Calibration, validation & verification

Sparnaaij, Martijn; Duives, Dorine C.

DOI
10.1016/bs.atpp.2025.03.002

Publication date
2025

Document Version
Final published version

Published in
Walking and Pedestrians

Citation (APA)
Sparnaaij, M., & Duives, D. C. (2025). Calibration, validation & verification. In W. Daamen, & D. Duives (Eds.), *Walking and Pedestrians* (Vol. 15, pp. 297-348). (Advances in Transport Policy and Planning). https://doi.org/10.1016/bs.atpp.2025.03.002

CHAPTER NINE

Calibration, validation & verification

Martijn Sparnaaij* **and Dorine C. Duives**
Department of Transport & Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands
*Corresponding author. e-mail address: M.Sparnaaij@tudelft.nl

Contents

Abstract

This chapter explains the processes of verification, calibration, and validation in pedestrian modelling. These are essential processes in the design and use of pedestrian models that together ensure accurate simulations of pedestrian behavior. Verification confirms that the model's implementation aligns with its conceptual design, calibration adjusts model parameters to improve accuracy, and validation assesses how well the model represents real-world pedestrian movements.

Verification involves a structured process of testing whether the implemented model accurately reflects the conceptual model. This is done through a series of verification test cases, which compare the simulated outcomes to what is expected from the conceptual model.

Advances in Transport Policy and Planning, Volume 15
ISSN 2543-0009, https://doi.org/10.1016/bs.atpp.2025.03.002

Calibration and validation are interrelated but serve different purposes. Calibration is an iterative process that fine-tunes model parameters to minimize errors between simulation results and reference data. Validation, on the other hand, assesses how accurate a pedestrian model replicates pedestrian behavior and dynamics. The state-of-the-art approach involves multi-objective calibration and validation, where multiple scenarios and metrics (i.e. objectives) are used to calibrate and validate the model.

The choice of objectives has a major impact on the calibration and validation results. Key is that the scenarios and metrics are chosen such that they cover and capture all the relevant behaviors and dynamics. Which behaviors and dynamics are relevant depends on the intended use of the model and the type of modelled behavior.

As most pedestrian models are stochastic or use stochastic parameters it is essential that during calibration and validation replications, repeating the simulation multiple time using the same inputs, are run to deal with this. Lastly, a sensitivity analysis of the model is also important to determine which parameters the model is most sensitive to. This guides the calibration process and can ensure that the calibration is as efficient as possible.

All these processes are explained in detail in this chapter. This includes descriptions of how to apply them in the context of pedestrian behavior modelling and what are important factors to consider. This chapter therefore provides guidance for both model developers in creating valid models and model users is assessing the quality of their model for the intended application.

1. Introduction

Pedestrian models are used to model the movement of pedestrians and crowds in a wide variety of contexts. This includes simulating the pedestrian movement in a new to design train station, or the movement of pedestrians when evacuating a building for a safety analysis but also simulating the movement of pedestrians in a city center. For all these applications it is vital that the modelling results are accurate. Calibration, validation and verification of pedestrian models is essential to ensure that these pedestrian models provide accurate predictions. Together, the three processes check if a model has been correctly implemented, tune the model parameters and provide insight into how accurate a model is in different contexts.

The processes are defined as follows:

- ***Calibration***: the process whereby the parameters of the model are systematically adapted such that the model replicates reality more accurately
- ***Validation***: the process to determine to which degree the model accurately represents the real-world dynamics in light of the intended use of the model (Department of Defense, 1996)

- ***Verification***: the process of determining that a model implementation accurately represents the developer's conceptual description and specifications (Department of Defense, 1996)

Before we go into more detail about what these three processes exactly entail and how to apply them in the context of pedestrian models, we first discuss how these processes fit into the pedestrian modelling process. Fig. 1 presents the steps that are part of the modelling process. When you create a new model or add a new feature (i.e. when you are a model developer), you must go through all steps of the modelling process. When you are a model user you do not need to go through any of these steps. However, the calibration and validation are still highly relevant to model users. Primarily because these processes strongly determine if a model can be used in a particular context.

The first step is the modelling step where you create a conceptual model. The conceptual model is the collection of equations, figures and pseudocode that describe the abstraction of reality that is the model. Pseudocode describes an algorithm (the operations, the flow, the order etc.) in a structured way that is independent of a specific programming language. In the case of pedestrian models, this reality is the movement behavior of pedestrians. The accompanying process of modelling is the confirmation process where you check how well your conceptual model represent reality. These two processes fall outside of the scope of this chapter and are therefore not described in more detail.

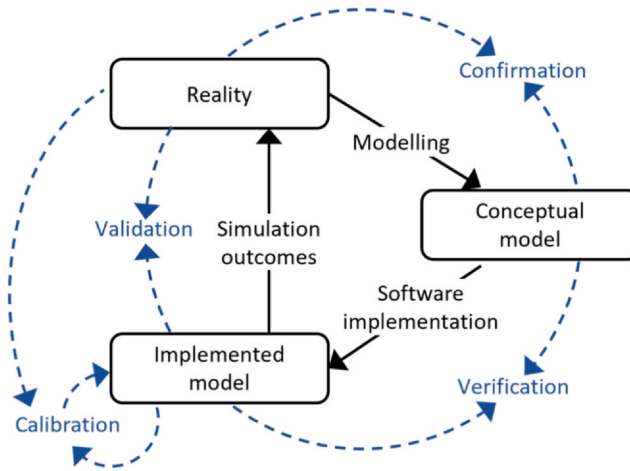

Fig. 1 An overview of the modelling process including the role calibration, validation and verification have in the modelling process. *Adaptation of Schlesinger (1979).*

The second step in the modelling process is the software implementation step. The conceptual model is translated into code so that a computer can perform the computations. The verification process checks whether the implementation is done correctly. That is, does the implemented model represent the conceptual model accurately. If not, the implementation must be revised. In Section 2 we describe the verification process in detail.

If the verification step is successful, the next step is to calibrate the model. This involves finding the values of the model parameters, that together result in the most accurate model representation of reality. In the case of pedestrian models this reality is the movement of the pedestrians. As the figure shows, this involves comparing the simulation outcomes to reality. Section 3 provides a detailed description of this process.

Lastly, the calibrated model must be validated. This last step is performed to check how accurate the calibrated model represents the pedestrian movement behavior in different contexts. If the results are unsatisfying, the first step is to update the calibration step, for example, by adding more data or changing the objectives, and run it again. If the subsequent validation step still has an unsatisfying result, the conceptual model must be updated to provide a more accurate representation of reality than the current conceptual model. Section 3 will provide a detailed description of this process.

When all steps have been successfully completed, the model can be used for simulating the pedestrian walking process within the contexts it has been calibrated and validated for. If you want to use the model outside of contexts for which it has been calibrated and validated the model must be re-validated and possibly re-calibrated for the intended new context. Therefore, the calibration and validation step are not only relevant for model developers but also for model users.

For example, a user has a pedestrian model at its disposal which has been calibrated and validated for use in a building evacuation context. The user now wants to use the model to simulate the day-to-day processes on a train station. The pedestrian behavior in these two contexts are not necessarily the same. For example, in evacuation the walking dynamics primarily involve unidirectional flows through corridors and doorways. In a train station bidirectional and crossing flows are far more common. Because of these differences, one must validate if the model, with its current parameter values, also provides an accurate representation of the walking processes in a train station. If this is not the case the model needs to be recalibrated to tune the parameters to this context.

It is important to note that a fully functional pedestrian model generally consist of multiple sub-models to simulate all the different choices, presented in Fig. 1 of Chapter 6, that determine the movement patterns of pedestrians. For each of the modeled behaviors, a specific sub-model is used, if this behavior is modelled in the first place and not provided as input. These sub-models have different modelling structures and use different modelling techniques. For example, for activity or route choice discrete choice models are commonly applied whilst for the route following and collision avoidance (operational choice behavior) approaches such as social forces are more common. The difference in the modelled behavior and the differences in modelling techniques mean that the calibration, validation and verification of these sub-models also differs.

In this chapter we focus primarily on the verification, calibration and validation of pedestrian walking models (operational choice behavior) because most modelling effort has focused on these models. Subsequently research on verification, calibration and validation has also mostly focused on these types of models. However, most techniques and methods explained in this chapter are also applicable to models at other levels. Furthermore, the basic principles of verification, calibration and validation are applicable to all types of models.The outline of this chapter is as follows: First, we discuss the process of model verification in more detail in Section 2. The processes of calibration and validation have a lot in common and are discussed together in Section 3. Pedestrian models are very often stochastic in nature so Section 4 discusses how to deal with this stochasticity in the context of calibration, validation and verification. Then we discuss the process of a sensitivity analysis and how it is relevant to calibration in Section 5.

2. Verification

Verification is the process of determining that a model implementation accurately represents the developer's conceptual description and specifications. It is performed by a model developer after a new model has been implemented or a model has been adapted. It involves a series of tests of different levels of complexity whereby if the model fails any of the tests the code needs to be revised to fix this.

The verification process is related to the validation process, and the two are sometimes confused. Both processes are designed to check the quality

of the model. However, they check different quality aspects of the model. Verification checks the quality of the implementation of the model. So, it asks the question, are you implementing the model correctly? It does this by comparing the model output to the output one would expect based on the conceptual model. Conversely, validation checks the quality of the model itself (and not the implementation). It asks the question, are you implementing the correct model? It does this by comparing the models' simulation results to the empirical behavior of pedestrians measured in a real-life context.

How you should verify a pedestrian model and how this includes both these types of tests is discussed in more detail below. Before doing so, we first describe the state-of-the-art regarding the verification of pedestrian models. This includes both the state-of-the-art guidelines featuring verification and the core lessons and techniques.

2.1 State-of-the-art of verification of pedestrian models

Currently, four guidelines exist that deal, in part, with the verification of pedestrian simulation models that model operational movement dynamics. These are the:

1. Revised guidelines on evacuation analysis for new and existing passenger ships (MSC.1/Circ.1533) from the International Maritime Organization (IMO) (International Maritime Organisation, 2016)
2. Guideline for Microscopic Evacuation Analysis (RiMEA) (RiMEA e.V., 2022)
3. NIST Technical Note 1822: The Process of Verification and Validation of Building Fire Evacuation Models (NIST) (Ronchi et al., 2013)
4. ISO 20414-2020: Verification and validation protocol for building fire evacuation models (ISO) (International Organization for Standardization, 2020)

In the remainder of the chapter, we refer to these guidelines by their abbreviation (defined in the brackets). These guidelines are all related to some degree and inspire or are inspired by each other. They are all focused on evacuation scenarios. The IMO guideline is specifically for passenger ships. The other three are specifically for buildings, and in the case of the ISO and NIST guidelines, specifically in the context of fire safety. Chapter 10 discusses these guidelines in more detail.

Table 1 on page 19 of (Wu, 2019) presents an overview of the different verification tests that exist in the four guidelines and which test is included

in which guideline. It encompasses 21 different verification test cases. Not all are strictly speaking verification test cases (according to the definition used in this chapter); some fall into the realm of validation. The ISO tests 11, 12, and 13, for example, are validation tests as these tests do not check the quality of the implementation but the quality of the model.

It is also important to note that these verification tests presume a certain conceptual model with certain features and focus on the most relevant parts for the given context (modeling walking behavior in an evacuation context using a microscopic model). So, some of the test cases included in these guidelines are not necessarily applicable to all pedestrian models. Nor is the list of test cases necessarily cover every basic aspect of pedestrian walking models that need to be verified in order to assess the quality of the implementation. (this is also acknowledged in all guidelines). Hence, these guidelines can provide inspiration and a good basis for verifying pedestrian models especially for the context of using microscopic pedestrian models in an evacuation context.

2.2 How to verify pedestrian models

The verification process includes several steps as Fig. 2 shows. The first step is to create the list of test cases which compare different aspects of the conceptual model to the implemented model. In the next subsection we discuss this in more detail.

