MSc thesis in Geomatics

From IFC BIM to Semantically Enriched 2.5D Indoor Navigation Graphs for Congestion-Aware Pedestrian Simulation

Xu Wang

June 2026

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors: Ken Arroyo Ohori
Martijn Meijers

Abstract

Indoor navigation in complex public buildings requires more than geometric shortest-path routing. In multi-level station environments, pedestrian movement is shaped by vertical connectors, access-control elements, semantic origin–destination regions, congestion, and heterogeneous walking behaviour. Although IFC BIM provides rich geometric and semantic information about building elements, it is not directly usable as a navigation or pedestrian simulation model.

This thesis develops and evaluates a reproducible workflow for transforming IFC BIM data into a semantically enriched 2.5D indoor navigation graph for congestion-aware pedestrian routing and simulation. The proposed workflow interprets movement-relevant IFC semantics, extracts floor-based walkable layers, samples graph nodes, filters obstacles and restricted areas, models stairs, escalators, elevators, gates, and platform transitions as semantic graph elements, and uses the resulting graph as the substrate for routing and agent-based simulation.

The workflow is first checked on a simplified two-floor pilot model and then applied to Dadongmen Station of Hefei Metro Line 1. The final default station graph contains 18,454 nodes and 134,852 edges across the public Platform, Concourse, and Transport levels. A sparse semantic-light graph variant is also generated to test the effect of graph abstraction. Six experiments evaluate static routing, congestion-aware replanning, pedestrian-profile effects, mixed-agent behaviour, algorithm-threshold sensitivity, and demand-load response.

The results show that the default graph supports station-scale routing and controlled 600 s pedestrian simulation. Static routing exposes connector bottlenecks, while congestion-aware replanning reduces waiting time and peak queue length but can increase mean and high-percentile travel time. Elderly agents experience substantially higher travel and waiting burdens than normal agents. The sparse graph remains routable, but produces higher waiting, larger queues, more rerouting events, and earlier demand-load degradation. These findings indicate that graph connectivity alone is not sufficient; indoor navigation graphs should also be evaluated through movement-performance metrics.

The thesis demonstrates that IFC-derived station information can be transformed into a semantically meaningful 2.5D movement substrate for routing, simulation, congestion-aware replanning, and graph-design evaluation. The results should be interpreted as controlled methodological validation rather than calibrated operational passenger-flow prediction.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Ken Arroyo Ohori and Martijn Meijers, for their guidance, feedback, and patience throughout this thesis. Their comments helped me clarify the research questions, improve the methodological depth, and strengthen the interpretation of the experimental results.

I am also grateful to the 3D Geoinformation group at TU Delft for providing a rigorous academic environment in which this work could be developed. I would like to thank those who supported me during the thesis process, including classmates, friends, and people who discussed technical or practical issues with me.

Finally, I would like to thank my parents, for their continuous support during my studies.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

ABM	Agent-based model
AFC	Automatic fare collection
BIM	Building Information Modeling
CSV	Comma-separated values
IFC	Industry Foundation Classes
LOD	Level of detail
OD	Origin–destination
OGC	Open Geospatial Consortium
RQ	Research question

1. Introduction

1.1. Background and Motivation

Indoor navigation has become increasingly important in large public buildings such as transport hubs, airports, hospitals, and commercial complexes. Compared with outdoor navigation, indoor movement takes place in enclosed and highly structured environments, where circulation depends on doors, corridors, stairs, escalators, gates, and floor-to-floor transitions. In these settings, movement is shaped not only by geometry, but also by semantics, access conditions, and local operational constraints [Isikdag et al., 2013; Open Geospatial Consortium, 2020].

This challenge becomes more pronounced in multistorey and high-density environments. In a station, pedestrians do not simply move along a fixed shortest path. Their movement is influenced by vertical circulation, bottlenecks, directional restrictions, and temporary congestion. A route that is geometrically short may become inefficient in practice if queues build up at stairs, escalators, or gates. For this reason, indoor navigation models need to support not only connectivity, but also movement analysis under realistic circulation conditions [Liu et al., 2021b; Tan et al., 2025].

