to the discrete nature of the grid and their distribution is denoted F_i. Additionally, the distribution of their absolute values is denoted G_i. The sensitivity measures μ_i, σ_i, and μ_i^* are defined in the following way:

$$\mu_i = \mathbb{E}_{\boldsymbol{X} \sim F_i} \left[EE_i \left(\boldsymbol{X} \right) \right], \tag{3.27}$$

$$\sigma_i^2 = \operatorname{Var}_{\boldsymbol{X} \sim F_i} \left(EE_i \left(\boldsymbol{X} \right) \right), \tag{3.28}$$

$$\mu_i^* = \mathbb{E}_{\boldsymbol{X} \sim G_i} \left[\left| EE_i \left(\boldsymbol{X} \right) \right| \right]. \tag{3.29}$$

Practical use and computation Morris originally proposed only the first two measures μ_i and σ_i. A large (absolute) value of μ_i signals that a shift of X_i by Δ leads to a large (absolute) change in the output. However, a μ_i close to 0 does not necessarily imply an unimportant variable because when f is non-monotonic and the individual EE_is have different signs, their effects could cancel out and the method would not detect this. To overcome this, the μ_i^*

measure was introduced by Campolongo which does not suffer from this issue. There could, of course, be cases where even this measure fails to detect an important variable, for example, if the function f was periodic with a period coinciding with the minimum step size δ. If one wants to be extra careful, one could try multiple different grids. It has been shown that the measure μ_i^* is a good proxy for the total effect index T_i [43] and even an analytical link has been proven [44].

The measure σ_i^2 indicates how much the factor participates in higher-order interactions. Intuitively, if the value of EE_i varies considerably across different sample points, it means that the values of the other factors also determine how big of an effect X_i has on the output in the given points.

The standard way of computing EEs and estimating the sensitivity measures is via *trajectories*. A trajectory is a sequence of $k+1$ points in Ω_p^k such that each pair of consecutive points differs only in a single coordinate, which is either increased or decreased by Δ, and this happens exactly once for every dimension. An example trajectory is shown in Figure 3.3. A single trajectory yields k elementary effects, one for each factor, obtained via (3.26) with the slight adjustment that if the factor's value in the trajectory was decreased instead of increased, then the order of the terms in the numerator is inverted. The μ_i, σ_i, and μ_i^* for each factor are then estimated from n independent trajectories, for a total cost of $n \times (k+1)$ model evaluations to obtain $n \times k$ elementary effects.

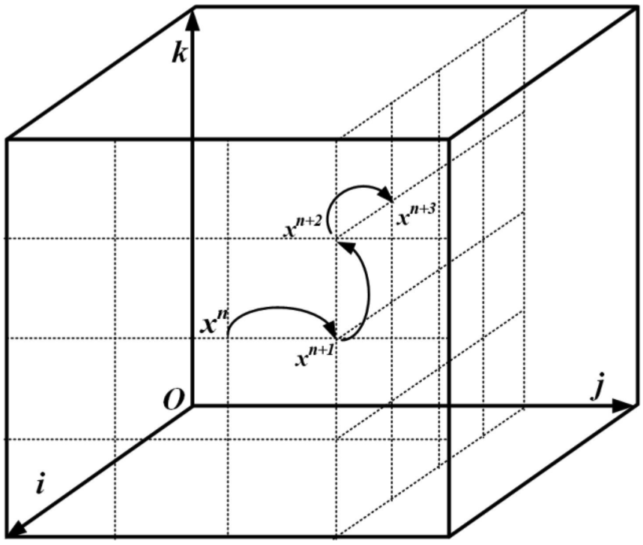


Figure 3.3: An example trajectory with three factors.

(Figure 1 from [45])

Campolongo in [43] proposed an improved way of sampling the trajectories by first generating a large number of trajectories N and then selecting a subsample of $n < N$ trajectories with maximum "spread" in the input space. Saltelli also developed a so-called *radial* design which does not generate trajectories at all but instead simply samples a number of random points in the input space and performs an OAT analysis at each point, averaging the values in the end [46]. The μ^* measure can also be extended to groups of factors, as shall be seen in Section 6.4.2. In this case, the factors are not perturbed one at a time, but all the factors belonging to a group are moved at once.

3.4 Software Solutions

There are packages implementing the described methods in every major language used for numerical computations. For instance, R language has the package `sensitivity`[4], Matlab has the Simulink environment[5], and Julia has the package `GlobalSensitivity.jl`[6]. There also exists dedicated software for sensitivity analysis such as optiSLang[7].

The language of choice for this thesis is Python and the leading SA library is SALib [47, 48]. The library is open-source and implements both the Sobol' method and the elementary effects method, as well as other popular techniques that weren't explored in this chapter such as the Fourier amplitude sensitivity test (FAST) [49] or derivative-based global sensitivity measure (DGSM) [44]. A form of the HDMR decomposition from (3.9) via least-squares regression is implemented as a method of surrogate modeling.

A topic that is closely connected to sensitivity analysis is experimental design, which is concerned with answering the question "How to properly sample the parameter space?". When the dimension of the problem is high, a full grid search (also called the full factorial design) is infeasible, but a purely random sampling strategy may leave large gaps in one part of the parameter space and form clusters in other parts. An overview of experimental design is provided in [27, Ch. 2]. SALib implements various more-sophisticated sampling strategies including fractional factorial, Latin hypercube sampling, or Sobol' sequences. The theory behind experimental design was not explored in this thesis aside from the brief mention of Sobol' sequences in Section 3.3.2.

SALib's API is based on a four-step pipeline:

1. Define the problem via the `ProblemSpec` class. The class is essentially an expanded dictionary that contains:
    - The list of variables and their names.
    - The distribution of each variable. By default, it is assumed that the variable follows a uniform distribution with given bounds, but

[4] `https://cran.r-project.org/web/packages/sensitivity`

[5] `https://www.mathworks.com/products/simulink.html`

[6] `https://docs.sciml.ai/GlobalSensitivity/stable`

[7] `https://www.ansys.com/products/connect/ansys-optislang`

different distributions such as normal or lognormal distributions are also available.

- Optionally, the group to which each variable belongs, if the method supports grouping.
- The output variable.

2. Generate a list of input samples via the `sample()` function.
3. Separately evaluate the model for each input sample.
4. Compute the sensitivity measure via the `analyze()` function.

Each SA method has its own implementation of the `sample()` and `analyze()` function. A basic visualization of the results is also provided.

SALib, however, does not provide tools for surrogate modeling apart from the aforementioned HDMR decomposition. This functionality is provided by standard Python numerical libraries such as NumPy[8] or SciPy[9]. More complex models are implemented in scikit-learn[10] or PyTorch[11] (neural networks).

[8] `https://numpy.org`

[9] `https://scipy.org`

[10] `https://scikit-learn.org`

[11] `https://pytorch.org`

Chapter **4**

Experimental Setting

This chapter opens up the practical part of the thesis. Before beginning the analysis, two concerns have to be addressed:

1. Selecting proper geometrical setting for analysis.
2. Modeling the pedestrians themselves.

Both of these points will be addressed in the coming sections, as well as a description of the selected approaches.

4.1 Geometry Selection

There are various different geometries that share the general characteristics of a cramped interior, i.e., narrow corridors and high occupant densities. Despite sharing these properties, one would not expect every cramped layout to exert the same pedestrian behavior. With consultation from the experts at the Faculty of Civil Engineering, CTU in Prague, we have chosen the following scenarios based on their accessibility and the potential of conducting real-life experiments:

Multi-decked train – A highly irregular geometry with multiple decks connected by stairs.

Lecture hall – A more homogeneous geometry with identical rows and single stairs on the side.

These geometries contrast each other in many of their properties. The train has a very closed and narrow interior, the field of vision is reduced due to the presence of walls, and the route to exit will differ considerably based on the initial position. The lecture hall, on the other hand, is a somewhat open geometry where the occupant can directly see what is happening and the evacuation process will consist of reaching the stairs and subsequently joining the queue in front of the exit regardless of the occupant's initial position.

4.1.1 Train

There have been several studies concerning the egress of trains and railway cars in the past. For example, [50] explored the behaviors of passengers induced by narrow aisles under laboratory conditions, [51] analyzed the evacuation of a metro train in Vienna with respect to the exit geometry, or [52] investigated the effects of reduced visibility due to the presence of smoke. There have also been models developed specifically to account for the intricacies of train evacuation [53, 54].

The main publication of interest is [1] by Najmanová et al., which is based on an experimental evacuation experiment of a ČD Class 471 train commonly known as *CityElefant*, conducted in Prague in 2018. The study focused on the effect of boundary conditions – types and widths of exit geometry, and the composition of the crowd. Three exit geometries were tested:

- Exit to a high platform.
- Exit to an open line via stairs.
- Exit to an open line via a 75 cm jump.

Five different exit door widths were tested – 65 cm, 75 cm, 90 cm, 110 cm, and 134 cm. Additionally, there were two different types of crowds investigated – a homogeneous crowd (HOM) consisting of 18–38 years old participants with no movement limitations, and a heterogeneous crowd (HET), in which 28 % of the group had some type of movement impairment (slower walking speed, greater diameter). The study also employed sensitivity analysis techniques on the effect of the exit width, exit type, and the number of movement-impaired individuals.

There was also a follow-up study presented at the Fire and Evacuation Modeling Technical Conference (FEMTC) 2022, which concerned itself with modeling the evacuation in Pathfinder and validating the model against empirical data [55]. Further sensitivity analysis was performed on the average maximum speed, average diameter, and a probabilistic measure of the occupancy of the bottom deck. Additionally, a preliminary analysis of the effects of the initial positions of the occupants was conducted.

This thesis aims to expand upon the results of [1] and [55]. The model of the train in Pathfinder was created as part of the aforementioned experimental study. To keep the dimensionality of the problem at a reasonable level, only the exit to a platform with a door width of 90 cm was chosen. A photo of the train interior along with a screenshot of the modeled geometry is shown in Figure 4.1.

The train itself has three decks – a bottom deck with a capacity of 23 occupants, a middle deck with a capacity of 13 occupants, and a top deck with a capacity of 31 occupants. The individual decks are connected via stairs and there is a small room with no seats near the exit door.

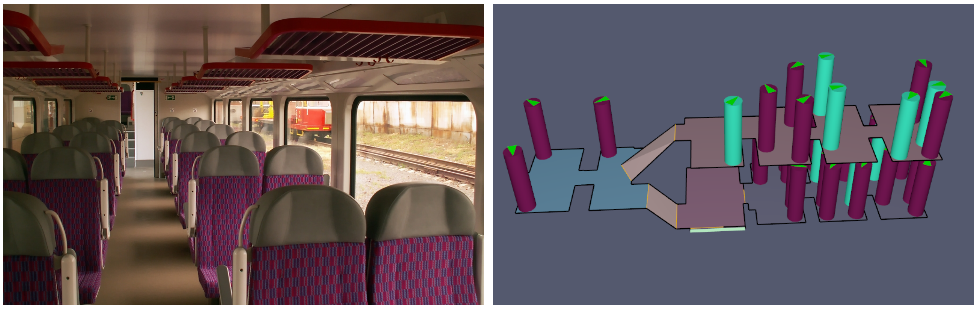


Figure 4.1: CityElefant train used in [1] and its Pathfinder model.

(Photo taken with permission from `https://www.railpage.net`)

4.1.2 Lecture Hall

The second geometry of choice is that of a lecture hall. There have been several experimental studies concerning lecture halls or theaters, for example, [56] conducted a series of evacuation trials at the Faculty of Engineering of Universiti Putra Malaysia, [57] did a similar experiment in China with emphasis on low visibility and developed a 3D extension of the social-force model for pedestrian modeling, and [58] focused on the exit choice of occupants in a monitored evacuation. The nature of the geometry with rows of seats arranged into a regular grid is similar to that of a cinema or a theater, which have also been the subject of various studies [59, 60, 61].

The lecture hall geometry used in this thesis is modeled according to the C-219 auditorium located at the Faculty of Civil Engineering, CTU in Prague. The auditorium can seat 73 people in seven rows. The first six rows have 11 seats each, and the last row near the exit has 7 seats. The rows are arranged on a slope, with the main desk and the first row at the lowest level. Each row is elevated by 21 cm from the previous one. The exit is located at the highest level of the room. There are two pairs of stairs, one on each side of the room. The Pathfinder model was provided by H. Najmanová. It will also be the location for an upcoming experimental evacuation experiment scheduled for June 5th, 2023.

There were minor complications while modeling the room, namely that sometimes three or more occupants would get stuck in the aisles, which was likely caused by a clash of priorities when one agent would attempt to go the long way around and join the queue from the other side, but there was a group of agents with different priorities in the aisle, effectively blocking him from getting to his target. This is a discrepancy in Pathfinder's implementation of tight geometries and is not realistic behavior. The issue was solved by prohibiting the occupants from using the second pair of stairs by adding a virtual wall at one end of each aisle. As lecture halls with only a single pair of stairs are still a valid design, this should not interfere with the findings of the

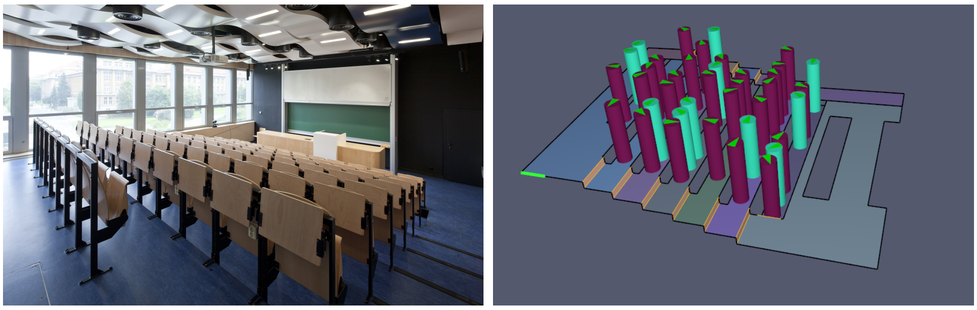


Figure 4.2: C-219 lecture hall at the Faculty of Civil Engineering, CTU in Prague, and its Pathfinder model.

(Photo taken from `https://portal.fsv.cvut.cz`)

analysis. An alternative solution could be obtained by allowing the occupants to "squeeze" themselves in the corridors, thus allowing the others to pass, but it is questionable whether this behavior is actually realistic.

4.2 Variable Selection

It is no surprise that the composition of the crowd has a major effect on evacuation time [62]. From a macroscopic point of view, the fundamental diagram has the form $J = \rho v$, which gives us three possibilities for manipulating the evacuation time:

- Adjusting J – Recall that $J = J_s w$, where J_s is the specific flow and w is the width of the exit. We could influence the evacuation time by considering a wider exit door. In general, doors cannot be expanded in a continuous manner. We could, at most, open a second wing of the door, but for this thesis, we shall consider the door width fixed.

- Adjusting ρ – The density in the room can be influenced by adding or removing occupants from the room. For practical purposes, the most interesting situations are those where the room is full, or almost full. Additionally, the perceived density can be controlled by increasing or decreasing the diameter of the occupants – when the pedestrians take up more space, fewer of them can fit into the room. However, in the previous study [55], the effect of the diameter (within reasonable values) proved to not be as influential.

- Adjusting v – The average velocity appears to be a good candidate for analysis. In real-life scenarios, we cannot influence the speed of the occupants, but in simulated environments, we have full control over the distribution of maximum velocities. Additionally, the acceleration times can be controlled and will also be taken to account for the analysis.

Another factor that clearly influences the evacuation time are the initial positions of the occupants. When the occupants are far from the exit, they have to cover more distance to reach the exit, thus taking longer to reach their destination. But when the density in the room is not uniform, i.e., the occupants are seated in clusters, undesirable effects such as excessive blocking can happen and lead to congestion. The true seating patterns of pedestrians depend on many factors including the type of venue or even cultural factors. For example, one study found that in lecture halls, the students tend to prefer the left-hand side of the rows [63], while in cinemas, the right-hand side is preferred [64]. The situation in trains is even more complicated due to the irregularity of the geometry. For this reason, I will stick to descriptive measures of a given seating configuration such as the average distance to the exit or the amount of "clustering" present.

Lastly, following the previous CTU studies, the heterogeneity of the crowd, represented by a portion of slower occupants, will be explored. The study in [65] showed a large decrease in average flow with the inclusion of multiple slow-walking pedestrians, which the experiments in this thesis will attempt to recreate. The simulated occupants will be divided into two groups – a group of able-bodied individuals and a group with movement limitations simulated by a slower maximum walking speed and higher acceleration time. The average difference in velocities between a group of younger and a group of older pedestrians in open space was observed to be approximately 0.26 m/s [66] (1.51 m/s vs. 1.25 m/s). The same study also measured a start-up time, which is the time it takes a pedestrian to start moving after the traffic light turns green, and found the average difference in reaction time of about 0.55 s (1.93 s vs. 2.48 s). The reaction time of Pathfinder agents is instant, therefore this quantity could be simulated by the acceleration time.

Chapter 5

Simulation Pipeline

The computation of sensitivity measures requires a large amount of data and collecting it by running the simulations manually would quickly prove tiresome. Pathfinder provides a basic Monte Carlo functionality that allows for randomizing the maximum velocities and initial positions of the occupants, but for more complex scenarios, finer control is needed. This chapter serves as a documentation of the Python package `simtools` developed for the purposes of this thesis which allows for fine control over the simulation pipeline. Originally, the simulation pipeline was a single command line utility that would run a number of simulations and collect the results, but with the increasing number of different types of experiments, each with a different set of parameters, maintaining a single script became very difficult. For this reason, I have chosen a more API-oriented approach that provides a simple interface over Pathfinder's command line tool and is heavily customizable, allowing the user to focus on designing the experiments instead of the technicalities connected to processing configuration files.

5.1 High-Level Overview

After creating a model in Pathfinder's graphical user interface and running the model, a configuration file is generated which is then used by Pathfinder to start the simulation process. This configuration file contains all the information necessary to conduct a simulation, including global parameters and metadata, the entire geometry, or the occupants themselves. The `simtools` package contains a class named `Simulation`, which processes the Pathfinder configuration file and creates an internal representation of the information provided within the input file. This includes extracting the defined profiles, the probability distributions that they reference, or the occupant properties along with their initial positions. A simulation with the parameters provided by the configuration file can then be run with the method `run_simulation()`, which starts Pathfinder and returns a `SimulationResult` object that contains the results of the simulation – basic statistics such as the total evacuation time, average evacuation time (with respect to the occupants), standard deviation of the evacuation times; and details about each occupant (initial position, maximum velocity, exit time, and others). In its simplest form, a simulation can be run in the following way:

```python
from simtools.simulation import Simulation
config = './config.txt'
sim = Simulation(config, out_dir='out')
results = sim.run_simulation()
print(f'Evacuation time: {results.max_evac_time:.2f} s.')
```

The resulting TET will be displayed and the results including the animation of the process will be in the `./out` directory.