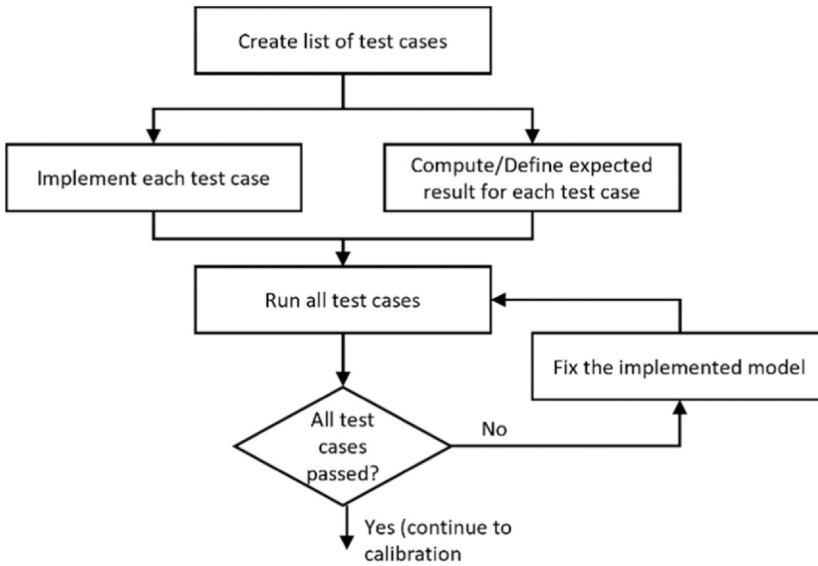

Fig. 2 The steps in the verification process.

Next, each of the test cases must be implemented so that the model can run one or more simulation steps of the specified scenario to compute the simulated outcome. Consecutively, for each of the test cases, the expected outcome must be defined or computed.

Then, all test cases can be run, and the simulated outcomes for each test case are compared to the expected outcome. If all the outcomes match, the model is verified and the next step, calibration, can be performed. If this is not the case this indicates that there are one or more implementation errors which need to be fixed. Once this has been done all test cases must be run again. This process repeats itself until the model is completely verified.

2.2.1 Create the list of verification test cases

The core of verifying a pedestrian model is taking the conceptual model and devise a list of test cases that provides a comprehensive overview of the quality of the implementation. How many and which test cases should be included in this list depends on the complexity of the model. A more complex model generally means more behaviors that are implemented and thus more code that needs to be tested. It also depends on the level of quality you want to achieve. The higher the level of quality the more test cases need to be included.

Because the list of test cases depends on the conceptual model, its complexity and the required level of quality, there is no comprehensive list of test cases that covers all possible pedestrian models. There are, however, some basic steps you can follow to compile the list of test cases and design the test cases themselves.

The first step is to identify the different components of the pedestrian behavior that are modelled. For example, on the operational level of pedestrian behavior of a microscopic model (the path following and collision avoidance behavior) this includes at least the following set of basic components:

- Walking unimpeded at a predefined speed (i.e. testing the most basic walking behavior)
- Following a path (i.e. testing the basic path following behavior)
- Interacting with obstacles
- Interacting with other pedestrians

Depending on the model, the operational pedestrian behavior can include more components like group behavior or the interaction with guiding information. For macroscopic and mesoscopic models, on the other hand, behaviors like the interaction with obstacles are not modelled and do not need to be included.

For each of the identified components, one or more test cases should be designed. Here, you need to take two things into account. First, the test case should be designed to only test the element of interest and reduce the impact of all other behavioral components to a minimum. For example, when testing the unimpeded walking component, no other pedestrians should be present in the simulation and the pedestrian should be far enough away from any obstacles to not be influenced by them. ISO tests 2 and 3 are examples of test cases that test this behavior for flat surfaces and stairs respectively.

Second, the test case should be simple enough that the expected outcome can be defined or computed. That is, a test case should only contain the minimal number of pedestrians, obstacles or other relevant model elements (e.g. guiding signs) and use the minimal number of time steps to test the behavioral component. This makes it feasible to compute the outcome by hand with a minimal risk of making errors. The expected outcomes of complex scenarios (e.g. tens or hundreds of pedestrians) are generally not easy to define or compute without the risk of making errors and are therefore mostly not suitable for verification purposes.

In the case of pedestrian models there is an exception to this rule of thumb of using simple cases. Namely, testing complex emergent phenomena such as lane formation. This can generally not be done well numerically because it is hard to quantify these phenomena well. However, using face verification (also called qualitative verification) you can still verify manually whether the model is able to produce certain emergent behaviors by visual inspection of the simulation results.

These steps should be repeated for all the different levels of behavior to ensure that all modelled behaviors are covered.

2.2.2 Example of designing a verification test case featuring obstacle interaction

To showcase how to create a test case we provide the example of creating a test case for a microscopic model testing the obstacle interaction. For this we designed the scenario depicted in Fig. 3. A pedestrian starts some distance from a wall and the floor field, that determines the preferred walking direction of the pedestrian, forces this pedestrian to walk towards the wall. Note that in an actual simulation one would not expect a floor field to point towards the wall because this is not realistic. However, verification tests do not have to represent a realistic situation because this is not what

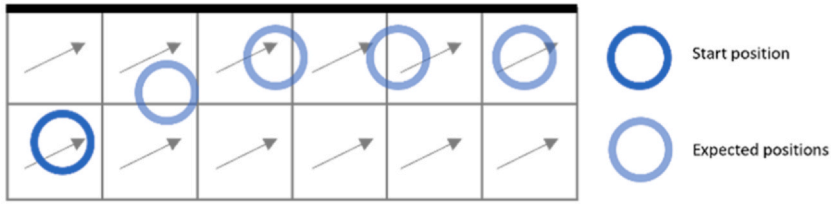

Fig. 3 Example of a verification case displaying the infrastructure and direction floor field.

they test. They test if the implementation is correct and often this is easier to test with artificial (and thus not necessarily realistic) situations.

The expected outcome is that the pedestrian walks towards the wall but is repulsed by the wall preventing the pedestrian from physically crossing the boundary of the wall. Numerically, the test should produce the trajectory of the simulated pedestrian and evaluate two things:

1. The pedestrian comes within a certain distance of the obstacle (i.e. this tests whether the pedestrian actually tries to walk through obstacles)
2. The pedestrian does not cross the wall (i.e. the pedestrian stays within a certain area)

2.3 Conclusion

By verifying the implemented model, a developer ensures that the implemented model behaves as expected according to the conceptual model. The verification is performed using many simple test cases that each test a single component of the modelled behavior. The use of many simple test cases, instead of a few complex cases, makes it much easier to design test cases and compute what outcome is expected based on the conceptual model. This makes the process less error prone and more robust. Furthermore, the simple test cases also ensure that in the case a test case fails, it is easy to identify which part of the code contains the error and needs to be fixed. When the implemented model is verified, it can be calibrated.

3. Calibration and validation

The next step in the modelling process is calibrating the model followed by validating the model. These two processes have different goals, but their processes are very similar. Therefore, we discuss these two processes together in this section. We start by defining each process and show

their similarities and differences (Section 3.1) and then describe the state-of-the-art (Section 3.2). Next, we introduce the concept of multi-objective calibration and validation (Section 3.3) and all the elements that make up this process. Lastly, we discuss each of these elements in more detail (Sections 3.4–3.6).

3.1 The process of calibration and validation

Calibration and validation are different processes with different objectives, but they share many elements. Fig. 4 shows these elements and their relation. The core technical difference between the two processes is that calibration is an iterative processes and validation is not. That is, in the calibration process the model parameters are systematically adapted to find the values that provide the best fit of the simulation output to the reference data. In the validation process, the quality of the model is assessed given a certain set of parameter values. That is, the fit of the model output to the reference data is tested.

Comparing the model's simulated pedestrian dynamics to the real-world pedestrian dynamics is the core of both processes. The real world is the thing you are modelling. For example, a corridor or street with pedestrians walking in both directions. This consists of the real system which in the case of pedestrian models is the pedestrian walking behavior. The real output of this system, which are the trajectories these pedestrians walk. And, the real input of the system. This is, among other things, the geometry of the infrastructure and the demand patterns. In the example of

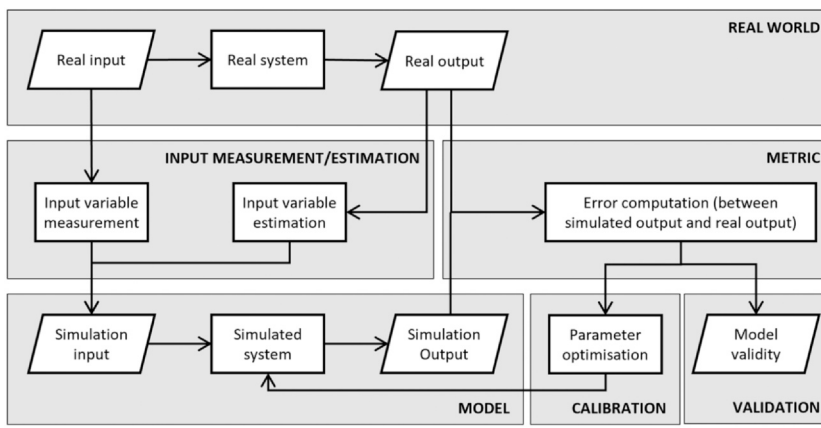

Fig. 4 The elements of the calibration and validation processes.

the corridor the inputs are the geometry of the corridor and the location and time where and when a pedestrian enters the corridor as well as relevant properties of the pedestrian, such as the pedestrian's preferred speed.

At the bottom of (Fig. 4) is the model. The model mirrors the real world whereby this model is always a mathematical simplified description of this real world that aims to capture the essential characteristics and behaviors of the real world (i.e. in this case the pedestrian movement behavior). It has a simulated system, which in this case is the pedestrian model modelling the real-world behavior of pedestrians. The model requires inputs such as the geometry, a demand pattern and pedestrian properties such as a distribution of the pedestrians' preferred speeds. The model also outputs data. What variables are included in this data depends on the type of model. For example, a microscopic model outputs simulated pedestrian trajectories whilst a macroscopic model outputs the velocity and density of each cell for each time step.

For calibration and validation, the real-world output and the model output need to be compared to assess how well the model represents the real world. This is done by computing the error between the real output and the simulated output using a certain metric. The metric is the combination of a variable of interest, for example the flow, and a difference measure, for example is the Root-Mean-Squared-Error (RMSE). The error is then a value that quantifies the difference between the flow in the real output and the flow in the simulated output.

The resulting error is used differently in the two processes. For the validation process, this is the end of the process. The computed error is a measure of the model's validity. For calibration, the error is input to the parameter optimization algorithm. This algorithm checks if the current parameter set is the optimal parameter set using the error. The optimal parameter set is the set of parameters that results in the best fit between the model and the real world. If the current parameter set is not optimal, a new simulation is run with a new parameter set. The output of this simulation is used to compute the error for this new parameter set and this is, again, fed back into the parameter optimization algorithm. This continues until the optimal parameter set is found.

It is important that the model input matches the real input closely to ensure that the model reproduces the real world well. Some inputs can be directly measured from the real input. These are, for example, the geometry and the location and time at which the pedestrians enter the

corridor. Other inputs cannot be directly measured from the real input but need to be estimated based on the real output. An example is the distribution of the preferred speeds of the pedestrians.

3.2 Recent calibration and validation approaches for pedestrian models

Pedestrian models have been calibrated and validated in many different ways. The core difference between these different ways is the use of different objectives. An objective is the combination of a scenario and a metric. Here the scenario is the description of the real-world case you are modelling.

The choice of objectives has a big impact on the calibration of a model. Different objectives result in different optimal parameter sets for the same model. This holds for both different scenarios and different metrics (Campanella et al., 2009b; Duives, 2016; Sparnaaij et al., 2019; Wolinski et al., 2014). More importantly, optimal parameter sets obtained using different objectives lead to different validation scores (Campanella et al., 2014). That is, the quality of the model depends on the objective used during calibration.

This means that a pedestrian model calibrated using one objective does not necessarily produce accurate results for other objectives. For example, a model calibrated using a unidirectional flow in a corridor does not necessarily produce accurate results for a bottleneck scenario. Nor does a model calibrated to reproduce accurate average flows through a bottleneck necessarily reproduces the spatial distribution of pedestrians in front of the bottleneck.

So, the quality of the model depends on the objective used during calibration. Similarly, in the case of validation, the validation score also depends on the objective used (Campanella et al., 2014). However, in contrast to calibration, the choice of objective does not determine the quality of the model but what we know of the quality of the model.

In both cases, using a single objective has a major limitation. Namely, we cannot assume that a model calibrated and validated using a single objective can reproduce other scenarios and metrics well. To deal with this limitation, you can use multiple objectives (i.e. multiple scenarios and metrics). Multi-objective calibration and validation are therefore the state-of-the-art procedures for calibrating and validating pedestrian models and should also be the default approach.