Indoor navigation does not have to start from BIM alone. Alternative representations can be derived from floor plans, point clouds, trajectories, or localization-oriented systems, each with its own advantages and limitations [Han et al., 2023; Fu et al., 2021; Nikoohemat et al., 2020; Khan et al., 2022]. Floor plans are widely available but usually provide weaker semantics and limited vertical structure. Point clouds can capture detailed as-is geometry, but they often require substantial reconstruction and semantic interpretation before a usable navigation model can be obtained. Trajectory- and localization-based approaches are valuable for operational guidance, but they typically depend on movement traces, sensing infrastructure, or real-time positioning support.

Building Information Modeling (BIM), and especially the Industry Foundation Classes (IFC), is used as the starting point of this thesis because it provides structured information about indoor geometry, building elements, storeys, spaces, and circulation components [Liu et al., 2021a; Kang and Li, 2017]. In a station environment, such information is relevant because pedestrian movement depends on vertical connectors, access-controlled transitions, and semantically distinct circulation elements. However, this does not mean that BIM is assumed to be the only or universally best source for indoor navigation. Rather, the thesis investigates how IFC BIM, when available, can be transformed into a movement-oriented graph representation for routing and simulation.

However, IFC is not directly a navigation or simulation model. It is a construction-oriented data model, and before it can be used for pedestrian routing or congestion-aware simulation, it must be transformed into a navigable representation that preserves movement-relevant structure while reducing unnecessary geometric detail.

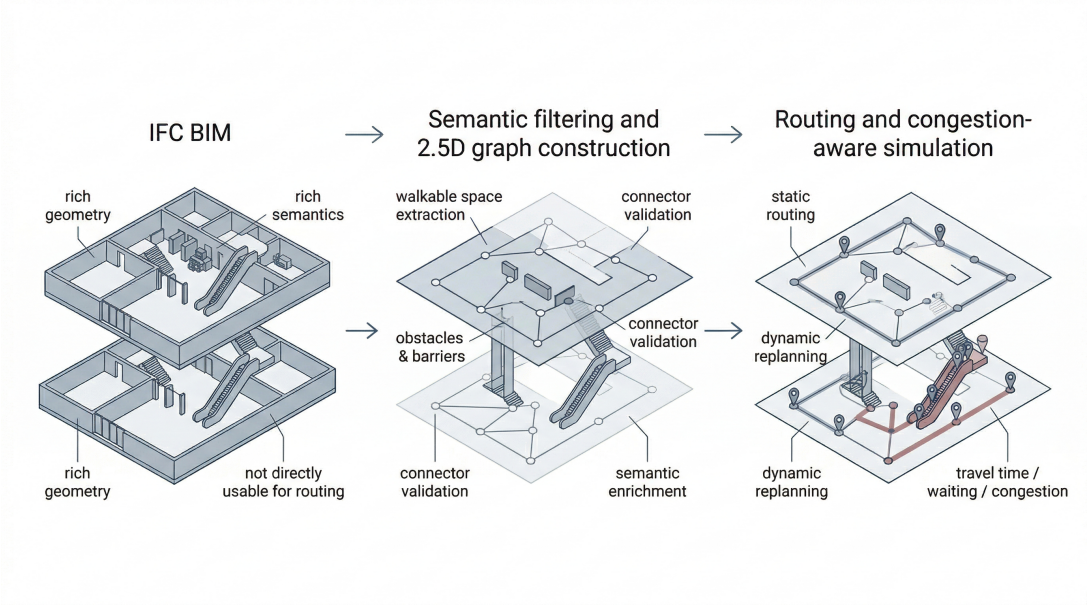


Figure 1.1.: From construction-oriented BIM to simulation-ready indoor navigation

Against this background, this thesis studies how IFC BIM can be converted into a semantically enriched 2.5D indoor navigation graph for routing and congestion-aware pedestrian simulation in a complex station environment (see figure 1.1).

1.2. Problem Statement

Although IFC provides rich geometric and semantic building information, it is not immediately suitable for pedestrian routing or simulation. Raw building models contain many elements that are irrelevant or ambiguous from a navigation perspective, such as technical spaces, overhead components, facade elements, and objects that do not directly affect pedestrian movement. At the same time, movement-relevant structures such as stairs, escalators, gates, and floor transitions often require explicit interpretation before they can be represented in a navigation graph.