However, this functionality is also supported by Pathfinder's Monte Carlo framework. The main added functionality comes in the form of *occupant variables* and *position generators*. Occupant variables allow the user to specify the quantity of interest that should be generated for each occupant prior to the simulation. The values can be specified as constants or via their distributions. There is also a special type of variable – `RandomnessVariable`, which processes the purely random parameters of the occupants (random seeds, priorities, and initial directions). Likewise, position generators provide a way of specifying how to generate occupant positions. By default, new positions are generated randomly from a given list of positions, which can be extracted from the configuration file or provided by the user. The variables and generators can be created before the simulation and then periodically updated with new values. The individual sources of randomness can also be easily controlled via methods that can, for example, re-generate only the random seeds for each occupant or only their positions. A diagram of the inputs and the outputs is shown in Figure 5.1. The example code in Listing 1 runs a total of 60 simulations in the train geometry with three different configurations of initial positions and two different maximum velocities, each averaged over 10 different random seed values. The available initial positions are in a separate file called `positions.csv`.

This approach is very general because it aims to be as reusable as possible. There are countless possible experimental designs and hand-picking a small number of them would force the users into using a predefined structure that may or may not fit their use case. Even so, a simple command line tool for convenience is also available for those who do not have programming experience or do not have the time to spend acquainting themselves with a new API. The following sections will provide a comprehensive overview of the `simtools` package and its features without going too deeply into the implementation details.

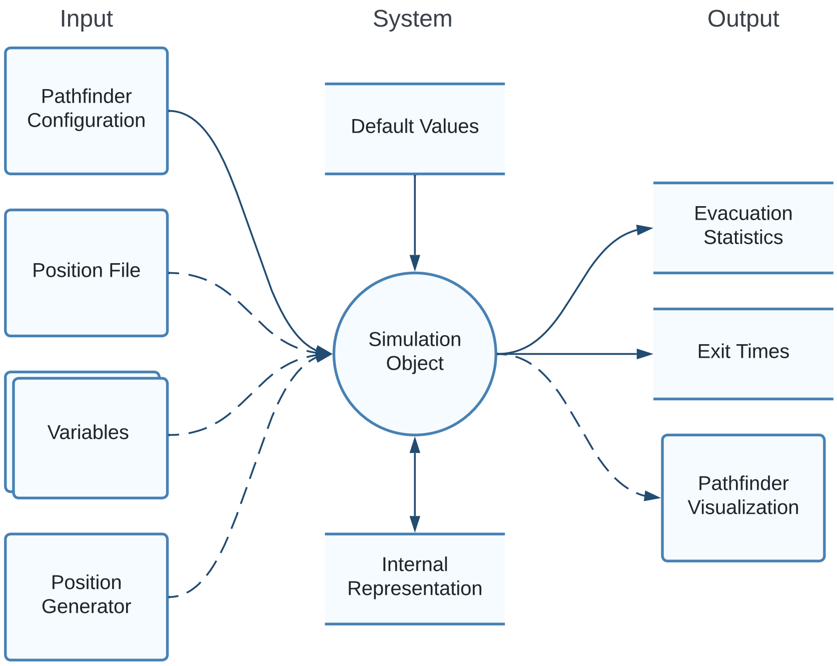


Figure 5.1: The inputs and outputs of the simulation object. Dashed lines denote optional arguments.

5.2 Simulation Object

The simulation object is represented by the class `Simulation` and it is the main controller of the simulation pipeline. It collects all the other auxiliary objects and calls their appropriate methods when they are needed. Its tasks include:

- Processing the configuration file.
- Preparing and maintaining the directory structure.
- Managing the individual variables.
- Generating occupants and processing them via other objects.
- Creating modified temporary configuration files.
- Running the simulations and collecting the results.

```python
# Import the necessary classes.
from simtools.simulation import Simulation
from simtools.variable import VelocityVariable

# Define the experiment.
NUM_OCCUPANTS = 30
NUM_POSITIONS = 3
VELOCITIES = [1.0, 1.2]
NUM_ITERATIONS = 10

# Initialize the variables and the simulation object
vel_var = VelocityVariable()
sim = Simulation(config='train.txt',
                 positions='positions.csv',
                 variables=[vel_var])

# Generate occupants and start the loop.
sim.generate_occupants(NUM_OCCUPANTS)
for i in range(NUM_POSITIONS):
    # Generate new positions.
    sim.regenerate_positions()
    for velocity in VELOCITIES:
        # Update the variable object and generate velocities.
        vel_var.to_constant(velocity)
        sim.regenerate_variables()
        for j in range(NUM_ITERATIONS):
            # Generate new seeds.
            sim.regenerate_randomness()
            # Run the simulation and show the results.
            results = sim.run_simulation()
            print(f'TET: {results.max_evac_time:.2f} s.')
```

Listing 1: A more complex simulation script.

Initialization The object is initialized with the input file generated by Pathfinder, whose structure is described in the technical reference [5]. The sections `occupants`, `profiles`, and `distributions` are all extracted and converted into a Python dictionary that contains all the defined attributes from their configuration section. Additionally, the distributions are converted from their text descriptors into SciPy distribution objects, for example, if the distribution with the key “`1`” corresponds to a truncated normal distribution, it is converted into a `scipy.stats.truncnorm` object. If a CSV file with the available initial positions is provided, it is processed and saved. Otherwise, the positions are extracted from the `occupant.loc` attributes in the occupant section. Finally, the directory structure is prepared. If the argument `out_dir` is provided, this directory is created and will house the results of the simulation, otherwise, the current directory is used (it can also be changed later via the `set_out_dir()` method).

Managing occupants and running simulations New occupants can be generated with the method `generate_occupants()`. If the input is an integer N, then N occupants with the default profile are created. Otherwise, the input is assumed to contain per-profile numbers, either as a list with the form `[N_1,...,N_p]`, where `p` is the total number of profiles in the configuration file, or as a dictionary with the form {`P_i:N_i`}, i.e., profile name and number of occupants key–value pairs. If the method is not called, then the occupants already present are used.

Once the occupants are generated, they can be further processed with respect to the individual sources of randomness – initial positions, physical parameters, or random seeds. Each of these components has a corresponding `regenerate_[SOURCE]()` method. The initial positions are handled by the position generators (Sec. 5.5), physical parameters are handled by the occupant variables (Sec. 5.4), and random seeds are handled by randomness variables (also Sec. 5.4). The occupants can be retrieved with the `get_occupants()` method or the `generate_and_return_occupants()` method.

Additionally, the occupants and their positions can be modified directly via a `set_occupant_variables()` call which accepts a dictionary with the form {`key: list_of_values`}, for example, assuming there are three occupants available,

```
sim.set_occupant_variables({
    "OccProfile.MAXVEL": [1.0, 1.2, 1.1]
})
```

will set their velocities to 1.0, 1.2, and 1.1. This is mainly useful when the values are pre-computed. However, this option should be used with caution as there are no checks performed as to whether the different keys are meaningful and an error will cause the simulation to fail.

The simulation can then run with the `run_simulation()` method which generates a temporary configuration file with the current list of occupants and saves the results in the specified directory. There is also a convenience `generate_and_run_sim()` method which performs all of the above at once.

5.3 Simulation Results

The results of a single simulation are represented by the `SimulationResults` data class. It is simply an object which contains the results of the simulation and can be processed by the user according to their needs. The following statistics are included:

max_evac_time – Evacuation time of the last occupant to exit (TET).

min_evac_time – Evacuation time of the first occupant to exit.

average_evac_time – Average evacuation time w.r.t. the occupants.

evac_time_std – Std. dev. of the evacuation times w.r.t. the occupants.

cpu_time, startup_time – Metadata about how long the process took.

occupant_stats – A Pandas data frame that contains the exit times for the individual occupants and the total distance they traveled. Their initial positions and profiles are also included.

The rest of the results including the Pathfinder visualization file can be found in the specified output directory.

5.4 Occupant Variables

Occupant variables represent the quantities under analysis such as velocity or acceleration. They are represented by the `OccupantVariable` class, which defines a single method `process_occupant()` used by the simulation object, and is meant for subclassing. The method takes a single occupant in the form of a dictionary with the keys and values linked to Pathfinder's occupant descriptors and can perform arbitrary operations over the said dictionary, assuming that once it modifies the dictionary, it will contain the valid definition of an occupant.

There are multiple predefined subclasses of `OccupantVariable` which are ready for use. A special one is `RandomnessVariable` which handles the parameters that govern the inherent randomness of the model, namely random seeds, priorities, and initial orientations. It can also be redefined in case the user wants to handle the inherent randomness differently.

The main subclass meant for use is the `ScalarOccupantVariable` which processes a single key for each occupant. It is initialized with the corresponding Pathfinder key and a distribution from which to generate the values. The default available distributions are constant, (truncated) normal, and uniform, but any discrete or continuous SciPy distribution can be used. For example, the definition

```
acc_var = ScalarOccupantVariable(key='OccProfile.ACCEL_TIME',
                                 distribution=1.2)
...
acc_var.to_normal(mean=1.0, std=0.2)
```

will create a variable that controls the acceleration time with a constant value of 1.2 s and later convert to a normally distributed variable with $\mu = 1.0\,\mathrm{s}$ and $\sigma = 0.2\,\mathrm{s}$. There is also a `VectorOccupantVariable` that works with multiple keys and distributions at once.

Once the variables are created, they have to be passed to the simulation object. This can be either done directly in the object's constructor via the parameter `variables`, or at any later point via the `register_occupant_variables()` method.

A simplified variable creation process is also possible. The variables can be inferred directly from the configuration file. This is done by specifying the variables of interest in the constructor of the simulation object with the keyword `infer_variables` which accepts a list of Pathfinder variable keys. Their respective distributions are then extracted from the `distributions` section. In the following example, the occupants' velocities and diameters are automatically processed from the configuration file.

```
sim = Simulation(
    'config.txt',
     infer_variables=['OccProfile.MAXVEL',
                      'OccProfile.DIAMETER']
 )
```

Working with profiles is also supported using `ProfiledOccupantVariable` which also takes a `profile` argument (the ID of the profile) and simply ignores all the occupants with different profiles.

5.5 Position Generators

Generating new positions is handled by the position generators. The base class `PositionGenerator` also defines a single method `generate_positions()` which takes in the number of requested positions (either a single integer or a list of per-profile numbers), a data frame of available positions, and, optionally, the dictionary of all the occupants in case the calculation requires them. The method is expected to return a simple list of the generated positions (their x, y, and z coordinates).

There is a predefined `DefaultPositionGenerator` which simply returns N random positions from the list. The geometries used in this thesis also have their own parametrized generators which will be described in the next chapter.

5.6 Working with Profiles

As was mentioned before, working with occupant profiles is fully supported. The profiles are automatically extracted from the configuration file when initializing the simulation object. All the methods that generate occupants also accept a `profiles` argument which allows the user to generate a given number of occupants from each profile, utilizing the values and distributions that they defined in Pathfinder GUI. Below is a code that will run 10 simulations, each with 10 occupants of profile 0 and 5 occupants of profile 1, which have different average velocities.

```python
vel0 = ProfiledVelocityVariable(profile=0).to_uniform(1.0, 1.2)
vel1 = ProfiledVelocityVariable(profile=1).to_uniform(0.6, 0.8)
sim = Simulation('config.txt', variables=[vel0, vel1])
for i in range(10):
    results = sim.generate_and_run_sim(profiles={0: 10, 1: 5})
    print(f'Evacuation time: {results.max_evac_time}')
```

5.7 Reproducibility

In order for the experimental results to be reproducible, fine control over random seeds used to generate the parameters is possible. Each occupant variable and position generator has a method called `set_random_state()` which updates its internal random generator with the provided seed and ensures that the values it generates will be reproducible. These methods are also accessible directly from the simulation object.

A particularly convenient use case for setting random states is fixing a set of parameters. For example, if one wanted to evaluate the model with 10 fixed sets of different velocities and 10 fixed sets of positions, one could generate them beforehand and set them manually before running each simulation. But this would require needlessly saving and loading whole matrices of data. Instead, one could simply generate 20 integer seeds and use them as random states. An example is shown below.

```python
sim = Simulation('config.txt', variables=[VelocityVariable()])
for vel_seed in range(10):
    for pos_seed in range(10):
        sim.set_variable_random_state(vel_seed)
        sim.set_positions_random_state(pos_seed)
        results = sim.generate_and_run_sim(num_occupants=30)
        print(f'Evacuation time: {results.max_evac_time}')
```

Internally, each variable owns a special `RandomState` object, which generates a sequence of pseudo-random integers that are then used to seed the individual distributions while generating values. This is to ensure that the different objects do not influence each other's random generators as would happen if the seeds were set globally.

5.8 Caveats and Future Improvements

The main downside of the package is that it is somewhat limited to Pathfinder. The concepts of occupant variables and position generators are simulator-agnostic, but the simulation object is not, although it could certainly be extended to work with other simulators.

From the feature side, working with occupant behaviors is currently impossible as there was no use case for them in this thesis. They could be implemented similarly to profiles, but the complexity of the implementation would increase since there would now be many profile–behavior combinations that would have to be handled unless behaviors were linked directly to profiles, which is not always the intended use.

Directly modifying the geometry of the simulation is also not implemented at the moment. Controlling, for example, the door widths or the properties of the stairs is certainly an interesting topic, but it would require creating an internal representation of the navigation mesh, which is not trivial.

Finally, there may still be bugs or design errors in the implementation as software engineering is not the primary aim of this thesis.

Chapter 6

Empirical Findings

This chapter will employ the methods introduced in Chapter 3 to conduct multiple experiments focused on analyzing the different sources of randomness and their effect on the total evacuation time with respect to the geometries described in Chapter 4. For the remainder of the thesis, I will be considering the following four scenarios:

- Lecture hall with 38 occupants (approximately at 50 % capacity).
- Lecture hall with 68 occupants (approximately at 90 % capacity).
- Train with 34 occupants (approximately at 50 % capacity).
- Train with 60 occupants (approximately at 90 % capacity).

I have chosen to keep the number of occupants fixed because preliminary analysis showed that it is by far the most important factor whose effect would drown out the effects of the more subtle variables. Besides, from a practical point of view, only the situations where the facility is near full capacity are interesting. The instances at 50 % occupancy are included so that the effect of initial positions can be studied and the ones at 90 % capacity were chosen to provide a more practical insight into the problem without completely disregarding the effect of initial positions. However, the ratio of able-bodied to movement-impaired occupants is not fixed and will be included in the analysis as a separate variable.

As for the variable choice, the effects of the following groups of variables will be analyzed (AB stands for able-bodied and WL for with-limitations):

- Ratio of WL to AB occupants r.
- Parameters of AB group – velocity v_{ab} and acceleration a_{ab}.
- Parameters of WL group – velocity v_{wl} and acceleration a_{wl}.

- Initial positions of all the occupants – exit-attraction coefficient $\alpha_{\text{all}}^{\text{ex}}$ and occupant-attraction coefficient $\alpha_{\text{all}}^{\text{occ}}$ (will be defined in Section 6.3).
- Initial positions of WL group – exit-attraction coefficient $\alpha_{\text{wl}}^{\text{ex}}$ and occupant-attraction coefficient $\alpha_{\text{wl}}^{\text{occ}}$.

A baseline velocity of the AB individuals will be based on Pathfinder's default value (1.19 m/s) and WL individuals will be roughly 30 % slower. The intervals for the sensitivity analysis will be built around this baseline value. The acceleration of the AB group will be slightly faster than the default value of 1.1 s at approximately 0.9 s and the WL group will, again, take roughly 30 % more time to accelerate. The ratio of WL to AB occupants will vary between 10 % and 30 %. Additionally, given a combination of v_{ab}, a_{ab}, v_{wl}, and a_{wl}, these values will be shared among all occupants of its type. More realistically, they could be generated from distributions with means at the given values, however, this proved to needlessly increase the variance of the individual methods and was later reconsidered. The exception is Chapter 6.2, where this variance is studied.

First, I will provide a brief overview of the previous use of sensitivity analysis in evacuation dynamics. Then, a short analysis of the uncertainty induced by different velocities, initial positions, and the stochasticity of the model will follow. Next, we will take a more detailed look at the effects of initial positions on a toy model and design a parametrized way to generate initial positions. After that, the actual sensitivity analysis will take place with an initial screening of the variables via elementary effects followed by the computation of Sobol' indices and interpretation of the results. The chapter will conclude with a comparison of different surrogate modeling techniques and how their results differ from the ones computed by the standard method.

6.1 Previous Work

Although not particularly common, there have been some uses of sensitivity analysis in evacuation dynamics. Aside from the articles that directly preceded this thesis [1, 55], sensitivity analysis was used in [67] to study the relationship between model parameters (average velocity, number of occupants, "personal space" and others) and maximum density in a simulated setting. The authors of the article also note that the use of SA techniques in pedestrian dynamics is very sparse. In [68], a sensitivity analysis of the evacuation safety level with respect to the model parameters of a CFAST fire and smoke growth model was conducted. A similar analysis that focused on traffic safety was performed in [69]. Finally, there was another study presented at the FEMTC 2022 conference that employed sensitivity analysis of evacuation time with respect to occupant speed, size, and pre-movement time, and compared them across different simulators [70]. They found that occupant speed dominated the other factors in its influence over evacuation time.

6.2 Randomness Study

Since the methods of sensitivity analysis expect a deterministic model, it is necessary to average multiple model runs with the same parameters in order to suppress the inherent stochasticity of the model. However, it would be good to know how significant the purely random effects are. We can obtain a general estimate of their strength via the decomposition generated by the Law of total variance from Section 3.3.2:

$$\operatorname{Var}(Y) = \mathbb{E}[\operatorname{Var}(Y \mid \boldsymbol{X})] + \operatorname{Var}(\mathbb{E}[Y \mid \boldsymbol{X}]). \tag{6.1}$$

The other sources of randomness can be traced to the agent parameters and the initial positions, since we consider the geometries and the numbers of occupants fixed. To get a baseline idea of the amount of uncertainty in the output attributed to each of these sources of randomness, I have generated 16 different random seating configurations and 16 different maximum velocity configurations for every scenario defined at the beginning of the chapter. The uncertainty in the velocities, in this case, is represented by random sampling from a (truncated) normal distribution, i.e., different velocity configurations correspond to different realizations of the random variable. Each scenario has approximately 20 % of movement-impaired occupants scattered uniformly throughout. The velocities of the able-bodied occupants were generated from $\mathcal{N}(1.25, 0.1^2)$ capped at 1.0 and 1.5. The movement-impaired individuals had their velocities generated from $\mathcal{N}(0.8, 0.1^2)$ and capped at 0.6 and 1.0. Each simulation was run 10 times with different seeds and priorities to obtain a sample of the inherent randomness in each point, obtaining a $16 \times 16 \times 10$ tensor. The variance decomposition can be computed directly from the formula above. The results are shown in Table 6.1.