Another promising approach is the use of Bayesian inference methods (Bode, 2020; Gödel et al., 2022). These methods are particularly tailored to find distributions of parameters instead of a single value (also called point estimate). And in pedestrian models it is not uncommon to find parameters that are defined as distributions. For example, the desired speed in many pedestrian walking models is often defined as a distribution. However, because these methods have not yet been shown to work for multiple objectives more research is still necessary to ensure that they can be applied to calibrate pedestrian models using multiple objectives.

In the next parts we describe in detail how to perform multi-objective calibration and validation. Since single-objective calibration and validation are just the simplified versions of this multi-objective approach, the following description also covers single-objective calibration and validation.

3.3 Multi-objective calibration and validation

Fig. 5 presents the multi-objective calibration process. The core of the process is that it uses multiple scenarios and multiple metrics. The process starts with the search space. The search space (Θ) defines the set of parameter sets within which the calibration process searches for the optimal parameter set. In each step of the calibration process, the optimization algorithm selects one parameter set (Duives, 2016)(θ). The process then runs simulations for each of the n scenarios (s_i) using the model ($f(x, \theta)$) and the given input (x) and parameter set (θ). Each scenario needs to be replicated r times to deal with any stochasticity in the model, the inputs or the parameter set (see Section 4 for more info). The output (Y_{sim}) of all

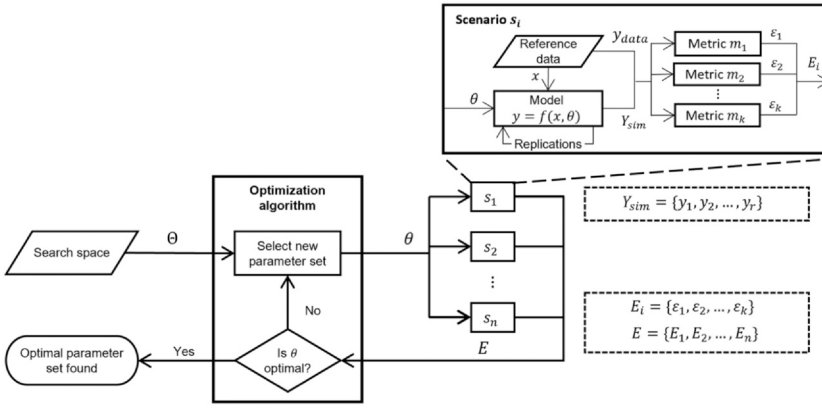

Fig. 5 The multi-objective calibration process.

replications is then compared to the reference data (y_{data}) of that scenario using k metrics. This results in a set of error values (E) which contains the errors for each combination of a scenario and a metric. This set is fed back into the optimization algorithm which determines if the current parameter set is the optimal parameter set. If this is the case the calibration process is finished. When this is not the case, the optimization algorithm selects another parameter set and repeats the processes until the optimal parameter set is found.

Multi-objective validation shares many elements with multi-objective calibration. Fig. 6 shows that multi-objective validation also uses multiple scenarios and metrics with the accompanying reference data. In contrast to the calibration process, the validation process starts with the optimal parameter set and, without iteration, results in a set of errors which represent the validity of the model.

In the reminder of this section we discuss the various elements of multi-objective calibration and validation in more detail. First, we discuss how to choose the objectives (Choosing calibration and validation objectives 3.4) which we follow up by describing how to deal with reference data and the input to the model (3.5). Both are relevant for calibration and validation. Lastly, we discuss the calibration specific elements related to the optimization of the parameter set (3.6).

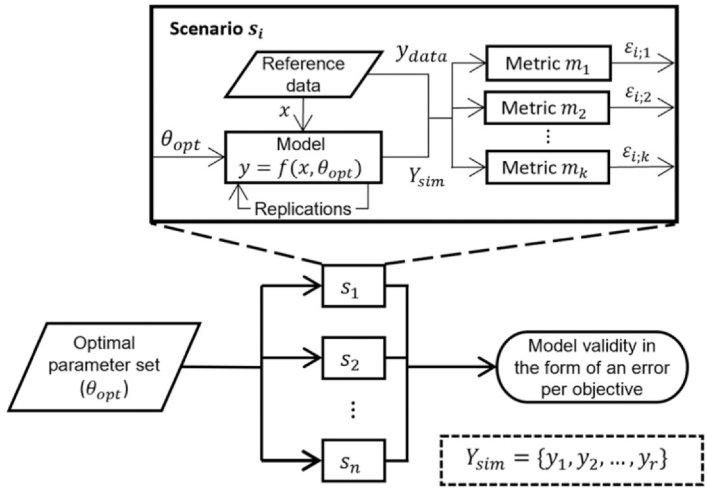

Fig. 6 The process of multi-objective validation.

3.4 Choosing calibration and validation objectives

The core of multi-calibration and validation is the choice of objectives. What objectives do you need to include and why? For which scenarios and for which metrics you want the model to produce accurate results depends on the intended use of the model. For example, the intended use of the model is to model building evacuation. In this case, you would want the model to produce accurate results for, among others, scenarios representing bottleneck dynamics and capacity, movement dynamics on staircases, and metrics such as the pre-evacuation time and the total effective evacuation time. Hence, this list of scenarios and metrics you want to accurately reproduce form the core of the objectives you need to include.

A second question is: What objectives can you include? This is determined by two factors, the data availability and the available computing resources and time. For every scenario and metric that you want to use during calibration or validation you need the relevant reference data. Therefore, any objective which is made up of a scenario or metric for which no data is available cannot be used.

The second factor to consider is the computation time required for performing the calibration. Each extra objective requires additional computation time. This is especially the case for scenarios as each additional scenario requires additional simulations to be run every iteration. Therefore, there needs to be a balance between the additional quality that is gained by adding an objective and the additional extra computation time it costs. That is, each added objective to the calibration should improve the accuracy of the model significantly in order to justify the additional required computational effort. For validation this is less relevant as you do not need multiple iterations.

Underneath, for both the scenarios and the metrics we describe the state-of-the-art knowledge on how the choice of these affect the calibration and validation results. We also describe how this knowledge can be used to inform the choice of scenarios and metrics for a given model and intended use. Note that compiling the list of objectives needed to calibrate and validate a model is not only relevant for model developers calibrating and validating a model. It is also a worthwhile exercise for model users that use a model for the first time for a certain application. By compiling the list of relevant objectives and comparing them to those which have been used to calibrate and validate the model, you gain insight into the validity of the model for you intended application. This also provides insight into whether

it is worthwhile or even necessary to validate (and potentially calibrate) the model for objectives it hasn't yet been validated for to ensure that the model captures behaviors critical to the intended application accurately.

3.4.1 Choosing scenarios

Ideally, the set of scenarios should cover the different types of situations and contexts you want the model to reproduce accurately. What these scenarios are for the given situations and contexts depends strongly on the type of model. For example, for a pedestrian model intended to be used for evaluating the design of a train station, the scenarios required to calibrate the route choice model are different than the scenarios required to calibrate the walking model. In this section we only focus on how to compile the list of scenarios for walking models because the calibration and validation of these models has been studied far more extensively. However, the presented strategies can also be applied to pedestrian models modelling the other levels of behavior.

In the case of walking models, the different situations distinguish themselves in the impact they have on the path following and collision avoidance behavior (the operational pedestrian walking behavior). So, differences in where you want to walk given the geometry of the infrastructure, where other people are walking, with what speed other people are walking in relation to you and in what direction people are walking in relation to you and your desired path. And differences in how you avoid potential collisions between a pedestrian and other neighboring pedestrians given variations in velocity (i.e., speed and walking direction) and type of walking surface.

Therefore, these situations should include the different types of flows in combination with different types of infrastructure, different levels of crowdedness and the different types of populations. For all these elements, that together make up a scenario, we know they impact the calibration and validation results. Either, from research on calibration and validation (Campanella et al., 2011; Duives, 2016; Sparnaaij et al., 2019) or from empirical research showing they impact the pedestrian movements and resulting emergent collective crowd behavior (Campanella et al., 2009a; Chattaraj et al., 2013; Hannun et al., 2022).

Each of the three elements is discussed in more detail below. Next, we provide guidance for choosing which scenarios to include in the calibration and validation.

3.4.1.1 Scenario characteristics 1: Infrastructure and flow types

The infrastructure and type of flow together determine what types of potential collisions pedestrians might experience. These can be, among others, head-to-tail collisions, head-on collisions, sideway collisions or a combination of them. The infrastructure and type of flow also determine the type of preferred path pedestrians want to follow. This can be, for example, straight or curved and crossing other streams or not. And they also determine the type of walking surface. Is it flat, inclined, is it stairs or an escalator.

We identify four atomic flow types. The unidirectional flow, the bidirectional flow, the crossing flow and the random flow. We also identify seven atomic infrastructures. The straight corridor, the corner, the intersection, the bottleneck, the open space, the stairs and the escalator. Combining these atomic flow types and infrastructures leads to a wide variety of situations with different underlying pedestrian walking behaviors. Below we dicuss a number of examples.

Unidirectional flow in a straight corridor describes a situation with a simple preferred path (straight-on) and mainly pedestrians following each other and potentially overtaking each other. So mainly, potential head-to-tail collisions which pedestrians need to avoid.

If the infrastructure now changes from a straight corridor to a corner the situation changes slightly. The path is now curved, and more variation is possible in the preferred path and thus the path following behavior. For example, do pedestrians cut the corner or do they follow a longer but smoother path? If the flow now becomes bidirectional instead of a unidirectional flow, there is the potential for head-on collisions which changes the collision avoidance behavior and thus describes again another situation.

Another example is a 4-way intersection. If this is combined with a unidirectional flow you get a situation where flows from 3 directions merge into one flow. This case includes the basic unidirectional behavior with following and overtaking and both straight and curved paths. But it also includes merging behavior.

If this intersection infrastructure is instead combined with bidirectional and crossing flows, you get a more complex situation. This includes behavior such as following, overtaking, crossing and merging. Potential head-on collisions, sideway collisions and head-to-tail collisions. And straight paths, curved paths, paths that cross other streams and those that do not.

Lastly, if we have a bottleneck with a unidirectional flow, pedestrians will show entering behavior (merging/queueing), exiting behavior (fanning out)

and the basic unidirectional behavior of following and overtaking. If this bottleneck is in the shape of an escalator, you get the walking/standing behavior on an escalator on top of this.

Next to the many combinations of infrastructure and flow type, each atomic infrastructure can have many variations. Straight corridors can have different widths, corners can have different angles, bottlenecks can have different lengths and widths, and different entrance and exit shape and intersections can have a different number of arms and different sizes of the crossing area.

Similarly, the atomic flow types can have variations. For each situation with multidirectional or merging flows, the ratio between the flows can vary. For example, for a bidirectional flow, the flows in both directions can be similar or the flow in one direction can be much higher than the flow in the other direction. Similarly for crossing flows or merging flows.

Lastly, pedestrians in a simulation can also be static. They are, for example, waiting at a location or performing an activity. So, there are also different possible ratios between static and moving pedestrians in a simulation.

3.4.1.2 Scenario characteristics 2: Level of crowdedness

The level of crowdedness describes how dense the pedestrian flows in the scenario are. This density strongly determines the number of potential conflicts a pedestrian needs to resolve. The more dense the flow, the more potential conflicts a pedestrian needs to resolve. Furthermore, the density also, in part, determines the speeds at which pedestrians move and the distance between pedestrians. Both of which affect the collision avoidance behavior.

We can identify three main levels of crowdedness. Free flow, capacity and congested. That is the left side of the fundamental diagram (free flow), the area around the peak (capacity) and the right side of the diagram (congested). See chapter 4 for more details. These levels have different properties regarding the number of potential conflicts a pedestrian will face, the speeds at which pedestrians walk and the distances between pedestrians.

At the free flow level of crowdedness pedestrians have few potential conflicts to solve, walk at higher speeds and the distance between pedestrians is generally big. So, pedestrians must avoid few collisions but need to consider potentially big differences in speed and generally will have plenty of space to change their path. This level corresponds to densities roughly in the range of 0–1 pedestrian per square meter.

At or around capacity, pedestrians must deal with more potential conflicts. Speeds are still relatively high but distances between pedestrians

are smaller. The densities at which the flow is at capacity vary wildly and is both infrastructure, flow type and population dependent. This can range from roughly 1.5 to 2.5 pedestrians per square meter.

At the congested flow level, pedestrians experience many potential conflicts and the distances between pedestrians are small. However, speeds are low due to the restricted movement. The densities at which flows are congested are generally higher than 3 pedestrians per square meter.

3.4.1.3 Scenario characteristics 3: Population

The population determines what the properties are of all pedestrians in the simulation and thus what the walking behavior is of all pedestrians in the simulation. What the properties are and how they are distributed is determined by the type of population and the level of heterogeneity within the population.