This creates two closely related problems. The first is a modelling problem: how to transform complex IFC data into a graph that represents walkable indoor space in a usable and consistent way. The second is an analysis and design problem: once such a graph has been built, how to develop it into a representation that supports not only route computation, but also plausible movement simulation under congestion and heterogeneous circulation conditions, and how to evaluate its performance in these tasks.

These problems are especially relevant in multi-level station environments, where routing is shaped by vertical connectors, local bottlenecks, and fluctuating pedestrian density. A useful graph representation must therefore do more than encode adjacency. It must also capture movement-relevant semantics, support dynamic route adaptation when local conditions change, and remain suitable for controlled routing and simulation analysis.

Accordingly, this thesis does not treat graph construction as an end in itself. Instead, it studies how IFC BIM can be transformed into a semantically enriched navigation graph that serves as an operational intermediate representation for routing, congestion-aware replanning, and pedestrian movement analysis (see table 1.1).

Table 1.1.: Conventional indoor routing versus the focus of this thesis.

Aspect	Conventional indoor routing	This thesis
Input data	Floor plans, simplified layouts, or pre-built networks	Multi-file IFC BIM with detailed geometry and semantics
Representation	Static topological or geometric route network	Semantically enriched 2.5D indoor navigation graph
Focus	Shortest-path guidance	Routing plus congestion-aware pedestrian movement analysis
Vertical circulation	Often simplified	Explicitly modelled through stairs, escalators, gates, and floor transitions
Dynamics	Usually static cost model	Local congestion can trigger route adaptation
Evaluation	Route correctness or usability	Travel time, waiting, congestion, and route change under simulation

1.3. Research Aim and Questions

The objective of this thesis is to develop and evaluate a reproducible workflow that transforms IFC BIM data into a semantically enriched 2.5D indoor navigation graph and uses this graph for pedestrian routing, congestion-aware simulation, and movement-performance analysis in a multi-level station environment.

More specifically, the thesis aims to:

- transform multi-file IFC BIM into a traffic-relevant and semantically enriched 2.5D indoor navigation graph;
- use the generated graph as a substrate for routing and graph-based pedestrian simulation;
- implement congestion-aware replanning on top of the generated graph; and
- evaluate how graph design, routing strategy, pedestrian profile, and demand level affect travel time, waiting, queueing, and route adaptation.

This thesis is guided by the following main research question:

> How can IFC BIM be transformed into a semantically enriched 2.5D indoor navigation graph and used to support congestion-aware pedestrian routing and simulation in a multi-level station environment?

The supporting sub-questions are:

1. How can multi-file IFC BIM data be transformed into a traffic-relevant and semantically meaningful 2.5D indoor navigation graph?
2. How do graph construction and semantic modelling choices affect the navigability and movement-performance behaviour of the resulting graph?
3. To what extent does congestion-aware replanning change pedestrian movement performance compared with fixed-routing simulation on the generated graph?
4. How do heterogeneous pedestrian profiles influence travel time, waiting, and congestion patterns in the station environment?

Figure 1.2 summarizes how the main research question is decomposed into sub-questions and how these sub-questions are linked to the methodological and experimental evidence used in the thesis.