Table 6.1: Decomposition due to the Law of total variance w.r.t. different velocity and position configurations. The units are squared seconds.

Scenario	Total variance	Unexplained variance		Explained variance	
		Absolute	Percentage	Absolute	Percentage
hall_38	1.738^2	0.871^2	25.106 %	1.504^2	74.894 %
hall_68	1.932^2	1.466^2	57.571 %	1.258^2	42.429 %
train_34	2.610^2	1.110^2	18.077 %	2.362^2	81.923 %
train_60	2.410^2	1.849^2	58.829 %	1.547^2	41.171 %

It is clear from the results that the effect of random seeds on the output can be significant, especially with higher densities. This can be expected, as when the facility is near its full capacity, most of the seats are taken and when congestion happens, most occupants move at a pace much lower than their maximum speed. Also, the total variance for the train scenario is significantly higher than for the lecture hall scenario. This can be explained by the larger complexity of the geometry where different starting positions can have very different corresponding exit distances. However, these positions and velocities represent the "average" scenario. The situation could be vastly different if the parameters were generated from different distributions.

Additionally, we can further partition the variances induced by the positions and the velocities via variance-based sensitivity analysis. Since the individual seating and velocity configurations are fixed, we can condition on them and manually compute the Sobol' sensitivity indices directly from their definition in (3.14). The results are shown in Table 6.2.

Table 6.2: Manually computed sensitivity indices w.r.t. randomly generated positions and maximum velocities.

Scenario	First-order effect S_i		Total effect T_i		Interaction S_{ij}
	Positions	Velocities	Positions	Velocities	
hall_38	0.424	0.209	0.791	0.576	0.366
hall_68	0.263	0.381	0.619	0.737	0.356
train_34	0.851	0.053	0.947	0.149	0.096
train_60	0.428	0.263	0.737	0.572	0.309

In this case, the positions generally have a larger effect on the outcome, with the exception of the nearly-full lecture hall. This observation is especially apparent with the half-empty train where almost all of the output variance can be attributed to the initial positions. This leads us to the natural hypothesis that the effect of initial positions decreases as the number of occupants increases and that the train geometry is more sensitive to the initial positions than the lecture hall geometry. As a sanity check, we can also observe the various properties of the sensitivity indices mentioned in Section 3.3.2, namely that the total effect T_i is equal to the sum of all the terms that contain i, and that the sum of the first-order and second-order indices is equal to 1 (except for rounding errors).

6.3 Effects of Initial Positions

Intuitively, the initial positions of the occupants should play a significant role in determining the evacuation time – when the occupants have to travel more distance, it is bound to take them more time. This property can easily be quantified by the average distance to the exit. However, this single number does not provide enough information. For example, in a lecture hall, if all the students were sat in a single row very close to the exit, their average exit distance would be low, but the evacuation time could easily be longer than if they were more evenly dispersed across the rows due to the high number of occupant-to-occupant interactions. Furthermore, if they were uniformly scattered throughout the room, we would get the same average exit distance as if they were in two large clusters, one close to the exit and one far from the exit, even though these are clearly different situations. It is not immediately clear how the "energy" of a seating configuration can be quantified in an interpretable manner. The "energy" in this context refers to a single number or a small set of numbers which describe a the properties of a given seating configuration.

The first measure I have chosen is the aforementioned average exit distance. The second one, in an attempt to add a different dimension to the analysis, is the average pairwise distance between occupants – when this number is low, the occupants are seated in a single large cluster, and the higher this number gets, the further away (on average) the occupants are from each other. The advantage of these measures is that they can be used to effectively generate seating configurations with the required properties, which is necessary for sensitivity analysis. The generation process will be described shortly. I have also experimented with several different statistics such as the variances of the exit distances or distances to k nearest neighbors but was unable to implement a proper way of generating such configurations.

Additionally, in the presence of a mixed crowd with a group of movement-impaired individuals, it is necessary to think about the initial positions on two levels – first, which positions are occupied, and second, which of these positions are taken by the slower occupants. The measures described in the previous paragraph can be applied recursively – first, they are computed for all the selected positions to obtain a global measure, and then, they are computed only for the movement-impaired group, obtaining a localized measure.

And lastly, there are two seemingly incompatible sides of the problem. On one hand, the statistics described above are computed *ex-post* – after the configuration has been generated. On the other hand, if we want to use them as input to sensitivity analysis, we need to be able to specify them *ex-ante* and only generate the configuration after the fact. The first approach would be preferable, however, the distributions of such statistics are apriori unknown and will depend on many factors including the specifics of the geometry or the number of occupants. In order to be able to generate initial positions with the desired properties, I have tried the following two approaches:

- A Monte Carlo approach in which a large number of random configurations is generated and the ones closest to the desired values (with respect to the L_∞ norm) are chosen. To keep the values in a fixed range, the values were normalized to lie approximately in $[-1, 1]$ range.

- A probabilistic approach in which the seats are sequentially chosen with probabilities proportional to their distance to the exit and to other occupants. The strength of the attraction is determined by two input coefficients, one for the exit attraction and one for the occupant-to-occupant attraction.

The first method was unsuccessful for several reasons. Firstly, it was computationally expensive, and even with the use of parallel processing it significantly slowed down the simulation time. Secondly, the range of the distances highly depends on the geometry and the number of occupants. In order to get it into a reasonable range I had to normalize it via a low and a high quantile to account for possibly skewing the range by outliers. This worked somewhat well because of the normality due to the Central limit theorem, but there were still instances where the process was unstable, especially with the train geometry. Thirdly, certain combinations of the inputs were highly correlated, for example, if we want the occupants to be very close to the exit, we cannot expect their pairwise distances to be arbitrarily high. This would also break the theoretical properties of the SA sampling process as it expects uncorrelated inputs. The second method was more successful and it will be described in the next section.

6.3.1 Generating Positions

With the two measures given above in mind (average distance to the exit and average inter-occupant distance), the goal is to create two proxy variables with a bounded range that can be used as input to the position generator and consequently to the sensitivity analysis. The configurations that it produces need to match the expected characteristics, at least on average.

The approach that I've chosen uses two *attraction coefficients* as the input variables – the exit attraction coefficient α^{ex} and the occupant attraction coefficient α^{occ}. Essentially, the exit and the individual occupants all generate a scalar field whose magnitude at each point is proportional to the distance from the source of the field scaled by the attraction coefficient. If these coefficients are positive then the exit and the individual occupants produce an attraction force and when the coefficients are negative, they produce a repulsion force. When the coefficient is zero, no force is generated. The resulting probability of selecting a free seat is obtained as a superposition of all the forces acting on it. For example, if the exit attraction coefficient is high, then the seats close to the exit have a high probability of being selected.

The seating configuration is generated sequentially, i.e., seat by seat. The field generated by exit attraction is fixed throughout the generation process, but the one generated by occupant attraction changes over time. The configuration is initially empty and the force field is given only by the exit attraction. Then, a seat is chosen which begins to generate an attraction field. The probabilities are recomputed and another seat is chosen. This continues until the required number of seats has been generated. Since there may be multiple occupants generating a field but only a single exit, the forces generated by the individual occupants are normalized by the current number of occupants in each time step.

Formally, given a distance metric d, an exit point s^*, and a set of n selected seats $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$, the force F acting on a free seat $\boldsymbol{s} \in S$ is obtained as

$$F(\boldsymbol{s}) = -\alpha^{\text{ex}} d(\boldsymbol{s}, \boldsymbol{s}^*) - \alpha^{\text{occ}} \sum_{i=1}^{n} \frac{d(\boldsymbol{s}, \boldsymbol{s}_i)}{n}. \tag{6.2}$$

The probability of selecting the seat $\boldsymbol{s}$ is then given by

$$p(\boldsymbol{s}) = \frac{\exp\left(F(\boldsymbol{s})\right)}{\sum_{\boldsymbol{s}' \in S} \exp\left(F(\boldsymbol{s}')\right)}, \tag{6.3}$$

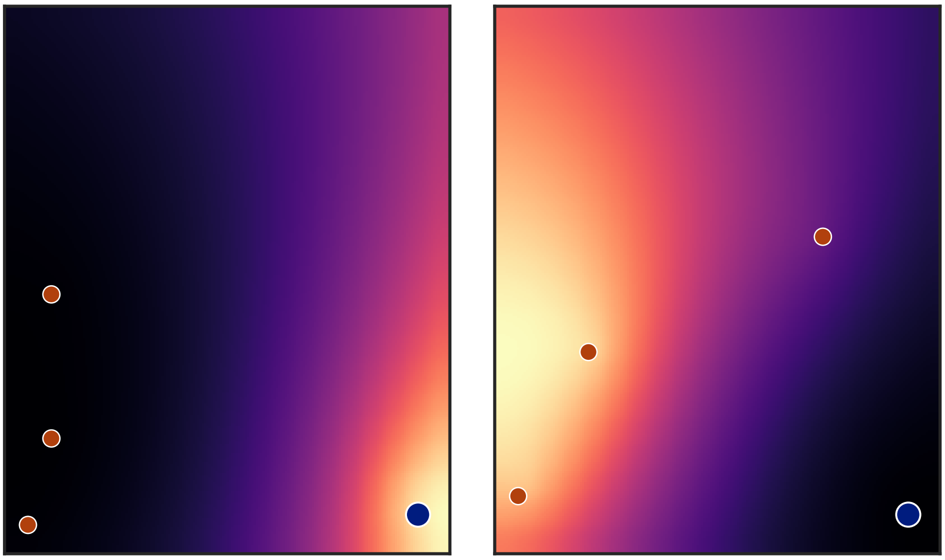
which is coincidentally the softmax function over the forces acting on each free seat. I have also experimented with using absolute and quadratic distance-decay, but the exponential decay showed the best empirical results. An example of the probability field generated by (6.3) in a continuous space is shown in 6.1. In this case, a simple Euclidean distance is used in a rectangular grid with dimensions similar to those of the lecture hall setting.

The selection of the distance metric d is crucial as it dictates which seats are considered close to one another. For example, in the train scenario, two passengers who are on different decks may be relatively close together with respect to Euclidean distance, but in reality, they do not interact at all. For this reason, I have implemented two custom distance metrics, one for each geometry, which simulate the true distance the occupant has to travel in order to reach the other point. This is done via checkpoints – for example, if the occupant is seated in a different row in the lecture hall, he first has to reach the stairs, then reach the other row, and only then can he reach the point of interest.

Additionally, the ranges for the coefficients α^{ex} and α^{occ} have to be fine-tuned for each geometry, because past a certain threshold, the attraction (or repulsion) becomes too strong, and the softmax function becomes saturated, returning a probability close to 1 for a single seat and almost 0 for all the others. Through empirical testing, I have determined the suitable ranges for both coefficients as

- $[-0.4, 0.4]$ for train geometry, and
- $[-0.6, 0.6]$ for lecture hall geometry.

The endpoints of these intervals cause the process to generate configurations close to the extremal cases described above and values from inside the intervals cause an almost linear transition between these extremes.



(a) $\alpha^{\text{ex}} = 0.2$, $\alpha^{\text{occ}} = -0.3$ (b) $\alpha^{\text{ex}} = -0.3$, $\alpha^{\text{occ}} = 0.4$

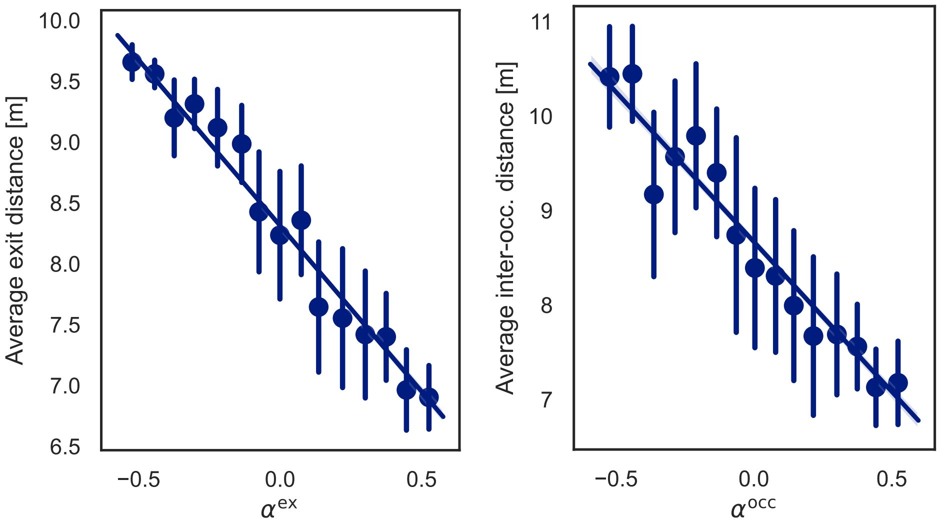
Figure 6.1: Examples of a (continuous) probability field generated by (6.3). The blue point is the exit and the orange points are the already-generated occupants. The brightness of the color is proportional to the probability of selecting a seat in said location. The setup on the left is designed to generate positions rather close to the exit but far away from the other occupants. The setup on the right will do the opposite. Note that the actual seating positions are discrete, this figure only serves to illustrate the interactions between the points.

The entire process as an algorithm is shown in Algorithm 1. The same procedure can be applied both when selecting the positions for all the occupants and when selecting the positions of the movement-impaired individuals.

The last thing that needs to be checked is whether the configurations generated this way actually have the properties that we expect, e.g., whether high values of α^{ex} truly correspond to configurations where occupants are close to the exit. Figure 6.2 shows a scatter plot of the input values of α^{ex} and α^{occ} with their average exit distances and average inter-occupant distances computed ex-post for approximately 600 configurations (hall geometry with 38 occupants). There is a clear correlation between the input coefficients and the properties of the sampled configurations. Some variance is present due to the stochasticity of the process, but on average, a linear trend can be observed, meaning that the α^{ex} and α^{occ} are a reasonable substitute for the desired statistics.

Algorithm 1 Position generation.

```
Input: Available positions $A$, Requested sample size $N$; $\alpha^{\text{ex}}$ and $\alpha^{\text{occ}}$
Output: Sampled positions $S$
function SAMPLE_POSITIONS($N$, $A$)
    $S \leftarrow \{\}$
    for $i \leftarrow 1$ to $N$ do
        $P(A) \leftarrow \{p(a; \alpha^{\text{ex}}, \alpha^{\text{occ}}), \forall a \in A\}$        ▷ Equation (6.3).
        $a_s \leftarrow$ SAMPLE($A$, $P(A)$)        ▷ Sample $A$ w.r.t. probabilities $P(A)$.
        $S \leftarrow S \cup \{a_s\}$
        $A \leftarrow A \setminus \{a_s\}$
    return $S$
```



(a) Exit distance. $R^2 = 0.84$.

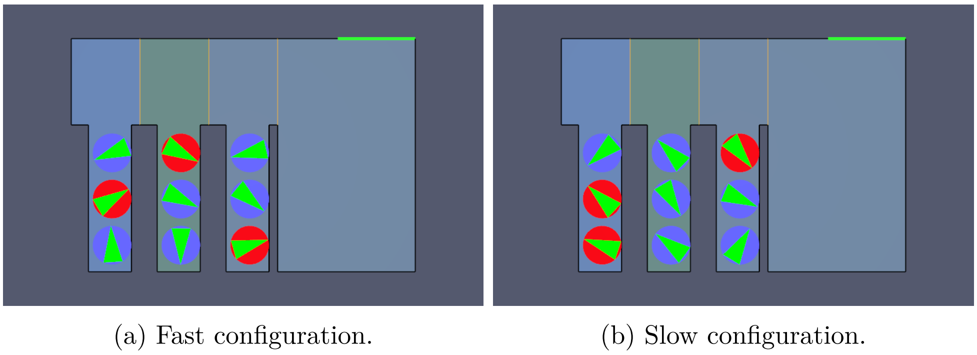
(b) Inter-occupant distance. $R^2 = 0.68$.

Figure 6.2: The correlation between the α coefficients and the statistics computed on the generated position configurations (lecture hall with 38 occupants). For a cleaner plot, the data was binned into 15 bins and the average of each bin as well as the standard deviation is shown.

6.3.2 Movement-Impaired Individuals

So far, we've only looked at the effects of the initial positions of all the occupants as a whole. But the problem also has a second dimension – which positions are taken by the individuals with movement impairments. This factor could be especially important when the facility is almost at full capacity and there is a large number of WL occupants.

As part of the preliminary analysis, I have modeled a down-scaled version of the lecture hall with three rows of three seats. A screenshot of this geometry is shown in Figure 6.3. With such a small geometry, it is possible to evaluate all the different seating configurations with a simple brute-force approach. Given 6 AB and 3 WL individuals, there is a total of $\binom{9}{3} = 84$ possible seating configurations. I have generated all of these combinations and evaluated the model 10 times for each of them. The able-bodied individuals were modeled by a constant maximum velocity of $1.2\,\mathrm{m/s}$ and acceleration time of $0.9\,\mathrm{s}$. The movement-impaired individuals had a maximum velocity of $1.2\,\mathrm{m/s}$ and acceleration time of $1.3\,\mathrm{s}$.


(a) Fast configuration.

(b) Slow configuration.

Figure 6.3: Miniature lecture hall. Red occupants correspond to individuals with movement impairments.

The results of the simulations were averaged for every configuration and sorted by total evacuation time. We have seen in Section 6.2 that the variance induced by random seeds can be significant. To check that the rankings obtained by averaging are consistent, I have divided the 10 independent runs for each configuration into two disjoint sets of 5 points and averaged each of them separately to obtain two rankings R_1 and R_2. The correlation between two rankings can be measured by *Kendall rank correlation coefficient* (*Kendall's τ coefficient*) which is computed as

$$\tau(R_1, R_2) = \frac{\#(\text{concordant pairs}) - \#(\text{discordant pairs})}{\#(\text{all pairs})} \approx 0.824, \tag{6.4}$$

where $\#(\cdot)$ means "number of". A value of τ higher than 0.8 can be regarded as very high correlation, meaning that the estimates obtained by averaging 5 different runs should be robust. The rest of the analysis will be performed over the average of all 10 runs.

Of the 84 total instances, I have extracted the 10 fastest and 10 slowest configurations and computed the occupancy of each seat based on the type of occupant seated there. The results are shown in Table 6.3. For example, the cell Fast→WL shows the proportion of the 10 fastest runs in which the given seat was taken by a movement-impaired occupant. The main observation from this table is that in the fast configurations, the bottom row is mainly occupied by fast individuals, meaning that the slow ones are closer to the exit. However, the total number of AB occupants is larger than the number of WL occupants, meaning that these numbers can overestimate the effect of AB positions and underestimate the effect of WL positions.