Different types of population can show different walking behaviors. These differences can be related to different aspects. For example, commuters at a train station will generally walk faster and more aggressively (e.g. accepting a smaller time to collision) than tourist strolling through a city. This is related to the difference in motivation. Or, for example, a population of young healthy people will generally show different walking behaviors than elderly. A difference is physical capability.

Within a population the walking behavior can also vary. Sometimes this variation is very limited, for example, in a population of students that participate in an experiment. In this case the population is almost homogeneous. Sometimes, the walking behavior within a population can be much more variable. For example, with a population of commuters there can be a large variation in the ages, physical capabilities and motivation. Some people must rush to catch their train whilst others have the time and take their time. In this case the population is far more heterogeneous.

3.4.1.4 How to choose the scenarios

There are a huge number of possible combinations of infrastructures, flow types, density levels and population compositions to form many scenarios. You can use a two-step approach to first compile a list of all relevant scenarios given the intended use of the model and then prune the list to make the calibration and validation effort feasible.

3.4.1.5 Compiling the list of relevant scenarios

In step one you take the intended use of the model, and you list for all three elements (combination infrastructure and flow type, density level and

population composition) the cases that are relevant. The unique combinations of these three lists form the list of relevant scenarios for both the calibration and validation step. We provide an example to show how the intended use of a model guides the choices for all three elements.

The example is of a model that is intended to simulate building evacuation. The main flow type in an evacuation scenario is unidirectional. The vast majority of people will be moving in the same direction towards the exits. In some cases, some people, like emergency personnel, might move in the opposite direction leading to bidirectional and crossing flows as well. We also expect most atomic infrastructure types to be present in an evacuation scenario except for open spaces and moving escalators. This leads to all unique valid combinations of the flow and infrastructure types shown below to be included. Invalid combinations are combination where the flow type cannot exist given the infrastructure. So, in this example the combination of the crossing flow with all infrastructure types except the crossing type are not valid. Hence, there are $2 * 5 + 1 = 9$ valid combinations in this example.

Flow types	**Infrastructure types**
Unidirectional	Straight corridor
Bidirectional	Corner
Crossing	Bottleneck
	Crossing
	Stairs

For the bidirectional and crossing flows the ratios between the flows are not expected to be 50/50 but closer to 95/5. The geometric design of the infrastructure types will also vary. Bottlenecks will have different widths, stairs will have different widths and lengths, crossings can have 3 or 4 arms and corners can have different angles. To capture this variation in geometric design it is advised to include multiple variations of each infrastructure type. How many depends on the infrastructure type and the variations you expect to occur given the intended use of the model.

In the case of evacuation most locations will have a flow that is close to capacity or beyond that. Hence, these two levels of crowdedness should be included. Lastly, the population in a building can vary significantly between different types of buildings. The more types of buildings the model is

intended to handle, the more different populations must be used during calibration and validation. If we assume the model is only intended for office building, we should ideally include a population that represents a heterogeneous population of working age.

So, we have $9 * 3 = 27$ different infrastructure and flow type scenarios (9 valid combinations of flow and infrastructure types and on average 3 variations per infrastructure). Then we have 2 different levels of crowdedness and 1 population. The unique combination of these three elements results in a list of 54 scenarios that ideally should be used in the calibration and validation of this type of model.

3.4.1.6 Pruning the list of relevant scenarios

The second step is to prune the list of all relevant scenarios down to a feasible set for the calibration and/or validation of the model. This is necessary to obtain a good balance between the added value of each scenario and the extra effort it takes in implementing the scenario and running all the simulations. For calibration, the added value of a scenarios is determined by how much it will improve the model's accuracy. For validation, the added value is determined by how much insight will it provide into the model's quality.

The first pruning step is to rank the scenarios. Which scenarios are most important to include (i.e. provide the most added value) and which scenarios provide less value when included? Ranking is performed using two criteria: (1) The richness of the data of the scenarios and (2) the importance of the scenario for the intended use.

Some scenarios are poorer in data than others (Campanella, 2016; Sparnaaij et al., 2019). That is, they lack certain situations that prevents them capturing certain behaviors well. These are generally the simpler scenarios. So, those with a simpler infrastructure and flow type (e.g. a unidirectional flow in a corridor), flows with a low level of crowdedness (free flow) and those with a homogeneous population. Therefore, the more complex a scenario (so the more interactions and the more types of interaction), the higher the scenario should be ranked.

Some scenarios represent cases for which it is more critical to get accurate results given the intended use of a model. These scenarios should be ranked higher than scenarios which represent less critical cases. For example, for a model intended for building evacuation studies it is, among other cases, critical to get accurate results for the flow through bottlenecks. Hence, the bottleneck scenario at the capacity crowdedness level is a

critically important scenario to include and should be ranked highly. On the other hand, the unidirectional flow in a corridor at free flow scenario is far less critical and thus can be ranked much lower.

The second pruning step is to remove all scenarios for which no reference data is available. For calibration these removed scenarios cannot be used. However, the list of these scenarios is still valuable. First, it provides insight into the cases where the model might not perform well when the optimal parameter set is used. Second, it identifies gaps in the data availability and thus for which scenarios it can be worthwhile to collect data. Thirdly, if new data sets become available, the list can be used to determine if a model should be recalibrated using more scenarios.

For validation the case is slightly different. The removed scenarios cannot be used for quantitative validation but can still be used for face-validation (qualitative). Other than that, the list of removed scenarios is valuable for the same reasons as for calibration.

The last pruning step is to select the scenarios to be included in the calibration and validation based in the remaining and ranked list of scenarios. This selection is done based on two criteria. (1) The balance between the number of scenarios to include and the (computational) effort necessary to complete the calibration and validation and (2) the balance between the different scenarios.

The effort that is necessary to complete the calibration and validation of the model is proportional to the number of scenarios. The more scenarios, the more effort it takes. This effort has two elements, the effort necessary to prepare all scenarios and the computational effort necessary to run all scenarios. Especially for calibration the computational effort is high because each scenario generally needs to be run many (hundreds of) times. Furthermore, the calibration step might need to performed multiple times to get good results. For validation the computational effort is less of an issue because of the non-iterative nature.

What the right balance is between the number of scenarios to include and the effort it requires depends on many things:

- The desired degree of insight into the model's quality. The more insight you want to get into the model's quality, the more validation scenarios need to be included. For calibration, more scenarios might increase the model's accuracy (i.e. the quality of the optimal parameter set). However, this is not necessarily the case.
- The computational efficiency of the model. The faster a model, the more simulations it can perform in the same amount time given a certain amount

of computing power. Hence, for fast models, adding more scenarios results in less added computational effort than for slower models.
- The amount of time available for scenario preparation. Each scenario that is included needs to be prepared. This involves implementing the scenario, estimating and measuring all relevant inputs and preparing the reference data set.
- The amount of time and computer power available for the simulations. Each included scenario requires many (hundreds of) additional simulations during the calibration. This is the number of calibration iterations times the number of replications necessary during each single iteration. Section 4 describes how to determine the required number of replications. For validation this results in only a few additional simulations, namely, the number of replications.

The second criterion for the selection of the scenarios is the balance between the scenarios. This is mainly relevant for calibration because if all scenarios weight equally, and certain cases are more represented than other cases, the calibration might be biased towards these cases. This can completely or partly be solved via the selection of scenarios. But it can also be completely or partly solved via the optimization algorithm. Generally, you want a good mix and is it questionable if many scenarios that are relatively similar has much added value. Also, in relation to the other criterion.

The advice is to select a good mix of the most highly ranked scenarios which cover all, or most, of the relevant and important behavior and critical cases for the calibration. For validation, the advice is to select a far larger set which includes the most critical cases and a good mix of less critical cases. Then perform the calibration and validation. Finally, review the validation results and determine if there are cases which are not included in the calibration set but do perform weakly in the validation. If this is the case, it might be worthwhile to include these in the calibration set and recalibrate the model with this extended set.

Overall, we can summarize the process of choosing calibration and validation scenarios for both walking models and the other types of pedestrian models (e.g. route choice models) as the following set of steps to perform:

1. Identify the relevant modelled behaviors given the intended us of the model.
2. Identify which characteristics of a simulation scenario (e.g. geometry of the infrastructure, demand patterns, other pedestrians, information such as signs, properties of pedestrians etc.) influence this modelled behavior.

3. Systematically create a full list of scenarios based on the identified scenario characteristics.
4. Rank the list based on the richness of the data of the scenarios and the importance of the scenario for the intended use.
5. Prune the list of scenarios by first removing scenario for which there is no reference data available. Then potentially remove more scenarios to achieve a balance between the number of scenarios to include and the (computational) effort necessary to complete the calibration and validation and to ensure a balance between the different scenarios

3.4.2 Choosing metrics

A metric is a function (m) that takes the simulated data (Y_{sim}) and the reference data (y_{data}) and computes the difference between the two (see Fig. 5). This results in an error (ε) that represents how accurately the model can reproduce a certain scenario given the input (x) and the chosen parameter set (θ). A metric has two parts, the variable of interest and the difference measure.

A variable of interest represents the "what" of the comparison. Which aspect of the behavior or the resulting flow is compared. For example, the flow at a certain cross-section, the distribution of the travel times or the exact trajectories. The difference measure determines "how" the difference between the variable of interest from the simulation and data is computed. For example, computing the difference using a Root-Mean-Squared-Error (RMSE) or computing the difference between two travel time distributions using the Kolmogorov-Smirnov (KS) statistic. For each metric you also need to define where and when to measure it. That is, you need to define the measurement area/location and the measurement period.

So, when choosing the metrics, three choices must be made. What variables to compare, what is an appropriate difference measure for each variable and where and when to measure each variable. And like the choice of scenarios there are many possible metrics. And like the choice of scenarios, the choice of metrics can severely impact the calibration and validation (Sparnaaij et al., 2019).

The choice of metrics, like the choice of scenarios, depends strongly on the type of model. This primarily holds for choosing the variables of interest and defining the measurement periods and areas. For example, for route choice models this will be different than for walking model. In line with the scenario choice section, we focus on the variables of interest of walking models.

3.4.2.1 How to choose the variables of interest

Many different variables have been and can be used to describe the behavior pedestrians and dynamics of pedestrian flows. These range from very detailed variables such as the trajectories of individual pedestrians to very aggregated variables such as the average flow through a bottleneck. The main difference between the variables is the aggregation level at which they describe the behavior and dynamics. Macroscopic variables describe the collective behavior, for example, in terms of the flows and fundamental diagrams. Microscopic variables, on the other hand, describe the behavior on the level of an individual pedestrian. So, for example, in terms of the trajectory of an individual pedestrian or the individual travel time. Lastly, the mesoscopic variables sit in between these two levels. They describe the behavior in terms of distributions of individual behavior. For example, the distribution of individual travel times.

Given that there are a large number of variables of interest the question is: How to choose the right variable/variables of interest? This depends on the model, the intended use of the model and the scenario. In short, which behaviors are modelled, at what aggregation level and what behaviors and dynamics are important to capture accurately given the intended use of the model. Given the large possible number of variables of interest and the wide range of models and intended usages, there is no detailed method that can be followed to choose the variables of interest. However, we describe several aspects that should be considered when compiling the list of variables to use.

Firstly, a model calibrated using a single metric does not necessarily reproduce other variables accurately (Campanella, 2016; Duives, 2016; Sparnaaij et al., 2019). This is especially the case for variables that describe different behaviors and dynamics. For example, a model calibrated to reproduce the flow through a bottleneck accurately, does not necessarily reproduce the spatial distribution pattern in front of the bottleneck accurately. Or, a model calibrated to accurately reproduce the travel time distribution does not necessarily reproduce the distribution of the effort accurately.

Secondly, the aggregation level of the variables plays a critical role in determining their applicability to different models. Not every variable can be used universally across all models. For instance, macroscopic models do not distinguish individual pedestrians so metrics that are based on the behavior or dynamics of individual pedestrians (i.e. most mesoscopic and microscopic metrics) cannot be used. Simply put, the lower the level of aggregation of the model, the more variables can be used.

Furthermore, a model calibrated using only variables of a specific aggregation level does not necessarily reproduce variables of other variables accurately. For example, a model calibrated using trajectories as the variable of interest, that is calibrated on the individual behavior of pedestrians does not necessarily reproduce the collective/aggregate dynamics such as the flow.