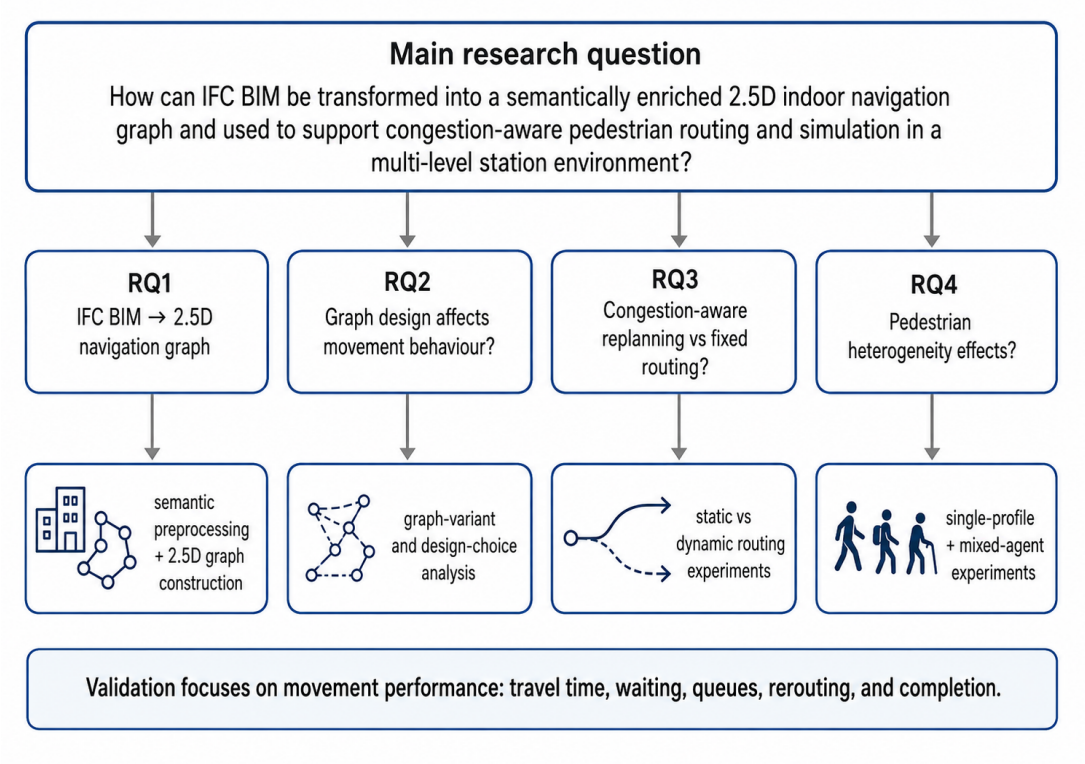


Figure 1.2.: Research questions, methodological evidence, and validation logic of the thesis.

1.4. Contributions

The main contributions of this thesis are:

- a reproducible workflow for transforming IFC BIM data into a traffic-relevant indoor navigation representation;

- a semantically enriched 2.5D graph model that represents walkable space, vertical connectors, and movement-related constraints;
- a graph-based simulation setup for testing static routing, congestion-aware replanning, and heterogeneous pedestrian profiles; and
- an experimental evaluation of how graph modelling, pedestrian heterogeneity, congestion-aware rerouting, and demand thresholds affect movement performance, queue formation, and simulation robustness.

1.5. Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 reviews related work on BIM-based and non-BIM indoor navigation, indoor graph abstraction, walkable-space extraction, graph-based pedestrian simulation, and congestion-aware replanning. Chapter 3 presents the methodology, including the IFC-to-2.5D graph workflow, graph-based simulation logic, congestion-aware replanning, algorithms, and implementation tools. Chapter 4 introduces the simplified pilot case used to check the internal correctness of the workflow before applying it to the full station case. Chapter 5 defines the experimental design and validation strategy, including the common simulation setup, evaluation metrics, and experiments used to test static routing, congestion-aware replanning, pedestrian heterogeneity, algorithm-threshold sensitivity, and building-capacity response. Chapter 6 presents and discusses the results, including the generated 2.5D navigation graph and the outcomes of the experimental comparison. Chapter 7 concludes the thesis, answers the research questions, summarizes the main contributions and limitations, and outlines future work. Appendix A provides the reproducibility self-assessment.

2. Related Work

2.1. Indoor Navigation Data Sources

Indoor navigation models can be derived from different types of spatial data. This section first reviews BIM- and IFC-based approaches, which are central to this thesis, and then compares them with alternative representations such as floor plans, trajectories, point clouds, and vision- or localization-based systems.

2.1.1. BIM- and IFC-based Indoor Navigation

Building Information Modeling (BIM), and particularly the Industry Foundation Classes (IFC), has become an important data source for indoor navigation research because it combines detailed building geometry with rich semantic information about spaces, openings, circulation elements, and functional components. Compared with conventional geometric floor plans, IFC provides explicit object-level descriptions of architectural elements such as walls, doors, stairs, and spaces, which makes it attractive for deriving navigation structures directly from design or as-built building models [Liu et al., 2021a; Isikdag et al., 2013].

Early work in this area emphasized that BIM is not only a geometric model but also a semantic model that can support navigation-oriented reasoning. A representative example is the BIM-oriented model proposed by Isikdag et al. [2013], who argued that indoor navigation requires more than metric geometry and should incorporate semantically meaningful entities such as transitions, access constraints, and user-relevant spatial interpretations. This perspective remains highly relevant: for indoor routing, a wall is not merely a polygonal boundary, and a stair is not merely a 3D solid; both must be interpreted according to their navigational role.