Table 6.3: Estimates of $\Pr(\mathit{Occ\_type} = t \mid \mathit{Run} = r)$ for each seat. The exit is located near the top row.

Row	Seat	Fast		Slow	
		AB	WL	AB	WL
Top	Aisle	0.9	0.1	0.4	0.6
Top	Middle	0.5	0.5	0.9	0.1
Top	Window	0.4	0.6	0.7	0.3
Middle	Aisle	0.3	0.7	0.6	0.4
Middle	Middle	0.5	0.5	0.6	0.4
Middle	Window	0.8	0.2	0.8	0.2
Bottom	Aisle	0.7	0.3	1.0	0.0
Bottom	Middle	0.9	0.1	0.6	0.4
Bottom	Window	1.0	0.0	0.4	0.6

Denoting the type of occupant as $T = \{\text{ab}, \text{wl}\}$ and the type of the run as $R = \{\text{fast}, \text{slow}\}$, we can realize that Table 6.3 shows an estimate of the probability $\Pr(T = t \mid R = r)$ for each seat, i.e., the probability that the seat is taken by an AB or WL occupant, given that the run is fast or slow. Using Bayes' theorem, we can estimate the probability $\Pr(R = r \mid T = t)$ for each seat, i.e., the probability of a fast or slow run given that the seat is taken by a specific type of occupant:

$$\Pr(R = r \mid T = t) = \frac{\Pr(T = t \mid R = r)\Pr(R = r)}{\Pr(T = t)}. \tag{6.5}$$

The probabilities $\Pr(T = t)$ are equal to 2/3 for AB occupants and 1/3 for WL occupants, and the probabilities $\Pr(R = r)$ are equal to 10/84 in both cases since that's how they were constructed. The results of these transformations are shown in Table 6.4. From this properly normalized table we can make clearer observations:

- The fast runs are characterized by having movement-impaired occupants close to the exit as can be seen by the higher probabilities of a fast run if the top and middle seats contain a slow occupant. The exception lies in the seat closest to the exit – in this case the occupant becomes the first in the crowd, slowing everyone behind him.
- The slowest runs are caused by a specific scenario – there is a slow occupant in the spot closest to the exit as well as in the spot furthest

from the exit, which causes a significant slowdown in two ways – the one close to the exit slows the evacuation in the initial stages and the one far from the exit slows it down in the later stages. This can be observed from the animations of the process generated by Pathfinder.

Table 6.4: Estimates of $\Pr(Run = r \mid Occ\_type = t)$ for each seat.

Row	Seat	Fast		Slow	
		AB	WL	AB	WL
Top	Aisle	0.160	0.035	0.017	0.214
Top	Middle	0.089	0.178	0.160	0.035
Top	Window	0.071	0.214	0.125	0.107
Middle	Aisle	0.053	0.250	0.107	0.142
Middle	Middle	0.089	0.178	0.107	0.142
Middle	Window	0.142	0.071	0.142	0.071
Bottom	Aisle	0.125	0.107	0.178	0.000
Bottom	Middle	0.160	0.035	0.107	0.142
Bottom	Window	0.178	0.000	0.071	0.214

Granted that these findings are specific to the small Pathfinder model and it remains to be seen whether they will generalize to larger scenarios, the main takeaways are that placing the slower occupants closer to the exit leads to better performance, but only up to a certain point, past which the slow occupants start to slow down the rest of the crowd. Additionally, the slow occupants shouldn't be placed too far from the exit because it will take them too long to evacuate. This is in line with common safety-engineering practices.

6.4 Elementary Effects Analysis

This section will start the proper sensitivity analysis with the method of elementary effects described in Chapter 3.3.3. The main use of this method is for factor fixing, i.e., determining which input factors are *not* important, which can be done with a relatively small number of model evaluations. The use of elementary effects in this section will be two-fold:

1. With factors divided into groups based on the source of randomness that they represent.

2. With the individual factors, separately for each group.

The goal of the first level is to obtain high-level knowledge about the importance of each source of randomness with a significantly reduced number of necessary model evaluations. The factors described at the beginning of the chapter are divided into the following groups:

AB parameters – velocity of AB occupants, acceleration of AB occupants

WL parameters – velocity of WL occupants, acceleration of WL occupants

All positions – coefficient $\alpha_{\text{all}}^{\text{ex}}$, coefficient $\alpha_{\text{all}}^{\text{occ}}$

WL positions – coefficient $\alpha_{\text{wl}}^{\text{ex}}$, coefficient $\alpha_{\text{wl}}^{\text{occ}}$

Number of WL occupants – ratio r

The second level aims to obtain information about which factors are non-influential and could be left out of the analysis. This will be done separately for each group while the values of the other factors will be fixed in an OAT style. To mitigate the risks of OAT analysis, the process will be repeated for three carefully chosen points in the parameter space and the results averaged.

6.4.1 Groups of Factors

The cost of the EE method in terms of the number of model evaluations is $N_t \times L_t \times N_s$, where N_t is the number of requested trajectories, L_t is the length of each trajectory (6 in the case of 5 groups), and N_s is the number of samples to estimate $\mathbb{E}[TET_{\boldsymbol{X}}]$ with. Since the individual trajectories are independent, we can incrementally update the estimates of the sensitivity measures with more trajectories and stop the computation once they are stable. Note that there are now two levels of randomness in estimating $\mathbb{E}[TET_{\boldsymbol{X}}]$ – the first is in the random seeds and priorities of a fixed seating configuration as discussed before, but the second one arises in the seat-generating process, which is bound to add more variance to the estimate. To compensate for this extra uncertainty, the model will be run 10 times for each $\boldsymbol{X}$ configuration. The speed of convergence also depends on how finely the variables' input ranges are discretized. The standard number of levels for the method is 4, but for this experiment, I have chosen 6 levels to get slightly more accurate estimates.

With $N_t = 64$, $L_t = 6$, and $N_s = 10$, the cost of analysis is 3840 model runs for each of the four scenarios. Adjusted for simulation time, with approximately 5 seconds for each model run, the total time of the analysis is $T \approx 768\,000\,\text{s} \approx 21\,\text{h}$. The input ranges for each variable are shown in Table 6.5. All α coefficients are additionally scaled by a factor of 0.6 for the lecture hall scenario and 0.4 for the train scenario to account for the different average distances in each geometry.

When working with groups of factors, only the μ^* statistic is available. The resulting values of μ^* for each scenario are shown in Table 6.6. The following observations can be made:

- None of the parameter groups are outright non-influential.

Table 6.5: Input variables of EE method.

Variable	Group	Range	Distribution
v_{ab}	AB parameters	$[1.1, 1.4]$	Uniform
a_{ab}	AB parameters	$[0.7, 0.9]$	Uniform
v_{wl}	WL parameters	$[0.7, 1.0]$	Uniform
a_{wl}	WL parameters	$[1.0, 1.2]$	Uniform
$\alpha_{\text{all}}^{\text{ex}}$	All positions	$[-1.0, 1.0]$	Uniform
$\alpha_{\text{all}}^{\text{occ}}$	All positions	$[-1.0, 1.0]$	Uniform
$\alpha_{\text{wl}}^{\text{ex}}$	WL positions	$[-1.0, 1.0]$	Uniform
$\alpha_{\text{wl}}^{\text{occ}}$	WL positions	$[-1.0, 1.0]$	Uniform
r	Ratio	$[0.1, 0.3]$	Uniform

- AB parameters are influential in every scenario and their influence increases with the number of occupants. The ratio parameter behaves similarly.

- WL parameters and positions have, on average, the lowest, but still non-negligible effect. Their influence increases with a larger number of occupants.

- The train scenario is much more sensitive to initial positions than the lecture hall scenario.

In addition to these observations, we have also obtained a large sample of model outputs which we can use to train a surrogate model. These samples will be revisited in Chapter 6.6. It should be noted that the actual numerical values of the μ^*s must be interpreted with caution. For example, when a variable has a μ^* twice as large as a different variable, it does not necessarily mean that it is twice as important because its magnitude depends on multiple factors including the ranges of the variables and the number of levels.

6.4.2 Individual Variables

In order to judge the effects of the individual variables, the experiment from the previous section will be repeated separately with respect to each group. While one group of variables is being tested, the other variables will be fixed at the values corresponding to one of these scenarios:

ID 0 – Standard scenario.

ID 1 – Large difference between AB and WL group; WL group close to exit.

ID 2 – Small difference between AB and WL group; WL group far from exit.

Table 6.6: Estimated μ^* values along with the width of their 95 % confidence intervals obtained by bootstrapping.

(a) Lecture hall, 38 occupants.

Group	μ^*	95 %CI
AB parameters	5.04	0.72
WL parameters	1.54	0.27
All positions	5.54	0.83
WL positions	1.35	0.25
Ratio	3.75	0.45

(b) Lecture hall, 68 occupants.

Group	μ^*	95 %CI
AB parameters	8.52	1.22
WL parameters	2.46	0.39
All positions	1.36	0.24
WL positions	2.26	0.45
Ratio	6.71	0.65

(c) Train, 34 occupants.

Group	μ^*	95 %CI
AB parameters	4.15	0.76
WL parameters	2.63	0.42
All positions	14.64	1.64
WL positions	2.20	0.53
Ratio	3.79	0.61

(d) Train, 60 occupants.

Group	μ^*	95 %CI
AB parameters	6.59	0.88
WL parameters	4.18	0.50
All positions	5.84	0.80
WL positions	2.95	0.53
Ratio	5.27	0.64

The concrete parameter values corresponding to each scenario can be found in Table 6.7. The values of the αs are, again, scaled to match the appropriate geometry. The ranges of the input variables are the same as in the previous section (Table 6.5). The resulting elementary effects will be computed as the average of the values obtained for each scenario. For complexity reasons, each parameter configuration will be evaluated only 3 times. The total cost of the analysis is $50 \times 3 \times 3 \times 3 \times 4 = 5400$ model evaluations, where the numbers are (in order) the number of trajectories, the length of each trajectory, the number of independent samples, the number of scenarios described above, and the number of groups. Adjusting for simulation times and different settings, the approximate computation time is $T \approx 30\,\mathrm{h}$. For this experiment, only a four-level grid is used.

Table 6.7: Scenarios for EE computation w.r.t. the individual variables.

ID	v_{ab}	a_{ab}	v_{wl}	a_{wl}	$\alpha^{\mathrm{ex}}_{\mathrm{all}}$	$\alpha^{\mathrm{occ}}_{\mathrm{all}}$	$\alpha^{\mathrm{ex}}_{\mathrm{wl}}$	$\alpha^{\mathrm{occ}}_{\mathrm{wl}}$	r
0	1.2	1.1	0.9	1.4	0.0	0.0	0.0	0.0	0.2
1	1.4	0.9	0.8	1.5	0.5	−0.3	0.4	−0.2	0.3
2	1.1	1.0	1.0	1.3	−0.5	0.3	−0.4	0.2	0.2

When working with individual variables, the μ and σ measures are also available. The results are shown in Table 6.8. Note that only the within-group values should be compared as each group was processed independently. The data shows that:

- Velocity has a larger influence than acceleration in all cases.
- The positional parameters have similar effects on the output with the exit distance being slightly more important. The exception is in train scenario with 34 occupants, where the exit distance dominates.
- The pairwise interactions between variables indicated by σ values may be significant as they are often in the same magnitude as the μ^* values.
- The positions of the WL group have a non-monotonic effect on the output as evidenced by the large (absolute) difference between μ^* and μ.

As a result, the only variable we could fix is the acceleration, however, the small reduction in the cost of the analysis is likely not worth the loss of information.

6.5 Variance-Based Analysis

With the (relatively) quick preliminary analysis finished, we can move on to large-scale variance-based sensitivity analysis. To summarize the findings up to this point:

- The inherent stochasticity of the simulation model is non-negligible, especially in high-density situations.
- Physical properties of the majority group have a significant influence in all cases, with the average velocity being the major contributor.
- The relative number of movement-impaired individuals is likewise highly influential and its influence grows when the geometry becomes more crowded. The concrete parameters and positions of the WL group are not as important as their presence itself.
- There are considerable differences in the behavior of the system with respect to the individual settings. The train scenario is significantly more sensitive to the initial positions.

The computation of the sensitivity indices provided by SALib is implemented in the way described in Chapter 3.3.2, with a slight modification that allows for the estimation of the second-order interaction terms. The sample matrices **A** and **B** are generated via Sobol' sequences which provide favorable sampling properties if the number of samples N is a power of two, i.e., if $N = 2^n$ for some natural n. A benefit of SALib's implementation is that the number of samples can be incrementally increased (in powers of two) during

Table 6.8: Pairwise elementary effects w.r.t. the individual variables.

(a) Lecture hall, 38 occupants.

Variable	μ^*	μ	σ
v_{ab}	3.98	−3.84	2.03
a_{ab}	1.64	0.86	1.81
v_{wl}	5.14	−5.10	2.26
a_{wl}	1.75	0.71	2.02
$\alpha_{\mathrm{all}}^{\mathrm{ex}}$	5.24	−5.12	3.43
$\alpha_{\mathrm{all}}^{\mathrm{occ}}$	3.07	2.30	3.23
$\alpha_{\mathrm{wl}}^{\mathrm{ex}}$	1.62	−0.25	2.04
$\alpha_{\mathrm{wl}}^{\mathrm{occ}}$	1.54	0.17	1.91

(b) Lecture hall, 68 occupants.

Variable	μ^*	μ	σ
v_{ab}	7.34	−7.34	2.26
a_{ab}	2.12	1.55	2.18
v_{wl}	8.18	−8.18	2.83
a_{wl}	2.35	1.21	2.51
$\alpha_{\mathrm{all}}^{\mathrm{ex}}$	1.95	−0.35	2.49
$\alpha_{\mathrm{all}}^{\mathrm{occ}}$	1.84	−0.10	2.18
$\alpha_{\mathrm{wl}}^{\mathrm{ex}}$	2.36	−1.32	2.40
$\alpha_{\mathrm{wl}}^{\mathrm{occ}}$	1.92	−0.34	2.43

(c) Train, 34 occupants.

Variable	μ^*	μ	σ
v_{ab}	3.26	−2.90	2.24
a_{ab}	2.11	1.29	2.24
v_{wl}	4.31	−4.20	2.49
a_{wl}	2.02	1.09	2.38
$\alpha_{\mathrm{all}}^{\mathrm{ex}}$	11.53	−11.50	7.27
$\alpha_{\mathrm{all}}^{\mathrm{occ}}$	6.79	5.89	6.58
$\alpha_{\mathrm{wl}}^{\mathrm{ex}}$	1.61	0.19	2.01
$\alpha_{\mathrm{wl}}^{\mathrm{occ}}$	1.74	0.02	2.26

(d) Train, 60 occupants.

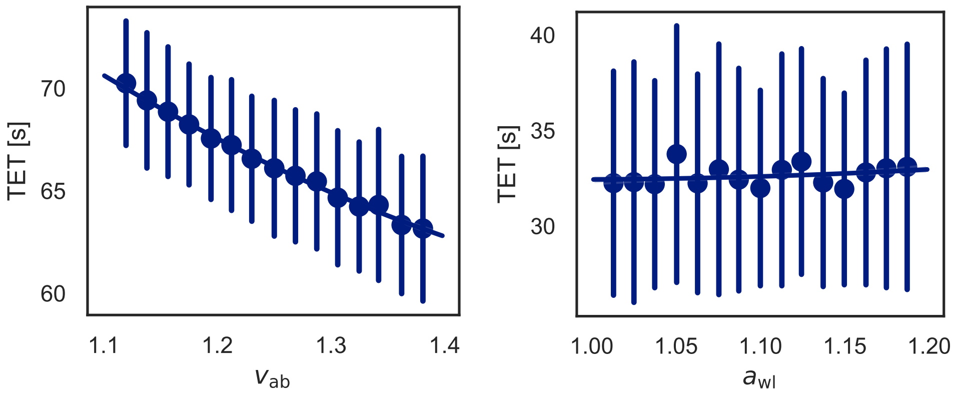
Variable	μ^*	μ	σ
v_{ab}	4.81	−4.71	2.72
a_{ab}	2.67	2.08	2.67
v_{wl}	6.64	−6.54	3.52
a_{wl}	2.96	2.21	2.99
$\alpha_{\mathrm{all}}^{\mathrm{ex}}$	5.18	−4.76	3.61
$\alpha_{\mathrm{all}}^{\mathrm{occ}}$	2.74	1.23	3.19
$\alpha_{\mathrm{wl}}^{\mathrm{ex}}$	3.03	−0.85	3.75
$\alpha_{\mathrm{wl}}^{\mathrm{occ}}$	2.48	−0.13	3.20

the computation. Given an input matrix $\mathbf{C}_N$ with $N = 2^n$ samples generated via Sobol' sequences, the matrix $\mathbf{C}_{2N}$ will be comprised of two submatrices with N rows, in which the first matrix is the $\mathbf{C}_N$ matrix from above, provided that they were both generated with the same random seed. This means that if the confidence intervals of the resulting SSI estimates are too wide, more samples can be generated to improve the accuracy without having to discard the previously computed ones.

The number of necessary model evaluations to estimate the first-order, second-order, and total effect indices is $N(2k+2)$, where N is the number of samples and k is the number of variables (9 in our case). Additionally, we need to estimate $\mathbb{E}[TET_{\boldsymbol{X}}]$ for each configuration of $\boldsymbol{X}$. In this case, I estimated this quantity with 9 model runs – 3 different seating configurations, each evaluated 3 times with different random seeds and priorities. Adjusted for simulation time and the four different scenarios, with 128 samples the estimated computation time is $T \approx 128 \times 20 \times 9 \times 4 \times 5\,\mathrm{s} = 128\,\mathrm{h}$. The input ranges are identical to those in the previous section (Table 6.5).

The obtained S_i and T_i estimates are shown in Table 6.9 along with their 95 % confidence intervals obtained by bootstrapping [71]. Selected interaction terms are shown in Table 6.10. A good way to visualize the dependence of TET on the individual variables is via scatter plots. An example of an influential variable and a non-influential one is shown in Figure 6.4. Such scatter plots for each variable are provided in Appendix A. Unfortunately, the confidence intervals for S_i and S_{ij} terms of the less influential variables are often wider than the value of the index itself, and providing more samples would be too computationally demanding. The next section will address this problem using surrogate models. There are also some negative numbers which are artifacts of the numerical method used to compute the estimates. Nevertheless, we can make the following observations:

- The observations made prior to the analysis still hold.
- The effect of velocities is mostly additive as evidenced by the similar magnitude of its S_i and T_i coefficients.
- The effects of acceleration are non-negligible and in the case of train with 34 occupants, they overcome the velocity in importance.
- The effects of all the positional variables are highly non-additive, which can be seen from the large difference between their S_i and T_i estimates as well as from the interaction terms in Table 6.10, although these estimates are burdened by a high degree of uncertainty.