Thirdly, variables can be strongly correlated. For example, the travel time and the speed are strongly correlated. Hence, if you use one, adding the other does not provide any new information about the model quality. Therefore, it is not efficient to include both. However, you can sometimes use these correlations to your advantage. Namely, if one of the two is a normalized version of the variable. In the example, the speed is a normalized version of the travel time, normalized by the distance of the travelled path. This makes is easier to compare results from pedestrians from different scenarios with very different path lengths. At the end of this section we go into more detail about the importance of normalization.

Lastly, different scenarios can have different behaviors and dynamics that are important. This should also be reflected by the variables of interest that are used for each scenario to form the objectives. Some variables of interest such as the fundamental diagram or the travel time distribution are relevant to most, if not all, scenarios. Others, such as the spatial distribution are very relevant in, for example, bottleneck scenarios or corner scenario but much less relevant in straight corridor scenarios.

In short, it is important to include multiple variables of interest when calibrating and validating a pedestrian model. These variables should capture different behaviors and dynamics at different aggregation levels (if relevant given the model) which are relevant given the intended use of the model and relevant to the selected scenarios whilst preventing overlap of strongly correlated variables.

Two additional things should be noted. Firstly, many variables that are used to describe pedestrian behavior are quantitative variables. However, qualitative variables are also used to describe pedestrian behavior. For example, to determine if a pedestrian model shows certain self-organizing phenomena like lane formation. This is hard to capture quantitatively but can easily be determined by visually comparing the data and the simulation results. The fact that qualitative variables require human inspection means that they generally cannot be used during calibration, due to the iterative nature of calibration. For validation they can be used though.

Secondly, the choice of variables strongly depends on the available data. If the variable cannot be computed based on the available data, it cannot be

used for calibration and validation. And thirdly, the more disaggregate the variable the more precise the input to the simulation needs to be to prevent large errors caused by imprecise inputs. In Section 3.5.2 we discuss this aspect in more detail.

3.4.2.2 Difference measures

For each variable of interest, a difference measure must be chosen. A function that takes the simulated data from all replications and the reference data and produces a single error value that quantifies how much the simulation results differ from the reference data for the given parameter set. Just like the variables of interest there are many differences measures to choose from.

There are several aspects that differentiate these options which we discuss in more detail below.

First, the variable of interest limits the options to choose from but unlike the variables of interest, the choice does not depend on the model type. Variables of interest that are single values, for example the total evacuation time, require different difference measures than variables that are distributions or time series. In the case of single value variables, a common difference measure is the root mean squared error (RSME) (Hyndman and Athanasopoulos, 2018) which is computed as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x_0)^2}$$

Where n is the number of simulated values, one per replication, x_i is the simulated value of replication i and x_0 if the reference value from the data. Other variations are the mean squared error ($MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_0)^2$) and the mean absolute error ($MAE = \frac{1}{n}\sum_{i=1}^{n} x_i - x_0$) (Hyndman and Athanasopoulos, 2018).

Although these measures are all based around the absolute difference between the simulated values and the reference value the interpretation of their error score differs significantly. The MSE represent the variance of the simulated values in with the value from the data being the mean. The RMSE is the standard deviation whereby the difference with the MSE is that large deviations are penalized less severely. Lasty, the MAE is the average deviation from the mean (i.e. the value from the data).

To show how this difference leads to different error value we apply the three difference measures to two different sets of values. Table 1 presents the two examples and the resulting error values. The main difference between the two examples is that in the first example the errors are symmetrically arranged around the reference value whilst in the second case they are all to one side of the reference value. The MSE and RMSE are the same in both examples which is expected given that the magnitude of the deviations in the same in both examples. The MAE differs though, in the first example it is 0 which is expected as the deviation cancels each other. In the second example it clearly shows that the results from the simulations are biased and result in a deviation in a particular direction.

This example shows a fundamental difference between the MSE and RMSE on one side and the MAE on the other side. The MSE and RMSE are primarily a measure of the precision of the simulation results. The MAE on the other hand is primarily a measure of accuracy. Ideally, you want your simulations to be both so none of the measures is a perfect difference measure. A solution is to combine them and for example compute both the RMSE and the MAE for a variable.

The difference measures above all use absolute errors but the difference between the simulated values and the reference value can also be expressed as relative errors. For example, in the form of the mean relative error ($MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{x_i}{x_0}$). The main advantage of using relative errors is that they provide a normalized error. This makes combining of different variable and scenarios easier. However, this assumes that for different variables and scenarios, a deviation of one time the reference value from the reference value is equally bad for all variables and in all scenarios. Unless this assumption holds, relative errors also still need to be normalized. Furthermore, relative difference metrics do not work well when the reference value is 0 or close to zero as the errors go to infinity. For more information about these types of difference measures the reader is referred

Table 1 Example presenting the difference between the three related difference measures.

	$x_i \in \{-2, -1,1,2\}$, $x_i = 0$	$x_i \in \{2,1,1,2\}$, $x_i = 0$
MSE	2.5	2.5
RMSE	1.6	1.6
MAE	0.0	1.5

to the more general model forecasting literature (e.g. (Hyndman and Athanasopoulos, 2018; Hyndman and Koehler, 2006; Kolassa and Siemsen, 2014)).

Variables of interest that are distributions, such as the distribution of travel times, require a different approach. For example, statistical methods such as the Kolmogorov-Smirnov (Massey Frank., 1951) test can be used to quantify the difference between the distribution of simulation values and the reference distribution from the data. In most cases, replications are required resulting in multiple distributions which can be combined into a single distribution containing the values of all replications. Another option is to describe the distributions using single values such as the mean and the standard deviation and use the methods described above to compare these values and get a single error value. Both (Ronchi et al., 2013) and (Sparnaaij et al., 2019) provide an example how this method of comparing distributions can be applied.

Lastly, for variables that are curves, such as timeseries, (Ronchi et al., 2013) present three functional analysis concepts that can be used to quantify the difference between the curve from the simulation and the curve from the reference data. These are the Euclidean Relative Difference (ERD), the Euclidean Projection Coefficient (EPC) and the Secant Cosine (SC)). For their detailed specifications the reader is referred to (Ronchi et al., 2013). In the case of time series you cannot combine the results from multiple replications into one curve. Hence, you get an error for each replication and these errors then need to be combined into one error score using one of the methods used for single value comparisons. Here the error from an individual replication replaces $x_i - x_0$ in these equations.

All in all, for different types of variables (single value, distribution and time series) there are many difference measures you can apply. What difference measure, or combination thereof, to choose for what variable depends on what deviation from the reference data you judge to be most important to minimize to get the most accurate model for the intended application.

3.4.2.3 Defining the measurement area and period

The metric defines what is measured and how the simulation output and reference are compared. But another vital aspect is where and when is the variable of interest measured. For each variable you need to define the measurement area and the measurement period. This must be done for each combination of a variable and a scenario.

It is vital that the measurement area and periods match between the reference data and the simulations. The measurement areas should be exactly the same in both the data and the simulations. The measurement periods can differ is, for example, the simulation needs a warm-up period.

The reason why it might be relevant to define the measurement area to be smaller than the area covered by the reference data and the measurement period shorter than the duration of the reference data is to measure only the behaviors and dynamics relevant to the scenario. For example, you have a scenario with a bidirectional flow in a corridor with a certain density level. The reference data for this scenario contains a warm-up period where the flow is not always bidirectional and the density level is not yet at the required density level. By limiting the measurement period you assure that any errors between the data and the simulations are errors related to how the model models bidirectional flows at the given density level. Similarly, if the data also covers the entrance and exits of the corridor where next to a bidirectional flow there are also exiting and entering flows, you will want to limit the measurement area. This ensures that any errors can be attributed to the model not reproducing the bidirectional behaviors and dynamic and thus preventing ambiguity.

3.4.3 Normalization

A core aspect of the search for the optimal parameter set using multiple objectives set is comparing the error scores of all these objectives to each other. Therefore, it is vital that these error scores are comparable. Otherwise, the calibration and validation might be unintentionally biased towards certain objectives. The differences are caused by the fact that different variables of interest have different units and orders of magnitude which means the error values can also differ in their units and orders of magnitude. But, also by the fact that different difference measure result in different error values with different units and orders of magnitude. And lastly, different scenarios and thus different reference data means different reference values.

For example, if one of the objectives is to minimize the absolute error in the average specific flow at a certain location and the other is to minimize the absolute error in the mean travel time between two location you have two errors with different units and different orders of magnitude. The absolute error in the specific flow has a unit of ped/m/s and has an order of magnitude of around 0.1 to 1 ped/s/m whilst the absolute error in the mean travel time is measured in seconds and can easily have an order of

magnitude around hundreds or thousands of seconds depending on the walking distances and the size of the modelled area. Or, the RMSE of the mean flow needs to be compared to the Kolmogorov-Smirnov statistic which results from comparing the travel time distributions.

The challenge in normalizing all the error values for all objectives is to determine what error in objective x is equally bad as what error in objective y. In (Sparnaaij et al., 2019), all error values are normalized using a metric specific reference value. So, a deviation from the reference value of magnitude a of metric x is equally bad as a deviation of magnitude b of metric y. What these references values should be is a judgement call whereby the intended use of the model and the values from the reference data can guide the choice.

3.5 Reference data, inputs and estimation of inputs

The reference data is a key element of the calibration and validation effort. It is not only the basis to which the simulation results are compared. It also determines the exact design of the scenario via the input to the model.

3.5.1 Reference data

There are two main categories of reference data you can use for calibration and validation. Values from the literature and observational data from experiments or real-life. For a detailed description of the various sources and features of pedestrian behavior data the reader is referred to chapter 4.

In the pedestrian dynamics literature, many papers describe data from experiments or real-life observations. For example, papers describe observed bottleneck capacities, observed flows or fundamental diagrams. When these papers also provide enough context about elements such as the geometry and inflows into the area, they can be used to build a simulation scenario. The results of simulations of these scenarios can then be used to compare to the values from the paper.

The advantage of this type of reference data is that it is easy to use (no need to process the data) and is readily available. However, generally the data is not very detailed and limited to macroscopic metrics and thus limited in the level of detail. Furthermore, the often limited description of the context results in significant uncertainties in the input. For example, if no detailed description is given of the inflow into the area this provides a high level of uncertainty. This uncertainty can negatively impact the calibration and validation as the next subsection explains.

Observational data from experiments and real-life provide a more detailed way to calibrate and validate pedestrian models. And nowadays many different data sets are available. Especially the pedestrian data archive (Institute for Advanced Simulation 7: Civil Safety Research of Forschungszentrum Jülich. 2025) is a rich source of reference data. Within the observational data, you can choose between using data from experiments or data from real-life observations.

Data from experiments has the advantage of being more controlled. The exact geometry, inflows and flow type are controlled and generally well described. This makes this type of data more suitable for calibrating and validation specific scenarios. Furthermore, data from experiments often provide more context, for example in the form of socio-demographic data. However, the disadvantage of experimental data is the fact that participants know they are part of an experiment and are being observed. This might influence their walking behavior resulting in less natural behavior than can be observed in data from real-life.

This is the major advantage of data from real-life, more natural behavior. However, the situation is not controlled and therefore the data is less clean with regards to the scenarios. It can have more variation in the flow types that exist in the data as well as differences in the flow ratios in the case of multidirectional flows. Furthermore, only little information about the population is available.

The availability of reference data strongly determines which objectives can be included in the calibration and validation. Therefore, use any reference data that is available despite the major limitations some sources of data have. However, be aware of those limitations and take difference between the quality of the reference data into account during the calibration and validation. For example, when defining the objective function during calibration (see Section 3.6.2) or when interpreting the validation results. Furthermore, for calibration it is especially the case that the principle of garbage in, garbage out holds. If the reference data is of bad quality, for example it contains big measurement errors, or lacks a lot of contextual information relevant for the input (see next section), this will potentially result is bad calibration results.

3.5.2 Inputs and estimation of inputs

For each calibration and validation scenario, the model needs input which includes the geometry, the demand patterns and the route splits among other things. The input is determined by the reference data (see both Figs. 4, 5 and 6). Some inputs are given, for example, the geometry. Other

inputs need to be measured, for example, the route splits and the demand pattern. And some inputs need to be estimated based on the output. For example, the preferred speed.

For both calibration and validation, it is vital that the input is accurate. The equations in Fig. 5 provide insight into why this is the case. The core of each calibration and validation effort is the quantification of the difference between the model output and the reference data. That is the error (ε). This error is caused by a combination of the three elements that determine the model output. The model itself, the parameter set (θ) and the model input (x).

In calibration and validation, the error is used to assess the model quality, given a certain parameter set. The more accurate the input is during validation, the more likely the computed error can be only/primarily attributed to the model and the parameter set. Therefore, the better the error represents the quality of the model for the given parameter set and hence the more accurate the calibration and validation results will be.