Subsequent studies increasingly focused on transforming IFC into graph-based or space-based navigation representations. One major line of work converts IFC building elements into IndoorGML-compatible structures, thereby shifting from construction-oriented object modeling to navigation-oriented spatial abstraction (see figure B.1). IndoorGML was explicitly designed by OGC to represent indoor space for navigation using geometric, topological, and semantic relationships rather than detailed architectural components alone [Open Geospatial Consortium, 2020; Kang and Li, 2017]. In this context, IFC provides the source semantics, while IndoorGML or graph models provide the navigable abstraction. Tools such as *IFC2IndoorGML* demonstrate that IFC-to-IndoorGML conversion is feasible in practice and can automate part of this transformation pipeline [Diakite et al., 2022].

Another line of research aims to perform path planning directly on simplified representations derived from BIM. For example, Liu et al. [2021a] reviewed IFC-supported indoor navigation studies and showed that the literature spans route planning, evacuation, accessibility analysis, and emergency response, but also noted that many approaches remain tightly

coupled to specific applications or building assumptions. Similarly, lightweight BIM-based path planning approaches have attempted to reduce the computational burden of operating on highly detailed BIM models by deriving simplified network structures for routing [Liu et al., 2021c]. These studies highlight a recurring trade-off: the richer the BIM semantics and geometry, the higher the potential realism, but the greater the need for abstraction before routing or simulation becomes computationally manageable.

Despite this progress, several limitations remain common in IFC-based indoor navigation research. First, many studies stop at connectivity extraction, IndoorGML conversion, or static route computation, and therefore do not fully address how the resulting graph can support later movement analysis under congestion or heterogeneous circulation conditions [Liu et al., 2021a]. Second, the semantic richness of IFC does not automatically translate into navigable space. In practice, substantial preprocessing is required to distinguish public circulation from technical rooms, overhead objects, facade components, and other elements that are geometrically present but irrelevant to pedestrian movement. Third, vertical circulation elements such as stairs, escalators, elevators, and gates are often more difficult to encode than ordinary horizontal walkable space because they introduce directionality, accessibility constraints, and potential bottleneck effects.

These limitations help explain both the research gap and the rationale for the present thesis. For a complex multi-level station environment, BIM is a particularly useful starting point because it provides explicit representations of storeys, spaces, stairs, escalators, and other circulation elements that are central to pedestrian movement. Compared with representations that must infer such structure from geometry or observations alone, BIM makes it possible to build a semantically controlled navigation model earlier and more directly. However, this advantage is only realized if the BIM is systematically filtered and transformed into a graph representation that is suitable for movement analysis. The contribution of this thesis is therefore not merely to derive connectivity from IFC, but to develop a reproducible BIM-to-graph workflow in which semantic preprocessing, graph construction, and later congestion-aware movement analysis are explicitly linked.

2.1.2. Alternative Indoor Navigation Representations

Indoor navigation does not depend exclusively on BIM or IFC. In many practical settings, detailed building information models are unavailable, incomplete, outdated, or difficult to access. As a result, a substantial body of research has explored alternative indoor navigation representations derived from floor plans, topological structures, trajectories, point clouds, and sensor-driven systems. These approaches demonstrate that indoor navigation can be supported by multiple forms of spatial abstraction, each with different strengths in terms of geometric fidelity, semantic richness, update cost, and computational efficiency (see figure 2.2).

One important family of methods derives navigation structures from architectural floor plans rather than BIM. In such approaches, the main task is usually to recover navigable space and topological relations from 2D geometric drawings that contain walls, doors, corridors, and room boundaries, but lack explicit spatial semantics and connectivity descriptions. For example, Han et al. [2023] proposed a navigation-oriented topological model construction algorithm for complex indoor space based on building plans, with the goal of recovering polygonal space entities and their topological relationships. Their work highlights a practical reality: many indoor environments still rely on floor plans as the primary available