(a) Lecture hall, 68 occ.

(b) Train, 34 occ.

Figure 6.4: An example of an influential vs. a non-influential variable. For readability, the scatter plot is constructed by binning the x-axis and drawing the average of each bin along with the standard deviation of the points belonging to said bin.

By summing up the first-order sensitivity indices, we can define a “degree of additivity” of the model as $\delta = \sum_{i=1}^{k} S_i$ and obtain the following values:

Lecture hall, 38 occ.: $\delta = 0.97$ **Train, 34 occ.:** $\delta = 0.82$

Lecture hall, 68 occ.: $\delta = 1.06$ **Train, 60 occ.:** $\delta = 0.78$

These values are clearly nonsensical as the sum of sensitivity indices cannot be higher than 1, without even accounting for the higher-order terms. This is a numerical artifact caused by the uncertainty in the estimates, which further motivates us to construct a surrogate model in order to obtain more robust estimates.

Table 6.9: Estimated first-order sensitivity indices S_i and total effect indices T_i based on 128 samples. The significant values variables are highlighted.

(a) Lecture hall, 38 occupants.

Variable	S_i	T_i
v_{ab}	0.25 ± 0.11	0.28 ± 0.07
a_{ab}	0.12 ± 0.09	0.21 ± 0.05
v_{wl}	0.07 ± 0.06	0.09 ± 0.02
a_{wl}	0.00 ± 0.06	0.07 ± 0.02
$\alpha^{\mathrm{ex}}_{\mathrm{all}}$	0.25 ± 0.15	0.46 ± 0.12
$\alpha^{\mathrm{occ}}_{\mathrm{all}}$	0.00 ± 0.08	0.18 ± 0.05
$\alpha^{\mathrm{ex}}_{\mathrm{wl}}$	0.03 ± 0.06	0.08 ± 0.02
$\alpha^{\mathrm{occ}}_{\mathrm{wl}}$	0.02 ± 0.05	0.07 ± 0.02
r	0.19 ± 0.08	0.25 ± 0.06

(b) Lecture hall, 68 occupants.

Variable	S_i	T_i
v_{ab}	0.36 ± 0.14	0.41 ± 0.10
a_{ab}	0.25 ± 0.11	0.26 ± 0.07
v_{wl}	0.06 ± 0.06	0.08 ± 0.02
a_{wl}	0.01 ± 0.04	0.04 ± 0.01
$\alpha^{\mathrm{ex}}_{\mathrm{all}}$	0.02 ± 0.04	0.04 ± 0.01
$\alpha^{\mathrm{occ}}_{\mathrm{all}}$	0.00 ± 0.05	0.04 ± 0.01
$\alpha^{\mathrm{ex}}_{\mathrm{wl}}$	0.02 ± 0.06	0.09 ± 0.03
$\alpha^{\mathrm{occ}}_{\mathrm{wl}}$	0.02 ± 0.04	0.04 ± 0.01
r	0.28 ± 0.11	0.37 ± 0.09

(c) Train, 34 occupants.

Variable	S_i	T_i
v_{ab}	0.01 ± 0.07	0.10 ± 0.03
a_{ab}	0.04 ± 0.07	0.15 ± 0.04
v_{wl}	-0.01 ± 0.07	0.10 ± 0.04
a_{wl}	0.02 ± 0.05	0.08 ± 0.02
$\alpha^{\mathrm{ex}}_{\mathrm{all}}$	0.75 ± 0.17	0.89 ± 0.23
$\alpha^{\mathrm{occ}}_{\mathrm{all}}$	-0.05 ± 0.15	0.34 ± 0.10
$\alpha^{\mathrm{ex}}_{\mathrm{wl}}$	0.00 ± 0.05	0.06 ± 0.02
$\alpha^{\mathrm{occ}}_{\mathrm{wl}}$	0.04 ± 0.06	0.10 ± 0.03
r	0.01 ± 0.06	0.12 ± 0.04

(d) Train, 60 occupants.

Variable	S_i	T_i
v_{ab}	0.20 ± 0.11	0.25 ± 0.07
a_{ab}	0.11 ± 0.12	0.22 ± 0.05
v_{wl}	0.08 ± 0.10	0.16 ± 0.04
a_{wl}	0.00 ± 0.07	0.10 ± 0.03
$\alpha^{\mathrm{ex}}_{\mathrm{all}}$	0.21 ± 0.12	0.34 ± 0.08
$\alpha^{\mathrm{occ}}_{\mathrm{all}}$	-0.03 ± 0.09	0.15 ± 0.05
$\alpha^{\mathrm{ex}}_{\mathrm{wl}}$	0.00 ± 0.07	0.16 ± 0.04
$\alpha^{\mathrm{occ}}_{\mathrm{wl}}$	0.03 ± 0.05	0.10 ± 0.03
r	0.15 ± 0.11	0.26 ± 0.07

Table 6.10: Selected second-order sensitivity indices S_{ij}.

Scenario	Variable	Variable	S_{ij}
hall_38	$\alpha_{\text{all}}^{\text{ex}}$	$\alpha_{\text{all}}^{\text{occ}}$	0.13 ± 0.25
hall_38	$\alpha_{\text{all}}^{\text{ex}}$	$\alpha_{\text{wl}}^{\text{ex}}$	0.11 ± 0.21
hall_38	$\alpha_{\text{all}}^{\text{ex}}$	$\alpha_{\text{all}}^{\text{occ}}$	0.15 ± 0.23
hall_38	$\alpha_{\text{all}}^{\text{ex}}$	r	0.11 ± 0.22
hall_38	$\alpha_{\text{all}}^{\text{occ}}$	$\alpha_{\text{wl}}^{\text{ex}}$	0.11 ± 0.13
hall_38	$\alpha_{\text{all}}^{\text{occ}}$	$\alpha_{\text{all}}^{\text{occ}}$	0.13 ± 0.12
train_34	$\alpha_{\text{all}}^{\text{ex}}$	$\alpha_{\text{all}}^{\text{occ}}$	0.30 ± 0.27
train_34	$\alpha_{\text{all}}^{\text{occ}}$	$\alpha_{\text{wl}}^{\text{ex}}$	0.20 ± 0.22
train_34	$\alpha_{\text{all}}^{\text{occ}}$	$\alpha_{\text{wl}}^{\text{occ}}$	0.19 ± 0.21
train_34	$\alpha_{\text{all}}^{\text{occ}}$	r	0.23 ± 0.21
train_60	v_{wl}	$\alpha_{\text{all}}^{\text{occ}}$	0.10 ± 0.14
train_60	$\alpha_{\text{all}}^{\text{ex}}$	$\alpha_{\text{all}}^{\text{occ}}$	0.19 ± 0.21
train_60	$\alpha_{\text{all}}^{\text{ex}}$	$\alpha_{\text{wl}}^{\text{ex}}$	0.13 ± 0.20
train_60	$\alpha_{\text{all}}^{\text{ex}}$	$\alpha_{\text{wl}}^{\text{occ}}$	0.14 ± 0.20
train_60	$\alpha_{\text{all}}^{\text{ex}}$	r	0.15 ± 0.19
train_60	$\alpha_{\text{all}}^{\text{occ}}$	$\alpha_{\text{wl}}^{\text{ex}}$	0.11 ± 0.12

6.6 Surrogate Models

We have seen in the previous section that despite the five-day computation time, the resulting SSI estimates still contain a high amount of uncertainty. Although the relative ranking of the variables based on their influence should reflect reality, it would be unwise to make definitive statements about the concrete percentages of variance attributed to the individual variables. In order to obtain estimates burdened by less uncertainty, we shall turn to surrogate modeling. Surrogate models are essentially meta-models – they build a model on top of another model using statistical or machine-learning techniques. Having accepted the fact that we will lose some of the information provided by the actual model, we can use the much-faster surrogate model to obtain an (almost) arbitrarily high amount of samples and get more robust sensitivity estimates.

During the course of the previous sections, we have accumulated a large amount of data, both from the elementary effects method as well as from the variance-based method. To get a qualitative idea about the dependence of the SSI estimates on the input data, all surrogate models will be trained on 3 distinct datasets:

EE – Data obtained from elementary effects analysis.

VB$_{\mathbf{128}}$ – Full dataset from variance-based analysis with 128 samples.

VB$_{16}$ – Reduced dataset with only the first 16 samples from VB$_{128}$.

Note that one "sample" from the VB method is, in fact, $2(k+1)\times 3\times 3 = 180$ samples, meaning that even the small dataset contains almost 3000 samples for each scenario. Since both VB datasets were constructed via Sobol' sequences, they should provide a reasonably representative sample of the parameter space. The EE dataset contains a total of 3840 points for every setting. Contrary to the VB datasets, the ranges of the individual variables are each discretized into 6 levels. It remains to be seen whether this discrete nature will significantly affect the performance of the models. Another advantage of the surrogate-modeling approach is that we do not need to average any of the values – the models should do that implicitly.

The first part of this section will construct a simple linear regression model and compute the standardized regression coefficients as described in Chapter 3.3.1. The second part will then employ more advanced machine-learning models and use them to compute the sensitivity indices, comparing them with those obtained in the previous section.

6.6.1 Linear Regression

The simplest surrogate-modeling approach involves fitting a simple Ordinary least squares model to the data and computing the standardized regression coefficients as $\beta_i = \frac{\sigma_{X_i}}{\sigma_Y} b_i$, where b_i is the standard least-squares regression coefficient. To keep this experiment simple, only the variables themselves will be considered, i.e., no polynomial features. The resulting β_i coefficients are shown in Table 6.11. In order to judge the reliability of these coefficients, the R^2 coefficients are also shown, both for the full datasets and for their reduced counterparts averaged over seeds and priorities (denoted R^2_{avg}). The most "important" variables are highlighted. We can observe the following behavior:

- Almost all the R^2_{avg} coefficients are higher than 0.8, meaning that the model is, in fact, linear to a fairly high extent.
- The performance on the EE dataset is more optimistic than on the more finely sampled VB datasets, meaning that the discretization process caused a non-negligible loss of information. The performance on both VB datasets is similar, with the exception of the train with 34 occupants, where the performance on the smaller dataset is still too optimistic.
- Despite this, there are very minor differences in the actual β values across the datasets.
- The relative ordering of the variables by their importance largely agrees with the S_i indices from the previous section, but there are some dif-

ferences, mainly that the linear model places high importance on the velocity of the WL group. It is unclear to me where this discrepancy originated as the individual factors have very little pairwise correlation.

To summarize, the use of standardized regression coefficients is an effective way to gauge the effects of individual factors on the output, provided that we know its limitations, mainly that it will fail to detect any interaction among the input variables.

Table 6.11: Standardized regression coefficients β_i for each input dataset. Coefficients of determination are also shown, where R^2 is computed on the whole dataset and R^2_{avg} is computed on the reduced dataset obtained by averaging the runs with identical parameters but different random seeds and priorities.

(a) Lecture hall, 38 occupants.

	EE	VB_{16}	VB_{128}
R^2	0.79	0.70	0.69
R^2_{avg}	0.91	0.86	0.85
v_{ab}	−0.38	−0.39	−0.39
a_{ab}	0.13	0.08	0.13
v_{wl}	−0.34	−0.33	−0.30
a_{wl}	0.05	0.09	0.03
$\alpha^{\text{ex}}_{\text{all}}$	−0.51	−0.55	−0.50
$\alpha^{\text{occ}}_{\text{all}}$	0.22	0.17	0.19
$\alpha^{\text{ex}}_{\text{wl}}$	−0.08	−0.11	−0.12
$\alpha^{\text{occ}}_{\text{wl}}$	0.02	0.03	−0.01
r	0.39	0.34	0.37

(b) Lecture hall, 68 occupants.

	EE	VB_{16}	VB_{128}
R^2	0.84	0.73	0.78
R^2_{avg}	0.94	0.92	0.93
v_{ab}	−0.56	−0.59	−0.56
a_{ab}	0.18	0.19	0.16
v_{wl}	−0.41	−0.46	−0.39
a_{wl}	0.03	0.10	0.04
$\alpha^{\text{ex}}_{\text{all}}$	−0.08	−0.08	−0.06
$\alpha^{\text{occ}}_{\text{all}}$	0.01	0.04	0.02
$\alpha^{\text{ex}}_{\text{wl}}$	−0.11	−0.17	−0.16
$\alpha^{\text{occ}}_{\text{wl}}$	−0.04	−0.03	−0.05
r	0.50	0.52	0.49

(c) Train, 34 occupants.

	EE	VB_{16}	VB_{128}
R^2	0.74	0.66	0.63
R^2_{avg}	0.88	0.81	0.77
v_{ab}	−0.10	−0.13	−0.14
a_{ab}	0.11	0.12	0.10
v_{wl}	−0.19	−0.15	−0.15
a_{wl}	0.05	0.07	0.03
$\alpha^{\text{ex}}_{\text{all}}$	−0.70	−0.72	−0.71
$\alpha^{\text{occ}}_{\text{all}}$	0.32	0.29	0.29
$\alpha^{\text{ex}}_{\text{wl}}$	−0.04	−0.02	−0.04
$\alpha^{\text{occ}}_{\text{wl}}$	0.00	0.03	0.00
r	0.23	0.12	0.15

(d) Train, 60 occupants.

	EE	VB_{16}	VB_{128}
R^2	0.77	0.62	0.62
R^2_{avg}	0.90	0.86	0.84
v_{ab}	−0.35	−0.34	−0.34
a_{ab}	0.28	0.27	0.26
v_{wl}	−0.34	−0.32	−0.31
a_{wl}	0.06	0.09	0.06
$\alpha^{\text{ex}}_{\text{all}}$	−0.39	−0.44	−0.41
$\alpha^{\text{occ}}_{\text{all}}$	0.14	0.17	0.16
$\alpha^{\text{ex}}_{\text{wl}}$	−0.16	−0.20	−0.20
$\alpha^{\text{occ}}_{\text{wl}}$	−0.02	0.00	−0.02
r	0.38	0.36	0.34

6.6.2 Machine-Learning Models

The final experimental section will train a set of more complex machine-learning models with the goal of predicting the total evacuation time as closely as possible given a combination of input factors. These surrogate models will then be used to estimate Sobol' sensitivity indices by generating a SALib input matrix with much more samples than before and using the surrogate model instead of the actual simulator to estimate the TET at each of these points, obtaining sensitivity indices with much less uncertainty. The datasets used to train the models will be identical to those in the previous section. I will use a standard machine learning pipeline which involves dividing the dataset into a train and test set. I will use the following two models:

- Ridge regression with polynomial features (Ridge)
- Random forest regressor (Forest)

Evaluation will now be done via mean absolute error on the test set, as the use of the R^2 coefficient becomes problematic with nonlinear models and a split dataset. The hyperparameter values will be chosen via grid search with cross-validation, separately for each scenario and dataset. The hyperparameter of the ridge regression is the regularization coefficient λ. The random forest has multiple hyperparameters including the number of trees, their maximum depth, the number of features per tree, and splitting criteria, all chosen via cross-validation.

The mean absolute errors for each scenario and dataset are shown in Table 6.12. The average test error for the lecture hall scenario is slightly over one second and for the train scenario, it is approximately 2 seconds, with the random forest generally reaching higher accuracy. Considering that the output is noisy due to the randomness of the model, and these errors are close to the lower bounds that have been identified in Chapter 6.2, these models can be considered a reasonable fit. The sensitivity analysis using surrogate models was conducted with 8192 samples, leading to very narrow confidence intervals. The resulting S_i and T_i are displayed in Table 6.13 with the most influential variables highlighted. We can make the following conclusions:

- Both models agree on which variables are the most influential and on their relative ranking, but disagree on their precise values, with the exception of lecture hall with 68 occupants, where the random forest is more biased towards v_{ab}.
- For the lecture hall scenario, there is little variation in the results with respect to the chosen dataset. For the train scenario, the EE dataset slightly differs in results, but not enough to influence the relative rankings of the variables.

- Ridge regression fails to detect any significant interaction even with polynomial features. The random forest succeeded in detecting some interactions as evidenced by the differences between S_i and T_i coefficients.
- The selected influential features are largely in agreement with those selected by the simple linear regression in the previous section.
- The velocity of the WL group was detected as important. In a way, the model can "construct" the total macroscopic velocity as a sum of v_{ab} and w_{wl} weighted by the ratio r.
- The S_i and T_i indices of the non-influential variables degraded to almost zero.

With the numerical analysis finished, we can summarize our findings.

Table 6.12: Mean absolute errors of the surrogate models in seconds. Tr. stands for the train set and Ts. for the test set.

	Ridge						Forest					
	EE		VB_{16}		VB_{128}		EE		VB_{16}		VB_{128}	
	Tr.	Ts.	Tr.	Ts.	Tr.	Ts.	Tr.	Ts.	Tr.	Ts.	Tr.	Ts.
hall_38	1.27	1.29	1.23	1.23	1.26	1.28	1.10	1.28	1.05	1.13	1.10	1.19
hall_68	1.44	1.49	1.41	1.39	1.41	1.41	1.29	1.48	1.27	1.37	1.32	1.39
train_34	2.12	2.27	2.20	2.35	2.29	2.24	1.79	2.20	1.80	2.06	1.93	2.01
train_60	1.96	1.98	1.89	1.91	1.95	1.96	1.75	2.01	1.68	1.91	1.77	1.88

6.7 Discussion and Closing Thoughts

Let us summarize our findings in this chapter:

1. The train geometry is inherently more complex and sensitive to initial positions than the lecture hall geometry.
2. With an increasing number of occupants, the problem becomes more linear, but the effect of the inherent randomness of the model also becomes more significant.
3. Overall, the main determining factors of the total evacuation time are the maximum velocities of the occupants and the number of movement-impaired individuals. In the lower-density scenarios, the overall average distance to the exit is also significant.
4. The individual methods agree about which factors are the most influential, but disagree on their exact amount of influence.

Table 6.13: First-order sensitivity indices and total effect indices obtained via surrogate models.

(a) Lecture hall, 38 occupants.

	Ridge						Forest					
	EE		VB_{16}		VB_{128}		EE		VB_{16}		VB_{128}	
	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i
v_{ab}	0.21	0.21	0.19	0.20	0.19	0.20	0.24	0.25	0.21	0.23	0.22	0.23
a_{ab}	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v_{wl}	0.26	0.26	0.26	0.26	0.24	0.25	0.15	0.17	0.08	0.10	0.10	0.12
a_{wl}	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00	-0.00	0.00
$\alpha_{\mathrm{all}}^{\mathrm{ex}}$	0.29	0.31	0.29	0.31	0.30	0.32	0.37	0.40	0.44	0.46	0.43	0.45
$\alpha_{\mathrm{all}}^{\mathrm{occ}}$	0.04	0.05	0.03	0.05	0.04	0.07	0.03	0.05	0.02	0.05	0.02	0.04
$\alpha_{\mathrm{wl}}^{\mathrm{ex}}$	0.00	0.01	0.01	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
$\alpha_{\mathrm{wl}}^{\mathrm{occ}}$	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00
r	0.14	0.14	0.14	0.15	0.14	0.15	0.13	0.15	0.15	0.18	0.16	0.17

(b) Lecture hall, 68 occupants.