The work by (Benner et al., 2017) is a rare example where this effect is studied. Their research shows clearly that improving the input can improve the fit of the model to the reference data during calibration. However, they also point out that due to missing details in the used dataset they had to make several assumptions. This meant that their input still contained a significant level of uncertainty and consequently, as the researcher themselves also remark, limits the level of certainty that the found parameter set is the optimal parameter set. Hence, when designing experiments to collect data that will (possibly) be used to calibrate and validate models, it is vital to consider what details would all be necessary to accurately estimate the inputs.

Besides accurate, model inputs should also be at the right level of detail. How detailed the inputs should match the aggregation level of the metric. For example, when microscopic metrics are used, the location and time at which pedestrians enter should match the reference data. Otherwise, a large error source is introduced. In the case of mesoscopic metrics, the output is stochastic (a distribution) hence a lower level of detail can be used for the input. The demand over time and per location should match the reference data as closely as possible but the exact position and time each individual pedestrian enters is not as relevant.

3.6 Search spaces and optimization algorithms

The main goal of the calibration process is to find the optimal parameter set out of a collection of possible parameter sets. The collection of

possible parameter sets is called the search space. An optimization algorithm searches through this search space to find the optimal parameter set. When calibrating a model, the search space must be defined, and the optimization algorithm must be selected. We first explain how to define the search space and then discuss how to choose an optimization algorithm. This section is primarily relevant for model developers or models user that must perform a (re)calibration of a pedestrian model. Furthermore, the techniques and methods we in this section are applicable to all types of pedestrian models.

3.6.1 Defining the search space

The search space defines the parameter sets among which an optimization algorithm searches for the optimal parameter set. Ideally, the search space contains the combination of every feasible value of every parameter. However, this would be an infinitely large set of parameter sets which would be impossible to search through in a reasonable time. So, the challenge is to define a search space which is extensive enough that a parameter set can be found which results in a good fit of the model to reality but that is small enough that this parameter set can be found within a reasonable time.

The size of the search space is determined by the number of parameters that is included and the number of values per included parameter. This number per parameter is itself a function of the chosen lower and upper boundaries and level of precision. So, to limit the search space size, only the most influential parameters should be included. And for those included parameters, reasonable boundaries must be chosen and an appropriate level of precision must be chosen.

To determine if a parameter is influential and thus must be included a sensitivity analysis should be performed. This sensitivity analysis will show how sensitive the model output is to changes in this parameter. The more sensitive the model is the parameter, the more influential it is and thus the more important it is to include this parameter in the calibration. Section 5 provides more information on how to perform sensitivity analyses for pedestrian models.

There is no strict method for defining reasonable boundaries for each parameter. However, as a guideline, the boundary should be the point at which the parameter value leads to an unrealistic model or unrealistic behavior. Depending on the parameter this can be determined based on the model structure and interpretation of the parameter. For example, in a

microscopic model, the lower bound of the radius can be chosen such that the maximum possible density cannot exceed a feasible maximum value. Similarly, a maximum radius can be chosen.

For parameters where there are no obvious lower and/or upper boundaries, the results of a sensitivity analysis can be used. Depending on the type of model, different measurements can be used to determine if the behavior is realistic or not. For example, in any model, unrealistic high flows, speeds or densities indicate unrealistic behavior. In microscopic models, other elements such as pedestrian moving through each other or obstacles also indicate unrealistic behavior.

There is also no strict method for defining the level of precision of a parameter. The level of precision is the step size between subsequent parameter values and thus determines how many values of the parameter are included between its upper and lower boundary. Again, the sensitivity analysis can provide guidance. The more sensitive a parameter the more precise the parameter value should be estimated. Also, the maximum desired size of the search space can be used to guide how many values you would maximally like to include and thus what the step size should be given the boundary values.

3.6.2 Choosing an optimization method

To find the optimal parameter set in the search space, an optimization algorithm searches through the search space using the errors of all objectives to determine the optimality of each tested parameter set. There exist many different optimization algorithms that can be applied to calibrate pedestrian models. These can be classified into single objective optimization algorithms (e.g. simulated annealing) which require the different objectives to be transformed into a single objective in each iteration of algorithm. And methods that can optimize using many objectives and produce an approximation of the Pareto front. Both are discussed in more detail below.

3.6.2.1 Single objective optimization

When you choose a single-objective optimization algorithm to calibrate a pedestrian model using multiple objectives the first step it defining the objective function. The objective function combines the errors of the different objectives into a single error. The optimization algorithm then tries to find the parameter set that minimizes this error.

How you combine the objectives into a single objective has a major impact on the calibration. It determines how influential each objective is

compared to all other objectives. A common approach is the weighted sum method (Kochenderfer and Wheeler, 2019). Every objective is assigned a weight, and the resulting objective function is the sum of all weighted objectives. The following equation shows this:

$$\varepsilon = \sum_{i=1}^{n} w_i \varepsilon_i$$

where n is the number of objectives, ε_i is the error of the i-th objective and, w_i is the weight assigned to the objective. The simplest version of his approach is to assign each objective the same weight. However, you can also choose to assign different weights to each objective. There are two main reasons to assign different weights. First, to balance the objectives when there is an imbalance in the set of objectives. Second, to align the weight of the objectives with their importance to the intended use of the model.

The first case is relevant when the set of objectives is imbalanced. This can be the case when, for example, due to a difference in data availability, some movement base cases are underrepresented in the set of objectives. For example, you want to calibrate your model for both bidirectional and crossing flows in both high and low densities. For the bidirectional movement base case you have data for both high and low densities but for the crossing case you only have data for the high-density case. If you use the same metrics for all scenarios, you have twice as many objectives for the bidirectional case as for the crossing case. However, you deem both movement base cases to be equally important. So, to ensure that the objective function reflects this, you need to make the weights for the crossing objectives twice as big as for the bidirectional objectives.

The weights can also be used to reflect that the model should be most accurate in certain scenarios and for certain metric given the intended use of the model. Giving these objectives higher weights ensures that the optimization algorithm will prioritize a parameter set that minimizes the errors for these objectives more than for other objectives.

There is no predefined method or strict strategy for choosing the weights. However, the two examples above provide two elements you need to consider when choosing the appropriate weights. The two key aspects that determine the choice of weights are the set of objectives that need to be combined into one objective and the intended us of the model which determines which objectives are most important. Note, that it is important that all objectives have been normalized (see Section 3.4.2) to ensure that they are comparable.

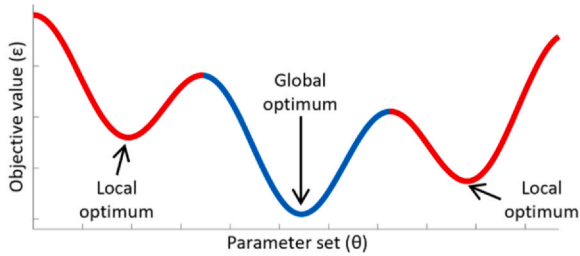

Fig. 7 Example showing local and global optima.

Although combining the errors of multiple objectives into one error value is not necessary for the validation process, it can still be useful. For example, when you want to compare different models, different versions of the same model or different optimal parameter sets for the same model and you want to quantify the model's validity using a single value.

A key aspect of the combined objective function of pedestrian models is that they are often non-linear and contain local optima. Especially local optima can be a challenge for certain types of algorithms. Fig. 7 shows an example of an objective function with two local optima. If a gradient decent optimization algorithm starts on any the points in the red parts of the objective function it will end in one of the local optima. As such it will not correctly identify the optimal parameter set that is found at the global optimum point. In short, any optimization algorithm you choose should be able to avoid getting stuck in a local optimum as well as be able to deal with a non-linear objective function.

We don't discuss all possible algorithms that fulfil these requirements but provide a few examples of commonly used algorithms, within the field of pedestrian modelling, with different characteristics. These are the grid search algorithm, genetic algorithms and maximum likelihood optimization algorithms. For more information on optimization algorithms, an overview of the variety of optimization algorithms that exist and their characteristics, the reader is referred to (Kochenderfer and Wheeler, 2019).

3.6.3 Grid search algorithm

A grid search algorithm does an exhaustive search through all possible points on a grid of parameter values (i.e the search space) to find the optimal point. The grid is defined by the unique combinations of all selected values of all parameters. This process uses three major steps.

First the grid needs to be created which itself requires two steps. First, for each parameter you must define the values to include. Generally, this means selecting an upper and lower boundary and a step size that discretizes the space between the boundaries. Second, you take the values of all K parameters and create a K-dimensional grid which contains all unique combinations of all values of all parameters.

The second major step is to run N * M simulations, where N is the number of scenarios and M is the required number of replications for each point in the grid. Then, as the third and last step, you compute the objective value for each parameter set using the results from the N * M simulations and the reference data of each scenario. The grid point that has the objective value with the smallest error represents the optimal parameter set.

The major advantage of the grid search algorithm is that it guarantees the optimal parameter set is found (within the given grid). However, it is computationally expensive especially when many parameters need to be optimized. Furthermore, the algorithm cannot find any optimal parameter set that lies between grid points. So, while it always finds the global optimal point on the grid this might not be the actual global optimum that would be found if the grid is less coarse.

Given the high computational burden the grid search method requires, it is often not the best choice for calibrating a pedestrian model which usually has many parameters. However, the method has is uses. For example, to gain insight into the shape of the objectives space which requires a good distribution of the sampled point over the space. Or, to fine-tune the value of a few parameters after a basic calibration effort. Both the work of Sparnaaij et al. (2019) and Duives (2016) shows how the grid search method is used to calibrate a pedestrian model within this context.

3.6.4 Genetic algorithms

Genetic algorithms are biology inspired algorithms that use evolutionary principles like mutation and crossover to search though the search space to find the fittest (i.e. optimal) solution. The basic principle is that the parameters that need to be calibrated are each translated into a genetic representation called a genome which combine into a chromosome. These chromosomes are then used to find an optimal solution using the following steps:

1. **Initialization**: An initial set of chromosomes is created which forms the first generation of solutions. A common approach is to distribute the chromosomes uniformly over the search-space.

2. **Evaluation**: The fitness of each chromosome is evaluated using the objective function. The lower the error from the objective function, the fitter the chromosome. This step involves running N * M simulations per chromosome where N is the number of scenarios and M the number of replications.
3. **Reproduction**: A certain number of chromosomes are selected for reproduction based on their fitness. Using cross-over and mutation a new set of chromosomes is created.
4. **Replacement**: The new chromosomes replace a part of the current population, to create a new generation of solutions. The choice of which chromosomes are replaced depends in part on their fitness, the less fit the more likely the chromosome is to be replaced. However, to prevent getting stuck in a local optimum, it is also important to maintain a diverse population.
5. **Iteration**: Steps 2–4 are repeated until a stopping criterion is met. This stopping criterion can be:
   - ***Error criterion***: A solution (chromosome) has a fitness (error) smaller than a predefined threshold. Using this criterion ensures that a given level of accuracy has been reached but not that an optimum, local or global, has been reached.
   - **Convergence**: New solutions of subsequent generations are not significantly fitter than the fittest solution. This indicates that an optimum has been found but it is not guaranteed that this is the global optimum.
   - **Fixed stopping point**: A fixed number of generations has been reached or the allocated budget (computation time/money) is spent.
   - **Combination**: A combination of the three options above can also be used.

The optimal parameter set is the best solution (fittest) that has been found during all iterations.

Genetic algorithms have the advantage that their computational burden is much lower than the grid search algorithm and thus can handle larger (more parameters, less coarse) search spaces. However, it is not guaranteed to find the global optimum, given the non-exhaustive nature of the algorithm.

In the literature, genetic algorithms or variations such as differential evolution are commonly used to calibrate pedestrian models. The work by Zhong and Cai (2015) provides an example of how differential evolution combined with a sensitivity analysis is used to calibrate a pedestrian model in a single-objective case. The work by Wolinski et al. (2014) provides a multi-objective example where a genetic algorithm is used.

3.6.5 Maximum likelihood optimization algorithms

Maximum likelihood optimization algorithms use a likelihood function, commonly the log-likelihood, that describes the probability that the model describes the observed data for the given parameter set. In pedestrian model calibration, it has generally been used in one specific context. Namely, to calibrate a microscopic pedestrian model using trajectory data whereby the observed and simulated trajectories of individual pedestrians are compared (e.g. Daamen and Hoogendoorn (2012)).