	Ridge						Forest					
	EE		VB_{16}		VB_{128}		EE		VB_{16}		VB_{128}	
	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i
v_{ab}	0.36	0.36	0.33	0.33	0.36	0.36	0.47	0.49	0.52	0.54	0.47	0.48
a_{ab}	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.01	0.01
v_{wl}	0.36	0.37	0.38	0.38	0.36	0.37	0.19	0.22	0.13	0.15	0.19	0.21
a_{wl}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\alpha_{\mathrm{all}}^{\mathrm{ex}}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\alpha_{\mathrm{all}}^{\mathrm{occ}}$	-0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\alpha_{\mathrm{wl}}^{\mathrm{ex}}$	0.00	0.01	0.02	0.02	0.02	0.02	0.01	0.02	0.01	0.01	0.01	0.01
$\alpha_{\mathrm{wl}}^{\mathrm{occ}}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	0.00
r	0.23	0.24	0.22	0.22	0.21	0.22	0.25	0.28	0.28	0.30	0.27	0.29

5. Studying the effects of initial positions is, in general, problematic.

6. If the actual model is very computationally demanding, using a surrogate model is crucial, although caution has to be taken when selecting and training the model.

To address these points, the first observation can be seen from a) the overall higher variance with respect to random seeds and priorities, b) wider confidence intervals when estimating the sensitivity indices, and c) higher σ values

Table 6.13: First-order sensitivity indices and total effect indices obtained via surrogate models (cont.).

(c) Train, 34 occupants.

	Ridge						Forest					
	EE		VB$_{16}$		VB$_{128}$		EE		VB$_{16}$		VB$_{128}$	
	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i
v_{ab}	0.03	0.03	0.03	0.03	0.03	0.03	0.01	0.01	0.03	0.04	0.01	0.01
a_{ab}	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v_{wl}	0.10	0.11	0.07	0.07	0.07	0.08	0.03	0.04	0.02	0.02	0.01	0.02
a_{wl}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	0.00	0.00	0.00
$\alpha_{\text{all}}^{\text{ex}}$	0.64	0.69	0.63	0.74	0.63	0.73	0.70	0.78	0.77	0.84	0.75	0.84
$\alpha_{\text{all}}^{\text{occ}}$	0.09	0.15	0.09	0.20	0.10	0.20	0.12	0.20	0.06	0.13	0.10	0.19
$\alpha_{\text{wl}}^{\text{ex}}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\alpha_{\text{wl}}^{\text{occ}}$	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00	-0.00	0.00	0.00	0.00
r	0.04	0.05	0.02	0.02	0.03	0.03	0.02	0.03	0.02	0.03	0.01	0.02

(d) Train, 60 occupants.

	Ridge						Forest					
	EE		VB$_{16}$		VB$_{128}$		EE		VB$_{16}$		VB$_{128}$	
	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i	S_i	T_i
v_{ab}	0.20	0.21	0.16	0.16	0.17	0.17	0.23	0.27	0.23	0.26	0.19	0.21
a_{ab}	0.05	0.05	0.04	0.04	0.04	0.04	0.03	0.04	0.03	0.04	0.04	0.04
v_{wl}	0.34	0.35	0.29	0.29	0.30	0.31	0.21	0.26	0.05	0.06	0.14	0.16
a_{wl}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\alpha_{\text{all}}^{\text{ex}}$	0.16	0.17	0.22	0.23	0.22	0.23	0.25	0.28	0.44	0.48	0.35	0.38
$\alpha_{\text{all}}^{\text{occ}}$	0.02	0.02	0.05	0.06	0.03	0.04	0.02	0.03	0.01	0.03	0.01	0.02
$\alpha_{\text{wl}}^{\text{ex}}$	0.02	0.03	0.03	0.03	0.03	0.05	0.01	0.02	0.02	0.04	0.03	0.04
$\alpha_{\text{wl}}^{\text{occ}}$	0.00	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00
r	0.14	0.15	0.16	0.16	0.15	0.15	0.13	0.16	0.11	0.13	0.16	0.18

in the elementary effects method.

The second point can, likewise, be seen from the higher proportion of the variance being explained by a simple linear model and the smaller differences between the S_i and T_i coefficients. This is ultimately a good thing because the high-density situations are the most dangerous and therefore the most crucial to optimize. If the evacuation time is more governed by the macroscopic characteristics of the crowd than the microscopic subtleties of the geometry and the initial positions, the analysis becomes simpler and more effective.

The third point is in agreement with the referenced literature – the macroscopic characteristics are, indeed, the main determining factor of the overall performance of the evacuation. In the presence of a mixed crowd, the macroscopic characteristics of the individual groups and their relative numbers are important, but their initial positions are less so. This is unfortunate because, in reality, we rarely have control over the specific composition of the crowd and can influence only their initial positions. Nevertheless, placing the slower occupants close to the exit is generally favorable.

The fourth point can be considered a good thing. In the end, we do not care about the exact percentage of the variance attributed to each variable, so long as they do not differ by orders of magnitude. There are many levels of approximation in the whole process – the real-life scenario is approximated by a simulation, and the dependence on the parameters is further approximated by either averaging or building a separate model. Likewise, the input ranges of the variables also have an effect on the magnitude of the outputs. Although they were chosen based on previous research, they do not take into account factors such as cultural differences. We cannot hope to determine the precise influence each parameter has on the output. It is more important to be able to tell *which* parameters are the most influential and how they relate to each other.

The fifth point comes with a crucial remark – when talking about the sensitivity to the exit distance or inter-occupant distance, we did not actually test this quantity, but we tested the sensitivity to the α coefficients defined earlier. The conclusions are based on the assumption that these α coefficients model their respective variables well, which was shown to be true, at least on average. However, the space of possible seating configurations is rather high-dimensional, and describing them with only 4 numbers is bound to leave much of it unexplored, which opens up the path for further research.

And finally, the sixth point is in line with the best sensitivity analysis practices described in [72]. Given the large number of input variables and the inherent stochasticity of the model, it is infeasible to run the entire analysis using only the actual simulator evaluations. A more practical approach is to first perform a screening analysis and discard the clearly non-influential variables, then collect as much properly sampled data as is reasonable given the time constraints, and perform the actual analysis using carefully chosen surrogate models, comparing the results in the end.

Conclusion

This thesis concerned itself with determining the factors that have the largest influence on the total evacuation time in the presence of cramped interior settings. This included a) selecting the appropriate evacuation scenarios for analysis, b) constructing a programming framework that allows for repeatedly running Pathfinder simulations with appropriately set parameters and collecting the data, and c) analyzing the influence of the individual factors by means of sensitivity analysis.

The two different geometry settings selected for the analysis were that of a double-decker train and a lecture hall. These geometries have numerous contrasting properties in terms of visibility, the spacing of the seats, or the paths to the exit. These geometries were then modeled in the egress simulator Pathfinder as a courtesy of the researchers from the Faculty of Civil Engineering, CTU in Prague. Additionally, two different evacuation settings were chosen for each geometry – one where the facility is at approximately 50 % capacity, which allows for studying the effects of initial positions, and one where the facility is at 90 % capacity, which is of more practical interest.

The programming solution involved developing a Python package that builds an internal representation of the input file generated by Pathfinder. It then allows the user to specify the variables of interest and their distributions, including the number of occupants, the macroscopic characteristics of the crowd, or the way in which their initial positions are generated. The simulation is automatically run and the results are collected.

The analysis itself consisted of several stages. First, a preliminary analysis of the effects of the inherent randomness of the model (random seeds, initial directions, and occupant priorities) was conducted, which showed that in low densities, the stochasticity of the model is not very significant, attributing to roughly 20 % of the output variance, while in high-density situations, it can amount to as much as 50 % of the output variance in some cases.

Then, a probabilistic approach to generating the initial positions of the occupants was proposed. The sampling process is parametrized by two numbers

– the attraction to the exit, and the inter-occupant attraction, which were later used as a parameter for the sensitivity analysis

The sensitivity analysis itself employed two standard methods – the screening method of elementary effects which aims to identify the most influential and the least influential variables with a relatively small number of model runs, and the variance-based method based on estimating Sobol' sensitivity indices that measure the proportion of the output variance attributed to each input variable. Special emphasis was placed on the heterogeneity of the crowd, where a certain percentage of the occupants had a form of movement impairment simulated by lower maximum speed and slower acceleration.

Both methods identified the macroscopic characteristics of the crowd (maximum speed of both groups and the number of movement-impaired individuals) as the most important variables in high-density situations. However, when the facility is only at 50 % capacity, the importance shifts to the initial positions of the entire group. The initial positions of the movement-impaired individuals were not shown to be as important as their presence in itself.

Since the variance-based approach is highly computationally demanding and the results are burdened by a high degree of uncertainty, several surrogate models have been trained on smaller samples of the data obtained during the simulation process and used as a proxy for the actual simulation model. The results of the surrogate analysis largely agreed with the earlier results, although the numerical values that they produce have to be interpreted with caution. An interesting observation is that the surrogate models showed consistent results, even when trained on only a small fraction of the full dataset or on the highly discretized dataset generated by the elementary effects method.

There is room for follow-up research. The average distance of the individual occupants to the exit was shown to be highly influential, but there can still be subtleties that this macroscopic statistic will miss. For example, it was shown in Chapter 6.3.2 that there exist particular seating configurations where the positions of the movement-impaired individuals will cause a major slowdown of the crowd. Another interesting question would be to determine whether these observations still hold with the presence of, for example, a wider aisle or a more narrow exit door.

Bibliography

[1] Najmanová, H.; Kuklík, L.; et al. Evacuation trials from a double-deck electric train unit: Experimental data and sensitivity analysis. *Safety Science*, volume 146, 2022: p. 105523, ISSN 0925-7535, doi:https://doi.org/10.1016/j.ssci.2021.105523. Available from: https://www.sciencedirect.com/science/article/pii/S0925753521003660

[2] Kmec, J. *Estimation of Fundamental Diagram from Evacuation Experiments.* Bachelor's thesis, Czech Technical University in Prague, Faculty of Information Technology, 2021.

[3] Schadschneider, A.; Chowdhury, D.; et al. Chapter Eleven - Pedestrian Dynamics. In *Stochastic Transport in Complex Systems*, edited by A. Schadschneider; D. Chowdhury; K. Nishinari, Amsterdam: Elsevier, 2011, ISBN 978-0-444-52853-7, pp. 407–460, doi:https://doi.org/10.1016/B978-0-444-52853-7.00011-7. Available from: https://www.sciencedirect.com/science/article/pii/B9780444528537000117

[4] Steffen, B.; Seyfried, A. Methods for measuring pedestrian density, flow, speed and direction with minimal scatter. *Physica A: Statistical Mechanics and its Applications*, volume 389, no. 9, 2010: pp. 1902–1910, ISSN 0378-4371, doi:https://doi.org/10.1016/j.physa.2009.12.015. Available from: https://www.sciencedirect.com/science/article/pii/S0378437109010115

[5] Pathfinder Technical Reference Manual. Dec 2022. Available from: https://support.thunderheadeng.com/docs/pathfinder/2022-3/technical-reference-manual/

[6] Fruin, J. *Pedestrian Planning and Design.* Metropolitan Association of Urban Designers and Environmental Planners, 1971. Available from: https://books.google.sk/books?id=AydSAAAAMAAJ

[7] Feliciani, C.; Murakami, H.; et al. A universal function for capacity of bidirectional pedestrian streams: Filling the gaps in the literature. *PLOS ONE*, volume 13, 12 2018: p. e0208496, doi:10.1371/journal.pone.0208496.

[8] Maerivoet, S.; De Moor, B. Traffic Flow Theory. *Physics*, volume 1, 08 2005.

[9] Schadschneider, A.; Klingsch, W.; et al. *Evacuation Dynamics: Empirical Results, Modeling and Applications.* New York, NY: Springer New York, 2009, ISBN 978-0-387-30440-3, pp. 3142–3176, doi:10.1007/978-0-387-30440-3_187. Available from: `https://doi.org/10.1007/978-0-387-30440-3_187`

[10] Chattaraj, U.; Seyfried, A.; et al. Comparison of Pedestrian Fundamental Diagram Across Cultures. *Advances in Complex Systems (ACS)*, volume 12, 06 2009: pp. 393–405, doi:10.1142/S0219525909002209.

[11] Weidmann, U. *Transporttechnik der Fussgänger: transporttechnische Eigenschaften des Fussgängerverkehrs ; (Literaturauswertung).* Schriftenreihe des IVT, IVT, 1993. Available from: `https://books.google.sk/books?id=rX2NvgAACAAJ`

[12] Predtechenskii, V.; Milinskii, A. *Planning for Foot Traffic Flow in Buildings.* TT, Amerind, 1978, ISBN 9780862493264. Available from: `https://books.google.sk/books?id=3AZaPwAACAAJ`

[13] Schadschneider, A.; Klüpfel, H.; et al. *Fundamentals of Pedestrian and Evacuation Dynamics.* 06 2009, ISBN 978-1-60566-226-8, pp. 124–154, doi:10.4018/9781605662268.ch006.

[14] Hurley, M.; Gottuk, D.; et al. *SFPE Handbook of Fire Protection Engineering.* Springer New York, 2015, ISBN 9781493925650. Available from: `https://books.google.sk/books?id=xP2zCgAAQBAJ`

[15] Timmermans, H.; van der Hagen, X.; et al. Transportation systems, retail environments and pedestrian trip chaining behaviour: Modelling issues and applications. *Transportation Research Part B: Methodological*, volume 26, no. 1, 1992: pp. 45–59, ISSN 0191-2615, doi: https://doi.org/10.1016/0191-2615(92)90019-S. Available from: `https://www.sciencedirect.com/science/article/pii/019126159290019S`

[16] Kielar, P.; Borrmann, A. Modeling pedestrians' interest in locations: A concept to improve simulations of pedestrian destination choice. *Simulation Modelling Practice and Theory*, volume 61, 11 2015, doi:10.1016/j.simpat.2015.11.003.

[17] Hartmann, D. Adaptive pedestrian dynamics based on geodesics. *New Journal of Physics*, volume 12, no. 4, apr 2010: p. 043032, doi:10.1088/1367-2630/12/4/043032. Available from: https://dx.doi.org/10.1088/1367-2630/12/4/043032

[18] Gipps, P.; Marksjö, B. A micro-simulation model for pedestrian flows. *Mathematics and Computers in Simulation*, volume 27, no. 2, 1985: pp. 95–105, ISSN 0378-4754, doi:https://doi.org/10.1016/0378-4754(85)90027-8. Available from: https://www.sciencedirect.com/science/article/pii/0378475485900278

[19] Helbing, D.; Molnar, P. Social Force Model for Pedestrian Dynamics. *Physical Review E*, volume 51, 05 1998, doi:10.1103/PhysRevE.51.4282.

[20] Sugiyama, Y. Optimal velocity model for traffic flow. *Computer Physics Communications*, volume 121-122, 1999: pp. 399–401, ISSN 0010-4655, doi:https://doi.org/10.1016/S0010-4655(99)00366-5, proceedings of the Europhysics Conference on Computational Physics CCP 1998. Available from: https://www.sciencedirect.com/science/article/pii/S0010465599003665

[21] Seitz, M.; Köster, G. Natural discretization of pedestrian movement in continuous space. *Physical review. E, Statistical, nonlinear, and soft matter physics*, volume 86, 10 2012: p. 046108, doi:10.1103/PhysRevE.86.046108.

[22] Kleinmeier, B.; Zönnchen, B.; et al. Vadere: An open-source simulation framework to promote interdisciplinary understanding. 2019, 1907.09520.

[23] Yang, S.; Li, T.; et al. A review on crowd simulation and modeling. *Graphical Models*, volume 111, 2020: p. 101081, ISSN 1524-0703, doi:https://doi.org/10.1016/j.gmod.2020.101081. Available from: https://www.sciencedirect.com/science/article/pii/S1524070320300242

[24] Reynolds, C. W. Steering Behaviors For Autonomous Characters. In *Game Developers Conference 1999*, San Francisco, California: Miller Freeman Game Group, 1999, pp. 763–782.

[25] Ben Amor, H.; Obst, O.; et al. Fast, Neat and Under Control: Inverse Steering Behaviors for Physical Autonomous Agents. 06 2003.

[26] Saltelli, A.; Tarantola, S.; et al. Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models. 06 2004, doi:10.1002/0470870958.

[27] A, S.; M, R.; et al. *Global Sensitivity Analysis: The Primer*. Chichester (England): Wiley, 2008, ISBN 978-0-470-05997-5. Available from: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470059974.html

[28] Yondo, R.; Andrés, E.; et al. A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. *Progress in Aerospace Sciences*, volume 96, 2018: pp. 23–61, ISSN 0376-0421, doi:https://doi.org/10.1016/j.paerosci.2017.11.003. Available from: `https://www.sciencedirect.com/science/article/pii/S0376042117300611`

[29] Herman, J. Extensions of SALib for more complex sensitivity analyses. Nov 2014. Available from: `https://waterprogramming.wordpress.com/2014/02/11/extensions-of-salib-for-more-complex-sensitivity-analyses/`

[30] Sobol', I. M. Sensitivity estimates for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, volume 2, 1990: p. 112–118, in Russian, translated in Mathematical Modelling and Computational Experiments, 1, 407-414 (1993).

[31] Morris, M. D. Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics*, volume 33, no. 2, 1991: pp. 161–174, ISSN 00401706. Available from: `http://www.jstor.org/stable/1269043`

[32] Hornberger, G.; Spear, R. An Approach to the Preliminary Analysis of Environmental Systems. *J. Environ. Manage.; (United States)*, volume 12:1, 01 1981.

[33] *Kolmogorov–Smirnov Test*. New York, NY: Springer New York, 2008, ISBN 978-0-387-32833-1, pp. 283–287, doi:10.1007/978-0-387-32833-1_214. Available from: `https://doi.org/10.1007/978-0-387-32833-1_214`

[34] Turányi, T. Sensitivity analysis of complex kinetic systems. Tools and applications. *Journal of Mathematical Chemistry*, volume 5, 1990: pp. 203–248.