Here, trajectories from multiple pedestrians from multiple scenarios can be used to calibrate the model in a multi-objective manner. However, to the authors' knowledge, the literature includes no examples where other metrics are incorporated. Hence, its proven use in calibrating pedestrian models is limited to this context. There are examples, where a non-microscopic model has been calibrated using maximum likelihood optimization (e.g. Hänseler et al. (2017)). However, these are single-objective calibration efforts.

3.6.6 Combinations

The work by Wolinski et al. (2014) shows that another option is to combine two different optimization algorithms to take advantage of the strength of both algorithms. An example of this is the combined use of a genetic algorithm with a greedy algorithm. The genetic algorithm is first applied to ensure a broad search of the search space, and the result is used by the greedy algorithm to refine this result.

3.6.6.1 Many-objective optimization

Instead of transforming multiple objectives into a single objective and applying a single objective function, you can also apply a multi-objective optimization algorithm. These revolve around finding the Pareto-optimal solutions that together from the Pareto front. We use the example in Fig. 8 to shortly explain both the Pareto-optimal solutions as well as the Pareto front.

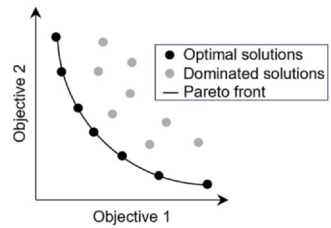

Fig. 8 Example of Pareto-optimal solutions and the Pareto front.

In the example we have two objectives. Each point in the graph represents a solution (i.e. a parameter set) and each solution has a certain error value for both objectives. The lower the error the better the solution is with regards to that objective. Pareto-optimal solutions are the points where the error for an objective cannot be reduced without increasing the error for another objective. These are also called non-dominated solutions, meaning there's no other solution that has a lower error value in all objectives. Non-optimal solutions are called dominated solutions as there is at least on point that has a lower error for one objective without it having worse errors for other objectives. Hence, this point in dominated by that other point. The Pareto-optimal solutions form the Pareto front. Each point on the front is optimal but with a difference balance between the objectives. The optimal solution is the point on the front that the user perceives as the best trade-off between the different objectives.

There are cases where a pedestrian model has been calibrated using a multi-objective algorithm. For example, the work by Zeng et al. (2017) where the NSGA-II, a modified genetic algorithm, is applied to obtain the Pareto-optimal solutions. Multi-objective algorithms generally work well when there are two or three objectives Li et al. (2018). However, pedestrian model calibration will generally have many more objectives than that. This is the domain of many-objective optimization for which various optimization algorithms exist Li et al. (2018). To the authors' knowledge, there exist no examples in literature of the application of these types of algorithms for pedestrian model calibration. So, the question is still open as to how well these methods would perform and how well the resulting Pareto front can be interpreted and used to select the optimal parameter set.

3.6.6.2 Conclusion

Overall, the choice of an optimization algorithm or combination thereof, the configuration of the optimization algorithm and the chosen stopping criteria is always a balancing act between:

- The certainty that the optimal parameter set (or a very good approximation) is found
- The computation effort required to find the optimal parameter set.
- The number of parameters that are calibrated.

Generally, the more certainty you want to more computationally expensive the calibration will be. And the more parameters the more computationally expensive the calibration procedure will be if you want to obtain the same level of optimality/accuracy.

Furthermore, the choice of the algorithm, its stopping criteria and its configuration should also reflect what objectives are more important than others given the intended application of the model. This can, for example, be achieved by using different weights when transforming multiple objectives into a single objective when a single objective optimization function is selected as the optimization algorithm.

Lastly, it is important to note that the two most important factors that determine the computational burden of the calibration are the cost of running a simulation and the number of scenarios. So, the slower the model you calibrate, the more computationally expensive the calibration. And the more behaviors and contexts the model must capture well, the more scenarios you need and thus also the more computationally expensive the calibration.

4. Stochasticity and replications

Most pedestrian models contain one or more sources of stochasticity. These are parameters or inputs that are described by a distribution to capture, for example, difference in the preferred speed of pedestrians in a population. Or random variables such as the fluctuation term commonly used in the social force model Helbing and Molnár (1997). These stochastic elements cause the output of the model to be stochastic as well. So, the simulation outcome differs between simulations even if they have exactly the same inputs and parameters. And, thus the output of the model is described by a distribution that described all possible outcomes given certain inputs and parameters.

Consequently, for any model that has one or more sources of stochasticity you need to perform replications. That is, repeat the simulation multiple times with the same input and parameters. This ensures that you have a sample of possible outcomes that describe the distribution of possible outcomes and thus captures the effect of the stochastic elements on the output well. The main question is, how many replications do you need to run? This can be determined in two ways, using the student *t*-test and by using convergence criteria. We discuss both in more detail. However, for both the same principle applies. The number of replications should be high enough that the sample of simulation outcomes approximates the distribution of all possible outcomes to the desired level of accuracy.

4.1 Determining the number of replications using the student *t*-test

The number of replications can be computed a-priori using the student *t*-test with the following equation (Dekking et al., 2005):

$$N = \left(\frac{S_{N_0} t_{\alpha/2}}{d}\right)^2$$

Where S_{N_0} is the sample standard deviation of the output computed based on N_0 simulations, $t_{\alpha/2}$ the critical value of the t-distribution at significance α and d the allowable error between the sample mean and the mean of the actual output distribution (the one you would obtain if you would run infinite replications).

A limitation of this method is that the student *t*-test assumes that the distribution of the simulation outcomes is a normal distribution. This is not necessarily the case for the outputs of pedestrian models. Therefore, you must test if the outputs are normally distributed when you apply this method. Furthermore, the value of N_0 is not a fixed or given value. It must be chosen such that S_{N_0} is a good representation of the standard deviation of the distribution of the outputs.

The challenge of determining the correct value for N_0 can be avoided by applying the method iteratively. You start by running several replications. You then check if the current number of replications (N_i) is greater than or equal to the number of replications you require according to the student *t*-test. Here, you compute the sample standard deviation (S_{N_i}) based on the outputs of all replications so far. If more replications are required, you run several more replications and check again. You continue to do this until the number of replications is sufficient ($N_i \geq N$). The following equation represents the equation you try to solve:

$$N_i \geq N = \left(\frac{S_{N_i} t_{\alpha/2}}{d}\right)^2$$

4.1.1 Determining the number of replications using convergence criteria

The work by Ronchi et al. (2013) presents an alternative method for determining the number of replications. They use convergence criteria to determine if the current distribution of outputs has converged to a stable distribution. That is, data from additional replications will not significantly

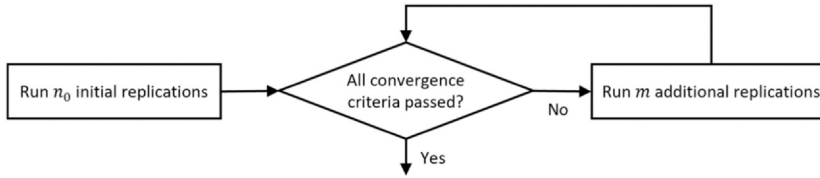

Fig. 9 Convergence method to determine the number of required replications.

change the distribution and therefore the current distribution is a good representation of the actual output distribution.

The method is summarized by the following equation:

$$diff\ (D_n,\ D_{n-i},\ p_{threshold}) = \text{not significant}\ \leq\ \forall i \in [1,\ b]$$

After n replications, check if the distributions, containing the results of all $n - i$ replications (D_{n-i}) are similar to the distribution containing the results of all n replications (D_n) according to a difference function *diff* (.) and a chosen threshold value $p_{threshold}$. If one or more distributions differ significantly run additional replications and perform the checks again. Repeat this until none of the distributions differ significantly. Fig. 9 present the method visually.

There are various options for the difference function. Ronchi et al. (2013) use two different convergence measures to test if the distribution of their outputs, the total evacuation time in their case, has converged. The two criteria are the relative difference in the mean and the relative difference in the standard deviation of the distributions. The relative differences should be smaller than the given threshold value. Sparnaaij et al. (2019) use the Anderson-Darling test to test if the distributions differ significantly.

The example of Ronchi et al. (2013) shows that you can use different criteria to test if the distribution has converged. In this case, the number of replications is sufficient if the distributions do not differ significantly according to all convergence criteria.

4.1.2 Replications and pedestrian model calibration and validation

If a pedestrian model is stochastic or has a stochastic input, this also impacts calibration and validation. Each scenario must be replicated to deal with the stochasticity. This especially impact the calibration because you need multiple replications of each scenario during each calibration step.

The number of replications you need in each step and for each scenario depends on the scenario itself, the input, the parameters and the metrics. Therefore, you must compute the required number of

replications for each combination of scenario and metric (i.e. each objective) separately and run as many replications as the metric with the highest number of required replications needs. In the case of calibration, the number of replications must be computed for each objective in every iteration step, because in each iteration of the calibration process the parameters change.

5. Sensitivity analysis

A sensitivity analysis is a method used to determine how changes in the parameters of a model impact the output. It especially focusses on determining how strongly the output changes when a parameter value is changed with a certain step size. The more strongly the output changes with a small change of a parameter value, the more sensitive the model is to changes to that particular parameter.

Information about the sensitivity of all parameters is essential for a good and efficient calibration of a pedestrian model. It provides insight into the ranges that should be included in the search space for each parameter and it provides insight into whether or not it is necessary to include the parameter in the calibration in the first place. The smaller the ranges and/or the fewer parameters included in the search space, the smaller the search space is, and the fewer iterations are required during the calibration.

There are two main approaches to a sensitivity analysis. A one-at-a-time sensitivity analysis and a more-at-a-time sensitivity analysis. In a one-at-a-time sensitivity analysis the value of one parameter is changed whilst the other parameters are kept constant. This is done for each parameter separately. In a more-at-a-time sensitivity analysis multiple parameters are changed. Although the one-at-a-time type of analysis is easier to apply and requires far less time to run, it has one major limitation. It only captures first order effects and neglects higher order effects.

In Fig. 10 we provide an example that explains this limitation. In this example we have a model with 2 parameters. (A) displays in 3-D how much the output changes when the two parameter values change with a certain percentage from their default value. This would be the output of a more-at-a-time sensitivity analysis. (B) presents the same graph but in 2D. When we apply a one-at-a-time sensitivity analysis for parameter 1, the value for parameter 2 is constant. However, the choice of the value of parameter 2 strongly influences the result of the one-at-a-time sensitivity

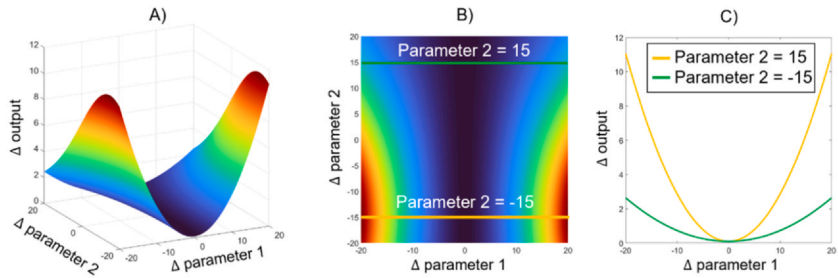

Fig. 10 Example showing the limitation of a one-at-a-time sensitivity analysis. Graph A) displays the sensity of the ouput based on the value of 2 parameters. Graph B) shows a top-down view of graph A). Graph C) displays the sensitivty of the output to changes in parameter 1, given a certain value of parameter 2. The color of the line corresponds to the same line in graph B).

analysis of parameter 1. Graph C) shows this where the sensitivity of parameter 1 for two different values of parameter 2 is plotted. Each line in (C) corresponds to the line of the same color in graph B). If the value of parameter 2 corresponding to the orange line is used, the sensitivity analysis will show that the model is much more sensitive to changes in the value of parameter 1 than would be the case if the value for parameter 2 corresponding to the green line would have been used.

In the example, the sensitivity of the model to changes in both parameters is strongly influenced by the value of the other parameter. This is however not necessarily the case for all parameters in pedestrian models. Therefore, a one-at-a-time method can be applied to determine the sensitivity of a pedestrian model's output to its parameters. However, it is strongly advised to test if the assumption that parameters do not significantly impact each other's sensitivity is valid.

Choosing to perform a one-at-a-time or a more-at-a-time sensitivity analysis is only one of the choices that must be made. There are multiple ways to perform a sensitivity analysis as multiple example in the pedestrian modelling literature (Duives et al., 2016; Hamdar et al., 2022; Sparnaaij et al., 2019) and the closely related traffic flow modelling literature (Punzo et al., 2014; Punzo et al., 2015) show. None of these is the best approach for all cases. It depends on the model, the goal of the sensitivity analysis and the availability of existing information and insight into the sensitivity of the model what method fits best. Regardless of the chosen sensitivity analysis approach, it is important to take the following elements into account.