[35] Saltelli, A.; Aleksankina, K.; et al. Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices. *Environmental Modelling & Software*, volume 114, 2019: pp. 29–39, ISSN 1364-8152, doi:https://doi.org/10.1016/j.envsoft.2019.01.012. Available from: `https://www.sciencedirect.com/science/article/pii/S1364815218302822`

[36] Saltelli, A.; Annoni, P. How to avoid a perfunctory sensitivity analysis. *Environmental Modelling & Software*, volume 25, no. 12, 2010: pp. 1508–1517, ISSN 1364-8152, doi:https://doi.org/10.1016/j.envsoft.2010.04.012. Available from: `https://www.sciencedirect.com/science/article/pii/S1364815210001180`

[37] Di Lullo, G.; Gemechu, E.; et al. Extending sensitivity analysis using regression to effectively disseminate life cycle assessment results. *The International Journal of Life Cycle Assessment*, volume 25, no. 2, 2020: pp. 222–239, doi:10.1007/s11367-019-01674-y. Available from: `https://doi.org/10.1007/s11367-019-01674-y`

[38] Ferretti, F.; Saltelli, A.; et al. Trends in sensitivity analysis practice in the last decade. *Science of The Total Environment*, volume 568, 2016: pp. 666–670, ISSN 0048-9697, doi:https://doi.org/10.1016/j.scitotenv.2016.02.133. Available from: `https://www.sciencedirect.com/science/article/pii/S0048969716303448`

[39] Sobol', I. M. Sensitivity estimates for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, volume 2, 1990: p. 112–118, in Russian, translated in Mathematical Modelling and Computational Experiments, 1, 407-414 (1993).

[40] Saltelli, A. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, volume 145, no. 2, 2002: pp. 280–297, ISSN 0010-4655, doi:https://doi.org/10.1016/S0010-4655(02)00280-1. Available from: `https://www.sciencedirect.com/science/article/pii/S0010465502002801`

[41] Sobol', I. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, volume 7, no. 4, 1967: pp. 86–112, ISSN 0041-5553, doi: https://doi.org/10.1016/0041-5553(67)90144-9. Available from: `https://www.sciencedirect.com/science/article/pii/0041555367901449`

[42] Blitzstein, J.; Hwang, J. *Introduction to Probability, Second Edition*. Chapman and Hall/CRC, second edition, 2019, doi:10.1201/9780429428357.

[43] Campolongo, F.; Cariboni, J.; et al. An effective screening design for sensitivity analysis of large models. *Environmental Modelling & Software*, volume 22, no. 10, 2007: pp. 1509–1518, ISSN 1364-8152, doi:https://doi.org/10.1016/j.envsoft.2006.10.004, modelling, computer-assisted simulations, and mapping of dangerous phenomena for hazard assessment. Available from: `https://www.sciencedirect.com/science/article/pii/S1364815206002805`

[44] Sobol', I.; Kucherenko, S. Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation*, volume 79, no. 10, 2009: pp. 3009–3017, ISSN 0378-4754, doi:https://doi.org/10.1016/j.matcom.2009.01.023. Available from: `https://www.sciencedirect.com/science/article/pii/S0378475409000354`

[45] Wang, C.; Peng, M.; et al. Sensitivity analysis based on Morris method of passive system performance under ocean conditions. *Annals of Nuclear Energy*, volume 137, 2020: p. 107067, ISSN 0306-4549, doi: https://doi.org/10.1016/j.anucene.2019.107067. Available from: `https://www.sciencedirect.com/science/article/pii/S0306454919305699`

[46] Saltelli, A.; Annoni, P.; et al. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, volume 181, no. 2, 2010: pp. 259–270, ISSN 0010-4655, doi:https://doi.org/10.1016/j.cpc.2009.09.018. Available from: `https://www.sciencedirect.com/science/article/pii/S0010465509003087`

[47] Herman, J.; Usher, W. SALib: An open-source Python library for Sensitivity Analysis. *The Journal of Open Source Software*, volume 2, no. 9, jan 2017, doi:10.21105/joss.00097. Available from: `https://doi.org/10.21105/joss.00097`

[48] Iwanaga, T.; Usher, W.; et al. Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. *Socio-Environmental Systems Modelling*, volume 4, May 2022: p. 18155, doi: 10.18174/sesmo.18155. Available from: `https://sesmo.org/article/view/18155`

[49] Cukier, R. I.; Fortuin, C. M.; et al. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. *The Journal of Chemical Physics*, volume 59, no. 8, 09 2003: pp. 3873–3878, ISSN 0021-9606, doi:10.1063/1.1680571, `https://pubs.aip.org/aip/jcp/article-pdf/59/8/3873/11219577/3873_1_online.pdf`. Available from: `https://doi.org/10.1063/1.1680571`

[50] Huang, S.; Lu, S.; et al. Experimental study on occupant evacuation in narrow seat aisle. *Physica A: Statistical Mechanics and its Applications*, volume 502, 2018: pp. 506–517, ISSN 0378-4371, doi: https://doi.org/10.1016/j.physa.2018.02.032. Available from: `https://www.sciencedirect.com/science/article/pii/S0378437118301080`

[51] Oswald, M.; Kirchberger, H.; et al. Evacuation of a High Floor Metro Train in a Tunnel Situation: Experimental Findings. In *Pedestrian and Evacuation Dynamics 2008*, edited by W. W. F. Klingsch; C. Rogsch; A. Schadschneider; M. Schreckenberg, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN 978-3-642-04504-2, pp. 67–81.

[52] Oswald, M.; Lebeda, C.; et al. Full-Scale Evacuation Experiments in a smoke filled Rail Carriage — a detailed study of passenger behaviour under reduced visibility. In *Pedestrian and Evacuation Dynamics 2005*,

edited by N. Waldau; P. Gattermann; H. Knoflacher; M. Schreckenberg, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ISBN 978-3-540-47064-9, pp. 41–55.

[53] Alonso, V.; Abreu, O. V.; et al. A New Approach for Modelling Passenger Trains Evacuation Procedures. *Procedia - Social and Behavioral Sciences*, volume 160, 2014: pp. 284–293, ISSN 1877-0428, doi:https://doi.org/10.1016/j.sbspro.2014.12.140, xI Congreso de Ingenieria del Transporte (CIT 2014). Available from: `https://www.sciencedirect.com/science/article/pii/S1877042814062417`

[54] Administration, F. R. Passenger train emergency systems: Development of prototype railEXODUS software for U.S. passenger rail car egress. Technical report DOT/FRA/ORD-14/35, U.S. Department of Transportation, 2014. Available from: `https://railroads.dot.gov/elibrary/passenger-train-emergency-systems-development-prototype-railexodus-software-us-passenger`

[55] Hrabák, P.; Kmec, J.; et al. Challenges In Train Evacuation Modelling Using Pathfinder. In *Fire and Evacuation Modeling Technical Conference*, Brno, Czechia: Thunderhead Engineering, September 2022.

[56] Othman, W. N. A. W.; Tohir, M. Z. M. A study on evacuation time from lecture halls in Faculty of Engineering, Universiti Putra Malaysia. *IOP Conference Series: Earth and Environmental Science*, volume 140, no. 1, apr 2018: p. 012096, doi:10.1088/1755-1315/140/1/012096. Available from: `https://dx.doi.org/10.1088/1755-1315/140/1/012096`

[57] Zhao, Y.; Lu, T.; et al. The microscopic characteristics of escape behaviours from a three-dimensional lecture theatre under conditions of good and zero visibility. *Safety Science*, volume 118, 2019: pp. 641–653, ISSN 0925-7535, doi:https://doi.org/10.1016/j.ssci.2019.05.054. Available from: `https://www.sciencedirect.com/science/article/pii/S0925753517309906`

[58] Kimura, M.; Sime, J. Exit Choice Behaviour During The Evacuation Of Two Lecture Theatres. *Fire Safety Science*, volume 2, 1989: pp. 541–550, doi:10.3801/IAFSS.FSS.2-541.

[59] Imanishi, M.; Sano, T. Route Choice and Flow Rate in Theatre Evacuation Drills: Analysis of Walking Trajectory Data-Set. *Fire Technology*, volume 55, no. 2, 2019: pp. 569–593, ISSN 1572-8099, doi:10.1007/s10694-018-0783-2. Available from: `https://doi.org/10.1007/s10694-018-0783-2`

[60] Nilsson, D.; Johansson, A. Social influence during the initial phase of a fire evacuation—Analysis of evacuation experiments in a cinema theatre.

Fire Safety Journal, volume 44, no. 1, 2009: pp. 71–79, ISSN 0379-7112, doi:https://doi.org/10.1016/j.firesaf.2008.03.008. Available from: https://www.sciencedirect.com/science/article/pii/S0379711208000362

[61] Kosmopoulos, A.; Katsoulis, S. An Innovative Evacuation System for Multiplex Cinemas. In *Pedestrian and Evacuation Dynamics 2012*, edited by U. Weidmann; U. Kirsch; M. Schreckenberg, Cham: Springer International Publishing, 2014, ISBN 978-3-319-02447-9, pp. 135–141.

[62] Larsson, A.; Ranudd, E.; et al. The impact of crowd composition on egress performance. *Fire Safety Journal*, volume 120, 2021: p. 103040, ISSN 0379-7112, doi:https://doi.org/10.1016/j.firesaf.2020.103040, fire Safety Science: Proceedings of the 13th International Symposium. Available from: https://www.sciencedirect.com/science/article/pii/S0379711220300448

[63] Harms, V. L.; Poon, L. J. O.; et al. Take your seats: leftward asymmetry in classroom seating choice. *Frontiers in Human Neuroscience*, volume 9, 2015, ISSN 1662-5161, doi:10.3389/fnhum.2015.00457. Available from: https://www.frontiersin.org/articles/10.3389/fnhum.2015.00457

[64] Karev, G. B. Cinema Seating in Right, Mixed and Left Handers. *Cortex*, volume 36, no. 5, 2000: pp. 747–752, ISSN 0010-9452, doi:https://doi.org/10.1016/S0010-9452(08)70550-1. Available from: https://www.sciencedirect.com/science/article/pii/S0010945208705501

[65] Minegishi, Y. Experimental study of the walking behavior of crowds mixed with slow-speed pedestrians as an introductory study of elderly-mixed evacuation crowds. *Fire Safety Journal*, volume 120, 2021: p. 103041, ISSN 0379-7112, doi:https://doi.org/10.1016/j.firesaf.2020.103041, fire Safety Science: Proceedings of the 13th International Symposium. Available from: https://www.sciencedirect.com/science/article/pii/S0379711220300345

[66] Knoblauch, R. L.; Pietrucha, M. T.; et al. Field Studies of Pedestrian Walking Speed and Start-Up Time. *Transportation Research Record*, volume 1538, 1996: pp. 27 – 38.

[67] Gödel, M.; Fischer, R.; et al. Sensitivity Analysis for Microscopic Crowd Simulation. *Algorithms*, volume 13, no. 7, Jul 2020: p. 162, ISSN 1999-4893, doi:10.3390/a13070162. Available from: http://dx.doi.org/10.3390/a13070162

[68] De-peng, K.; Shou-xiang, L.; et al. Global Sensitivity Analysis of Occupant Egress Safety Model. *Procedia Engineering*, volume 11, 2011: pp. 179–184, ISSN 1877-7058, doi:https://doi.org/10.1016/j.proeng.2011.04.644, the 5th Conference on Performance-based

Fire and Fire Protection Engineering. Available from: `https://www.sciencedirect.com/science/article/pii/S1877705811008320`

[69] Tu, H.; Tamminga, G.; et al. Sensitivity analysis on heterogeneity of driving behavior for evacuation studies and its impacts on traffic safety. In *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 456–461, doi:10.1109/ITSC.2010.5625148.

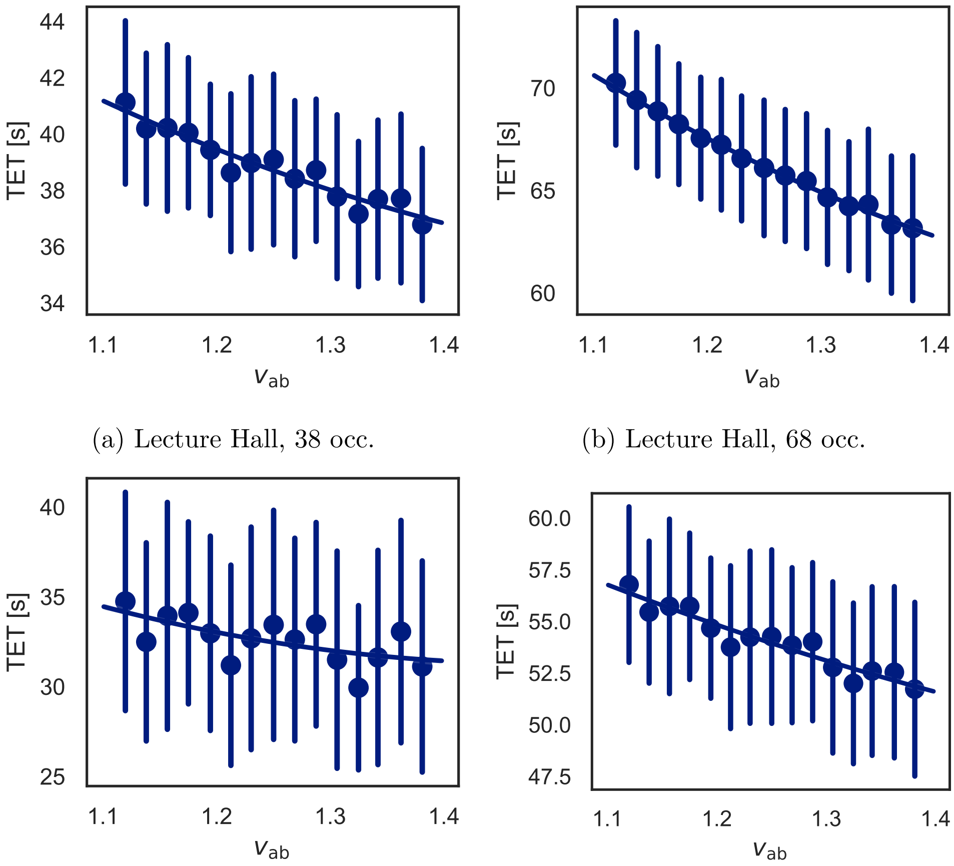
[70] Jullien, Q.; Dréan, V.; et al. Evacuation modelling – benchmark analysis of input parameter sensitivity of simulation software. In *Fire and Evacuation Modeling Technical Conference*, Brno, Czechia: Thunderhead Engineering, September 2022.

[71] Archer, G.; Saltelli, A.; et al. Sensitivity measures, ANOVA-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation - J STAT COMPUT SIM*, volume 58, 05 1997: pp. 99–120, doi:10.1080/00949659708811825.

[72] Campolongo, F.; Saltelli, A.; et al. From screening to quantitative sensitivity analysis. A unified approach. *Computer Physics Communications*, volume 182, no. 4, 2011: pp. 978–988, ISSN 0010-4655, doi:https://doi.org/10.1016/j.cpc.2010.12.039. Available from: `https://www.sciencedirect.com/science/article/pii/S0010465510005321`

Appendix **A**

TET Scatter Plots



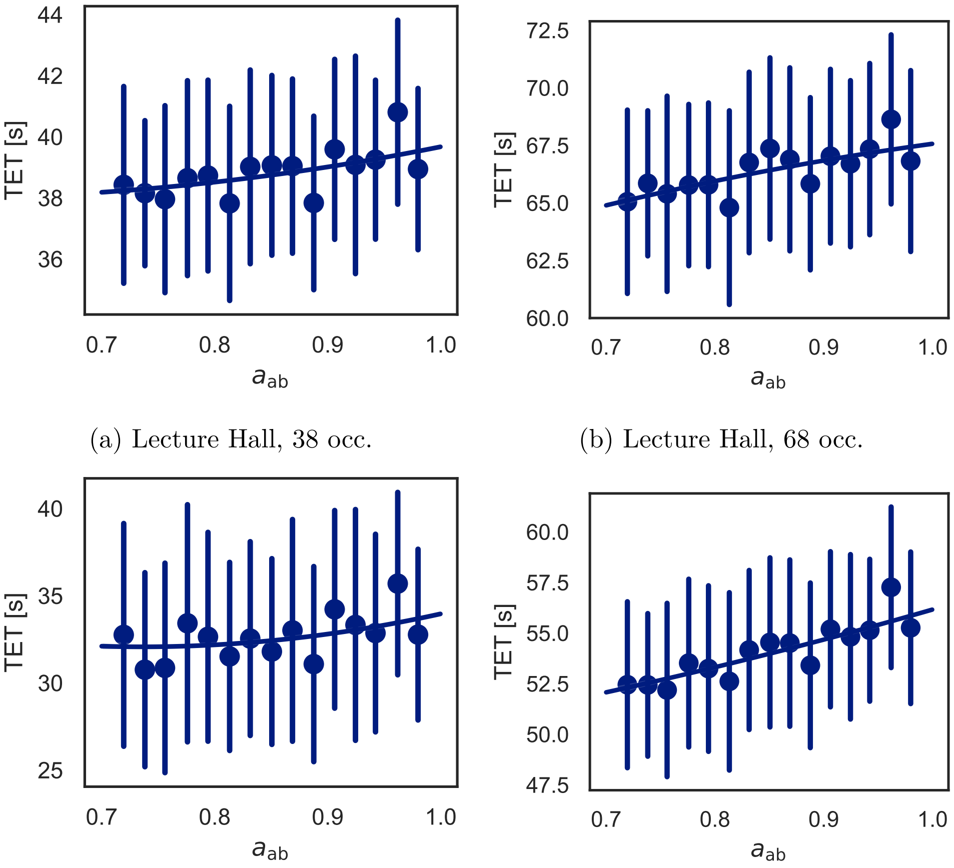
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.1: Velocity of AB group.



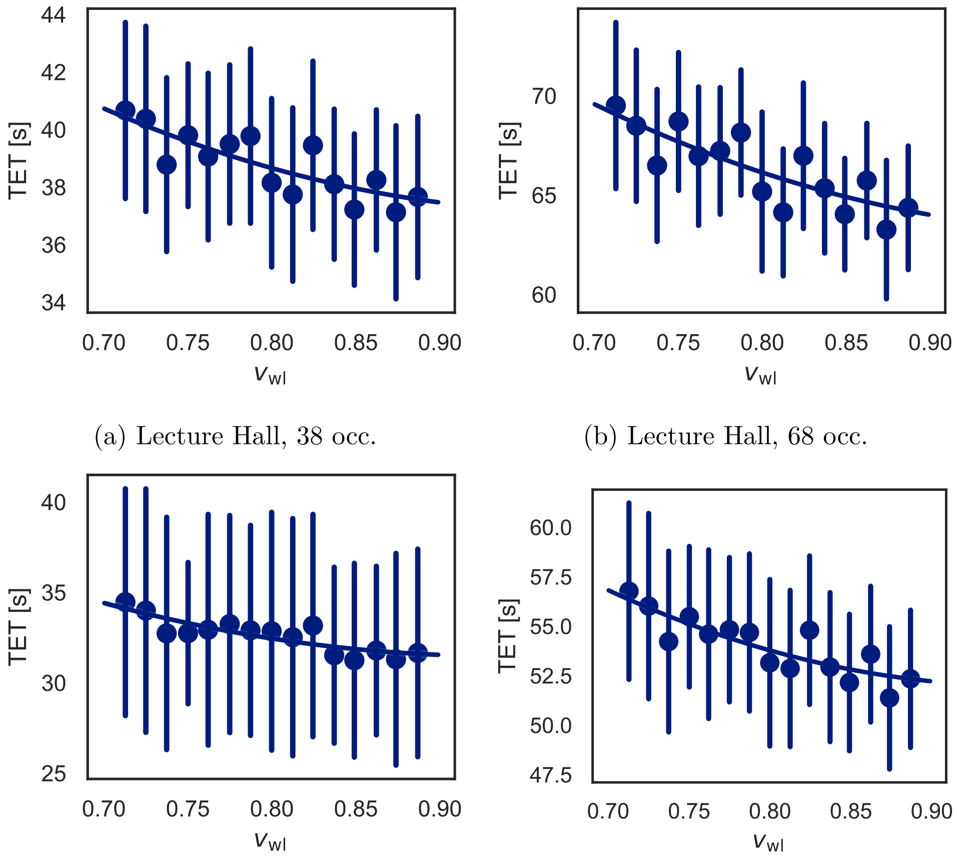
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.2: Acceleration of AB group.



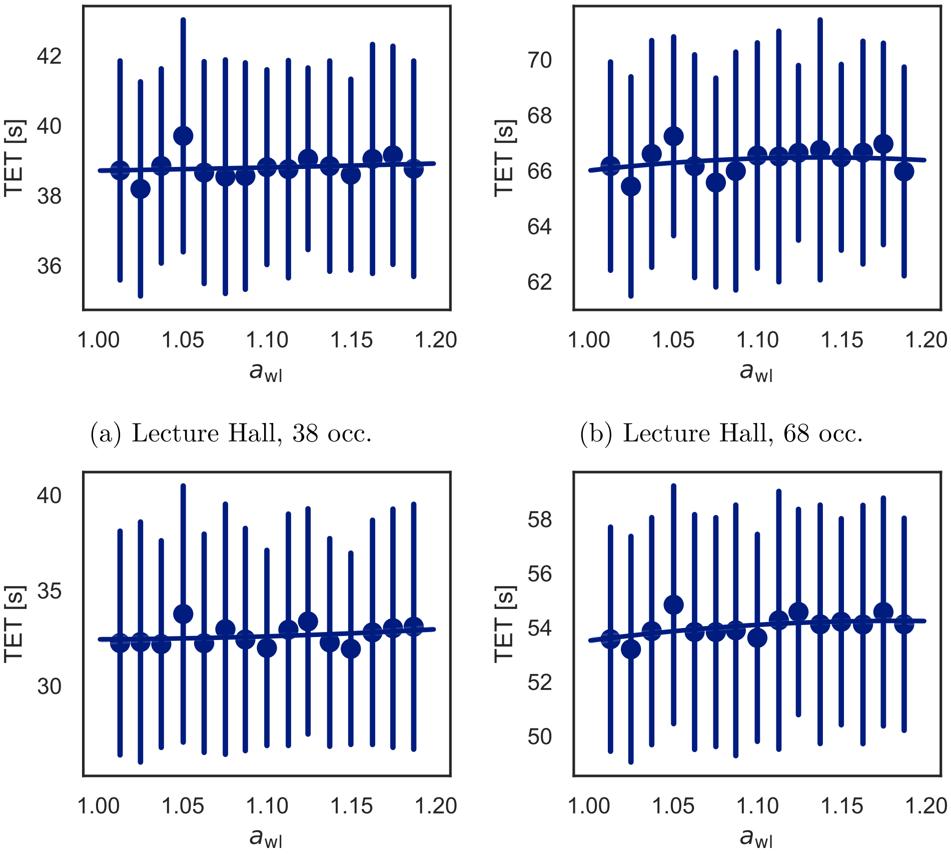
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.3: Velocity of WL group.



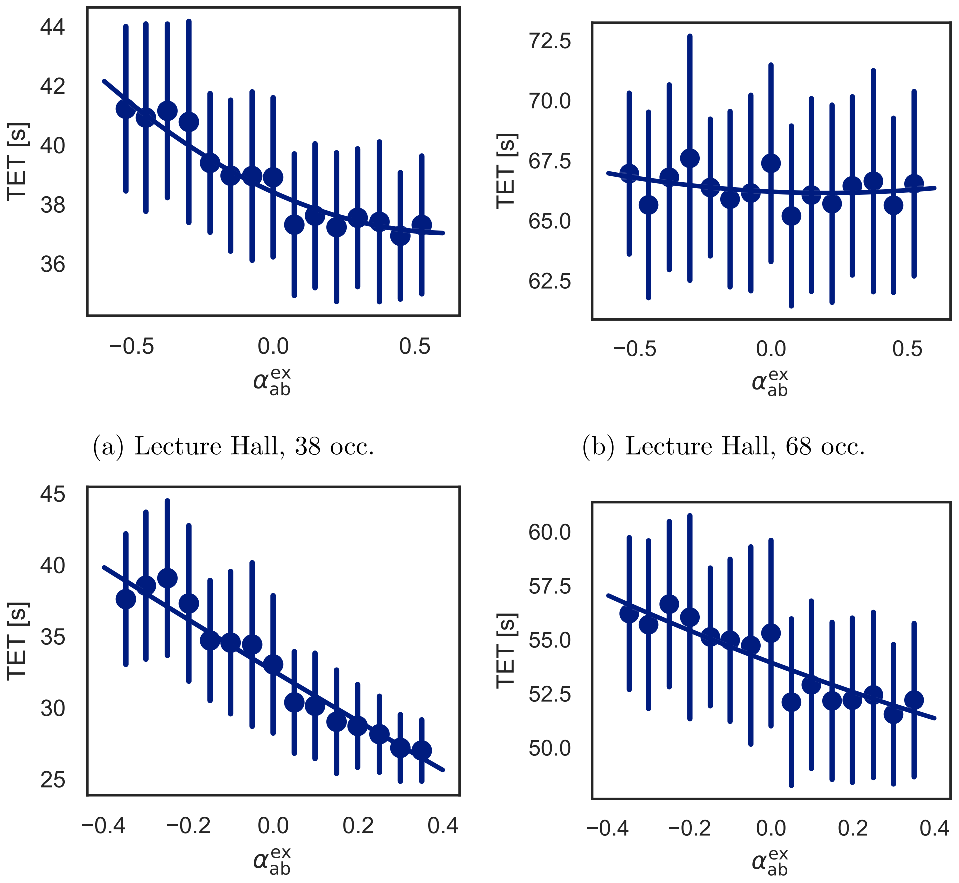
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.4: Acceleration of WL group.



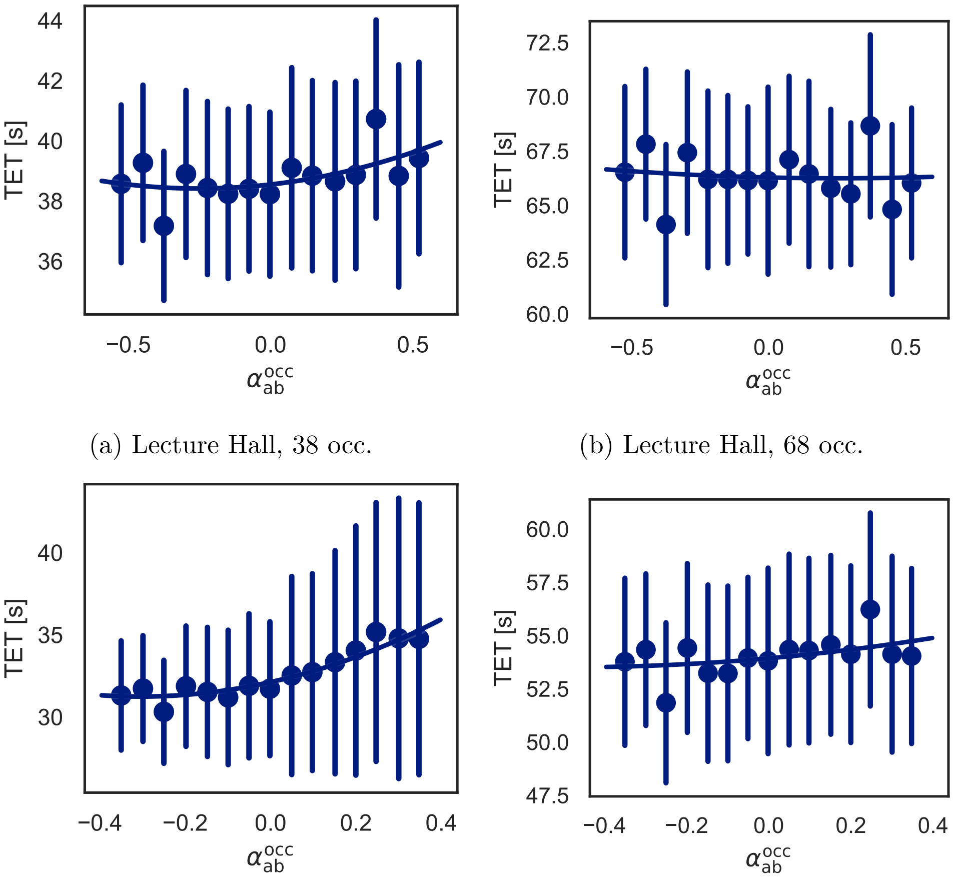
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.5: Exit distance of all occupants.



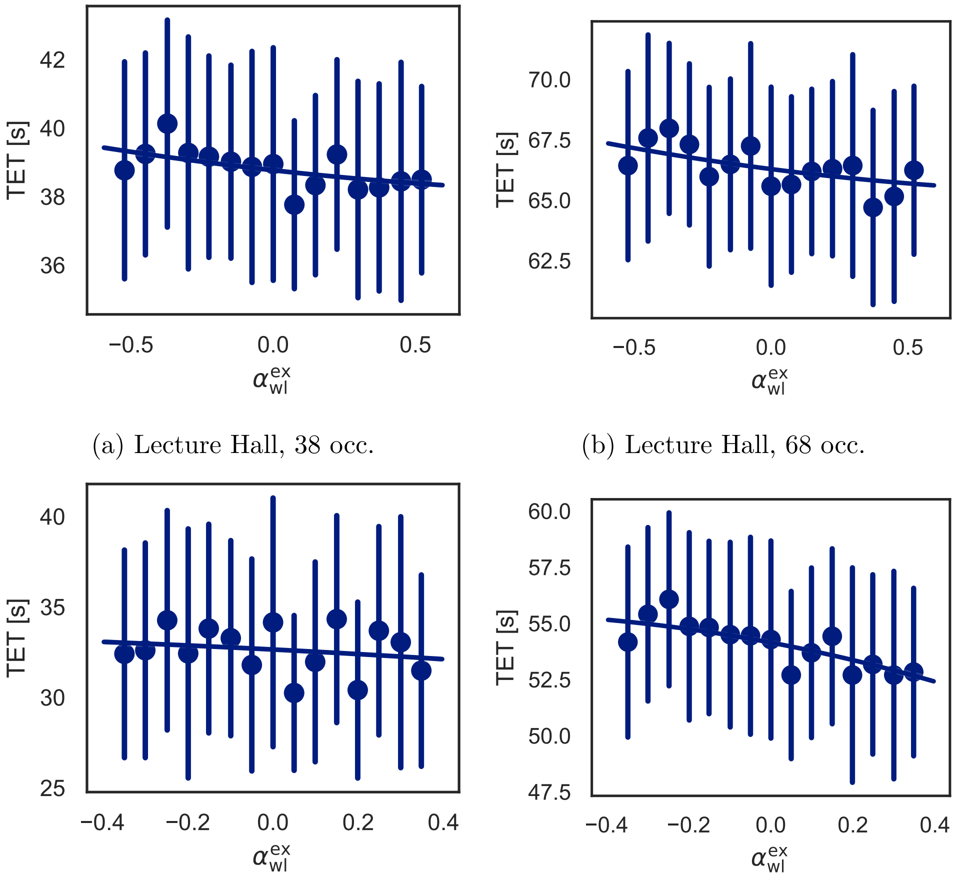
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.6: Inter-occupant distance of all occupants.



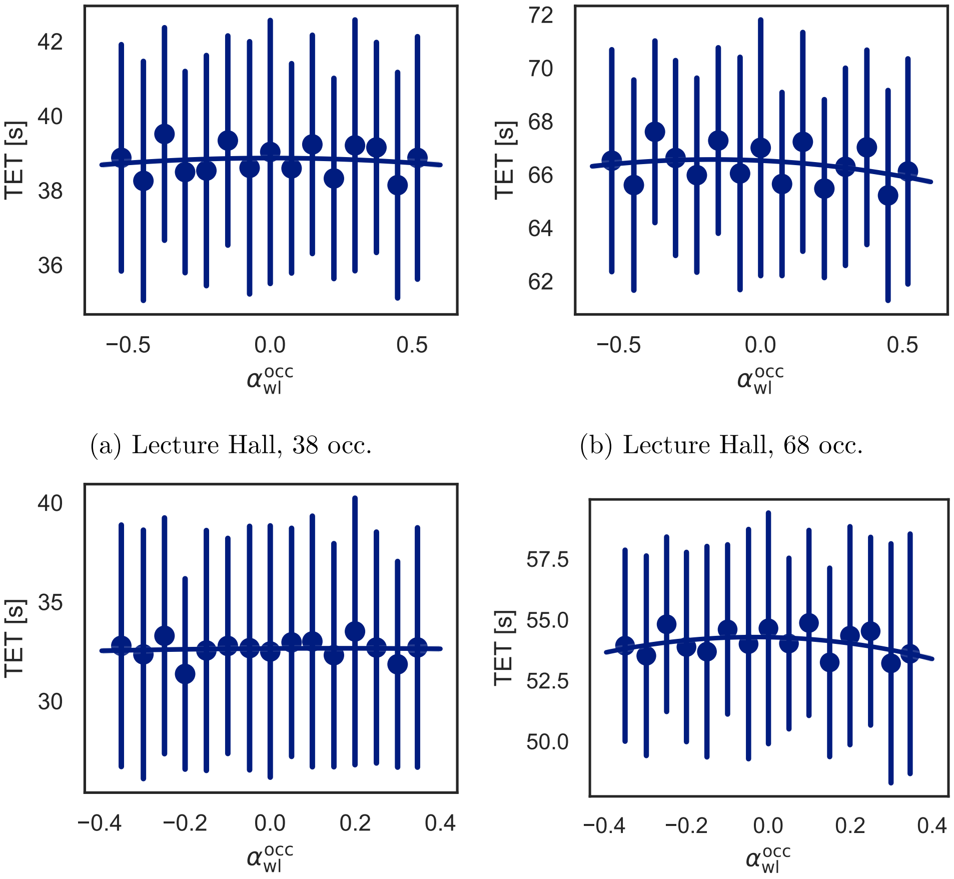
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.7: Exit distance of WL occupants.



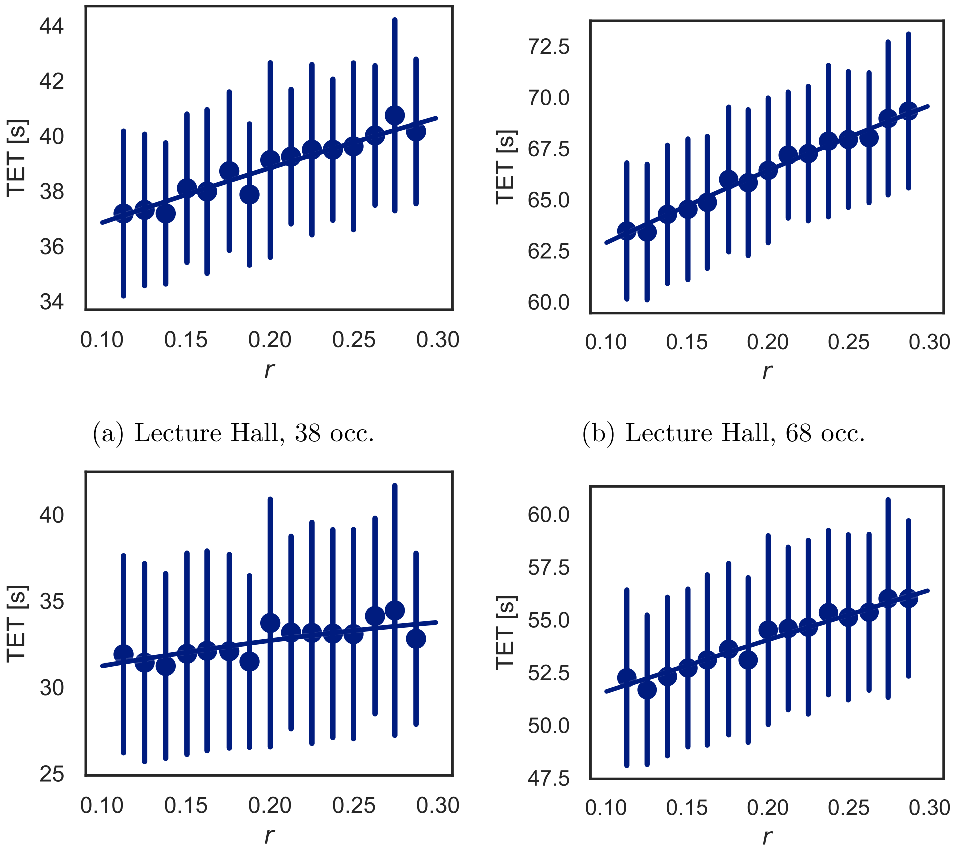
(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.8: Inter-occupant distance of WL occupants.



(a) Lecture Hall, 38 occ.

(b) Lecture Hall, 68 occ.

(c) Train, 34 occ.

(d) Train, 60 occ.

Figure A.9: Ratio of WL occupants.

APPENDIX **B**

Contents of CD

```
README.txt ........................ the file with CD contents description
src ................................. the source code files of the thesis
    simtools ................................... the simtools package
    Experiments ............................ the individual experiments
        Randomness ...................... the analysis of the randomness
        BruteForceSeating ..... the analysis of WL-group initial positions
        ElementaryEffects ................... the groupwise EE analysis
        ElementaryEffectsPartial ............. the pairwise EE analysis
        Global ............................. the variance-based analysis
        Surrogate ......................... the surrogate model analysis
    hall.txt ....................... Pathfinder input of the lecture hall
    train.txt ............................ Pathfinder input of the train
    hall_all_positions.csv ...... the initial positions of the lecture hall
    train_all_positions.csv .... the initial positions of the lecture train
    LaTeX .......................................... LaTeX source code
text ........................................ the thesis text directory
    thesis.pdf ............................. the thesis in PDF format
```