5.1 The model's sensitivity to a parameter depends on the simulated scenario and metric

A model's sensitivity to a parameter depends on the simulated scenario and metric just like the calibration and validation results for exactly the same reasons. Therefore, it is important to include all relevant scenarios and metrics in the sensitivity analysis. Sections 3.4.1 and 3.4.2 provide guidance to select the relevant scenarios and metrics respectively. When selecting the relevant scenarios and metrics for a sensitivity analysis you do not need to account for data availability as is required for calibration and validation. As long as the model can simulate the scenario and produce the metric they can be included in the analysis. Sections 3.4.1 and 3.4.2 also provide guidance on how to prioritize scenarios and metrics when it is not feasible to include all of them in the analysis.

5.2 Not every parameter value results in realistic behavior

Each parameter will have a range of values within which it produces realistic results. For example, the parameter that determines the strength of the obstacle repulsion force in social force models has a range of values whereby the pedestrian will not walk through a wall nor will never get even close to a wall (i.e. which would both be unrealistic behavior). It is relevant to know which value range of each parameter produces realistic behavior when calibrating a model and performing a sensitivity analysis. In the case of calibration, it is very inefficient to include parameter values that results in unrealistic behavior.

For a sensitivity analysis this is also the case. Furthermore, it is important that the conclusion about a model's sensitivity to parameter is drawn based on the range of values that produce realistic results. For example, in Fig. 11 the model's sensitivity to two parameters is shown. From the figure you

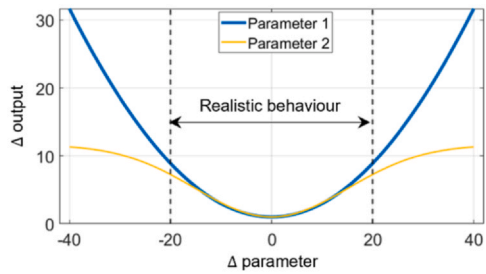

Fig. 11 Example showing the difference between the sensitivity of a large parameter value range versus the value range that results in realistic behavior.

could conclude that the model is much more sensitive to changes to parameter 1 than to changes to parameter 2. However, if parameter 1 only produces realistic results for values between the dotted lines (−20 to 20), this conclusion does not hold. The model's sensitivity to the value of both parameters is comparable in within this range.

Note that if there is no information about the parameter ranges that produce realistic behavior, finding these ranges is the first step of a sensitivity analysis. Here, you can apply an iterative approach whereby you start with a large but coarse value range, check if the parameter values produce realistic behavior and use these results to extent and or refine the value range until a good estimate of the value range is obtained.

Also note that the value range of a parameter that results in realistic behavior can, and often does, depend on the value of other parameters.

5.3 Replications are necessary when the model contains stochastic elements

Like calibration and validation, if a model has stochastic elements, replications are necessary.

A sensitivity analysis is an essential first step of the calibration process with regards to the parameters. However, the same techniques can also be used to investigate to what degree the mode is sensitive to errors or uncertainties in the input. This is valuable information because, as described in Section 3.5.2, errors or uncertainties in the input negatively affect the calibration and validation. And, if a model is sensitive to errors or uncertainties in a particular input, it is particularly important to reduce these errors and uncertainties.

6. Summary

This chapter explains the processes of calibration, validation and verification that together ensure a pedestrian model produces accurate results. The most important aspect of calibration and validation is the choice of scenarios and metrics. You need multiple scenarios and metrics to cover and capture all the relevant behaviors and dynamics. Which behaviors and dynamics are relevant depends on the type of behavior that is modelled (route choice, walking behavior etc.) and on the intended use of the model. Generally, this means that you need multiple different scenarios and multiple metrics.

Another key aspect of calibration and validation is data to which the simulation results are compared. Therefore, the availability of data, or lack thereof, strongly determines what behavior and what contexts can be calibrated and validated. Furthermore, the quality of the data strongly determines the quality of the calibration and validation. The better the quality of the data the better the calibration and validation results.

Another essential aspect is that most pedestrian models are stochastic or use stochastic parameters. Therefore, during calibration and validation it is essential to deal with this stochasticity by means of running replications. And lastly, a sensitivity analysis is a very useful tool to determine which parameters are most important during the calibration ad can help increasing the efficiency of the calibration.

References

Benner, H., Kretz, T., Lohmiller, J., Sukennik, P., 2017. Is calibration a straight-forward task if detailed trajectory data is available? Transportation Research Board 96th Annual Meeting.

Bode, N., 2020. Parameter calibration in crowd simulation models using approximate bayesian computation. Collective Dyn. 5, 340–347. https://collective-dynamics.eu/index.php/cod/article/view/A68.

Campanella, M.C., 2016. Microscopic modelling of walking behaviour [Doctoral dissertation, Delft University of Technology]. Delft University of Technology, Netherlands.

Campanella, M.C., Hoogendoorn, S., Daamen, W., 2009a. Effects of heterogeneity on self-organized pedestrian flows. Transp. Res. Rec.: J. Transp. Res. Board 2124, 148–156.

Campanella, M.C., Hoogendoorn, S., Daamen, W., 2014. Quantitative and qualitative validation procedure for general use of pedestrian models. In: Weidmann, In.U., Kirsch, U., Schreckenberg, M. (Eds.), Pedestrian and Evacuation Dynamics 2012. Springer International Publishing, pp. 891–905.

Campanella, M.C., Hoogendoorn, S.P., Daamen, W., 2009b. Improving the nomad microscopic walker model. IFAC Proc. Vol. 42 (15), 12–18.

Campanella, M.C., Hoogendoorn, S.P., Daamen, W., 2011. A methodology to calibrate pedestrian walker models using multiple-objectives. Pedestrian and Evacuation Dynamics. Springer, pp. 755–759.

Chattaraj, U., Seyfried, A., Chakroborty, P., Biswal, M.K., 2013. Modelling single file pedestrian motion across cultures. Procedia - Soc. Behav. Sci. 104, 698–707. http://www.sciencedirect.com/science/article/pii/S1877042813045552.

Daamen, W., Hoogendoorn, S., 2012. Calibration of pedestrian simulation model for emergency doors by pedestrian type. Transp. Res. Rec.: J. Transp. Res. Board 2316, 69–75.

Dekking, F.M., Kraaikamp, C., Lopuha, H.P., Meester, L.E., 2005. The t-test. A Modern Introduction to Probability and Statistics: Understanding Why and How. Springer, London, pp. 399–413. https://doi.org/10.1007/1-84628-168-7_27.

Department of Defense, 1996. Verification, validation and accreditation (VV&A) Recommended practices guide [Report]. https://acc.dau.mil/adl/en-US/649735/file/71953/Verification,%20validation,%20_%20Accreditation%20(VV_A).pdf.

Duives, D.C., 2016. Analysis and modelling of pedestrian movement dynamics at large-scale events [Doctoral Thesis], http://repository.tudelft.nl/islandora/search/?collection=research.

Duives, D.C., Daamen, W., Hoogendoorn, S.P., 2016. Continuum modelling of pedestrian flows—Part 2: sensitivity analysis featuring crowd movement phenomena. Phys. A: Stat. Mech. Appl 447, 36–48.

Gödel, M., Bode, N., Köster, G., Bungartz, H.-J., 2022. Bayesian inference methods to calibrate crowd dynamics models for safety applications. Saf. Sci. 147, 105586. https://www.sciencedirect.com/science/article/pii/S0925753521004276.

Hamdar, S.H., Talebpour, A., D'Sa, K., Knoop, V., Daamen, W., Treiber, M., 2022. Behavioral-based pedestrian modeling approach: formulation, sensitivity analysis, and calibration. Transp. Res. Rec. 2676 (4), 334–347. https://doi.org/10.1177/03611981211059767.

Hannun, J., Dias, C., Taha, A.H., Almutairi, A., Alhajyaseen, W., Sarvi, M., Al-Bosta, S., 2022. Pedestrian flow characteristics through different angled bends: exploring the spatial variation of velocity. PLOS ONE 17 (3), 1–21. https://doi.org/10.1371/journal.pone.0264635.

Hänseler, F.S., Lam, W.H.K., Bierlaire, M., Lederrey, G., Nikolić, M., 2017. A dynamic network loading model for anisotropic and congested pedestrian flows. Transp. Res. Part. B: Methodol 95, 149–168. http://www.sciencedirect.com/science/article/pii/S0191261515301442.

Helbing, D., Molnár, P., 1997. Self-organization phenomena in pedestrian crowds. Self-Organization of Complex Structures: From Individual to Collective Dynamics. Gordon and Breach Science Publisher, Amsterdam.

Hyndman, R.J., Athanasopoulos, G., 2018. The forecaster's toolbox. Forecasting: Principles And Practice. OTexts.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. Int. J. Forecast 22 (4), 679–688. https://www.sciencedirect.com/science/article/pii/S0169207006000239.

Institute for Advanced Simulation 7: Civil Safety Research of Forschungszentrum Julich, 2025. Pedestrian Dynamics Data Archive https://doi.org/10.34735/ped.da.

International Maritime Organisation, 2016. Revised guidelines for evacuation analysis for new and existing passenger ships [Report](MSC.1/Circ.1533). https://puc.overheid.nl/nsi/doc/PUC_642556_14/1/.

International Organization for Standardization, 2020. Verification and validation protocol for building fire evacuation models [techreport](ISO 20414:2020).

Kochenderfer, M.J., Wheeler, T.A., 2019. Algorithms for Optimization. The MIT Press, Cambridge.

Kolassa, S., Siemsen, E., 2014. PART IV: forecasting quality. Demand Forecasting for ManagersBusiness Expert Press. http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=4648346.

Li, K., Wang, R., Zhang, T., Ishibuchi, H., 2018. Evolutionary many-objective optimization: a comparative study of the state-of-the-art. IEEE Access 6, 26194–26214.

Massey Frank, J., 1951. The Kolmogorov-Smirnov test for goodness of fit. J. Am. Stat. Assoc 46 (253), 68–78. https://www.tandfonline.com/doi/abs/10.1080/01621459.1951.10500769.

Punzo, V., Ciuffo, B., Montanino, M., Daamen, W., Buisson, C., Hoogendoorn, S.P., 2014. Sensitivity analysis. Traffic Simulation and Data: Validation Methods and Applications. CRC Press, Boca Raton, FL, USA, pp. 119.

Punzo, V., Montanino, M., Ciuffo, B., 2015. Do we really need to calibrate all the parameters? Variance-based sensitivity analysis to simplify microscopic traffic flow models. IEEE Trans. Intell. Transp. Syst 16 (1), 184–193.

RiMEA e.V., 2022. RiMEA e.V. Richtlinie fur Mikroskopische Entfluchtungs-Analysen [techreport]. http://www.rimea.de/de/.

Ronchi, E., Kuligowski, E.D., Reneke, P.A., Peacock, R.D., Nilsson, D., 2013. The process of verification and validation of building fire evacuation models (Technical Note (NIST TN) - 1822). https://doi.org/10.6028/NIST.TN.1822.

Schlesinger, S., 1979. Terminology for model credibility. SIMULATION 32 (3), 103–104. https://doi.org/10.1177/003754977903200304.

Sparnaaij, M., Duives, D.C., Knoop, V.L., Hoogendoorn, S.P., 2019. Multiobjective calibration framework for pedestrian simulation models: a study on the effect of movement base cases, metrics, and density levels. J. Adv. Transp. 2019, 18.

Wolinski, D., Guy, J.S., Olivier, A.-H., Lin, M., Manocha, D., Pettre, J., 2014. Parameter estimation and comparative evaluation of crowd simulations. Computer Graph. Forum 33 (2), 303–312.

Wu, Y., 2019. Testing the International Standards Organization verification and validation protocol for evacuation simulations-an application to the FDS+ Evac model Lund University.

Zeng, W., Chen, P., Yu, G., Wang, Y., 2017. Specification and calibration of a microscopic model for pedestrian dynamic simulation at signalized intersections: a hybrid approach. Transp. Res. Part. C: Emerg. Technol 80, 37–70. https://www.sciencedirect.com/science/article/pii/S0968090×17301213.

Zhong, J., Cai, W., 2015. Differential evolution with sensitivity analysis and the Powell's method for crowd model calibration. J. Comput. Sci. 9, 26–32. http://www.sciencedirect.com/science/article/pii/S1877750315000514.