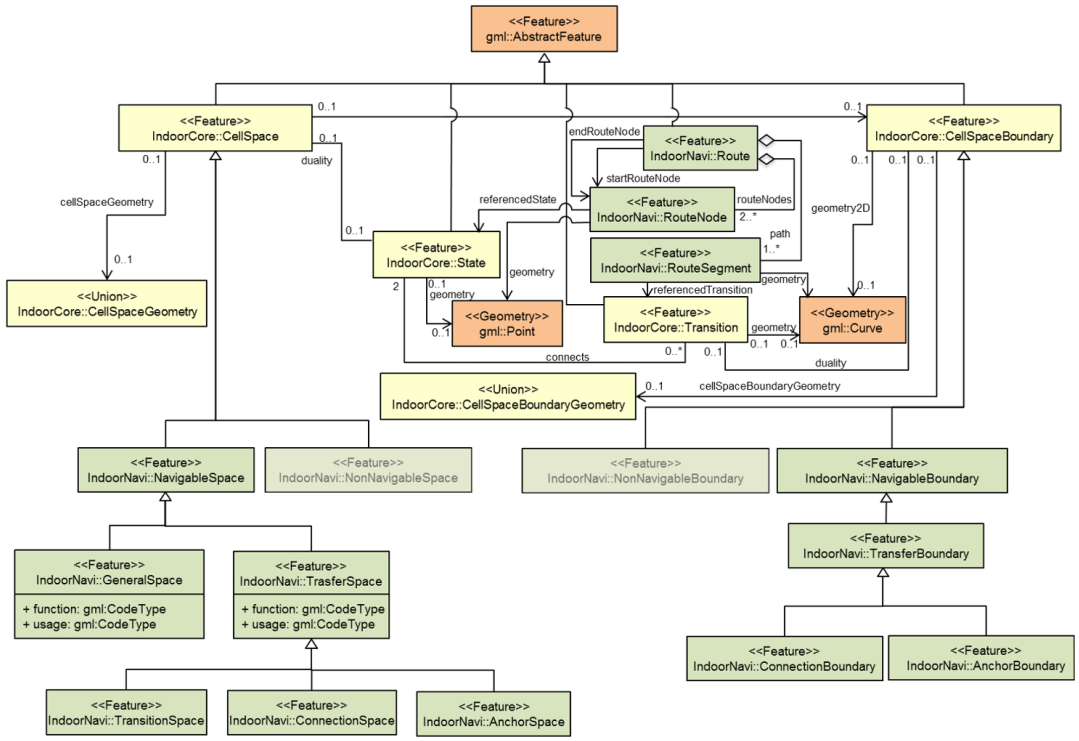


Figure 2.1.: UML diagram of IndoorGML Navigation module [Open Geospatial Consortium, 2020]

source, and navigation models must therefore be reconstructed from geometry and inferred adjacency rather than directly extracted from semantically structured building models.

A related line of work focuses on generating route networks from geometric floor-plan structures using corridor axes or skeleton-like representations. Lewandowicz and Lisowski [2019] proposed a modified methodology for generating indoor navigation models based on floor plans and medial-axis-inspired transformations. The appeal of this type of representation lies in its simplicity: instead of preserving all geometric detail, the navigable environment is reduced to route-like centerlines or simplified corridor structures that are directly suitable for path computation. However, such representations often work best in corridor-dominant layouts and may require additional processing to handle wider open areas, irregular spaces, or complex vertical circulation elements.

Beyond design documents, another class of non-BIM methods reconstructs indoor navigation graphs from movement data. Fu et al. [2021] presented an automated approach for constructing indoor 3D navigation graphs from large-scale low-frequency indoor trajectories without relying on auxiliary building data. Their workflow includes trajectory simplification, floor-plan extraction, and inter-floor topological connection detection. Trajectory-driven approaches are attractive when large volumes of positioning traces are available, because they can capture how people actually move through space rather than how a building was originally designed. At the same time, the resulting graphs are highly dependent on trajectory coverage, positioning error, and sampling density, which can limit completeness in less frequently visited areas or at constrained vertical connectors.

Indoor navigation representations can also be derived from point clouds and reconstructed building geometry. Nikoohemat et al. [2020] introduced a workflow that reconstructs 3D building models from indoor point clouds and extracts fine-grained navigation networks for path planning. Their method additionally reconstructs obstacles and stairs, enabling multistorey routing based on as-observed spatial structure rather than design semantics. Point-cloud-based workflows are valuable in scan-rich contexts and for existing buildings without reliable BIM, but they usually require substantial geometric reconstruction before a usable navigation structure can be obtained. Compared with BIM-derived methods, they often begin with richer as-is geometry but weaker native semantics.

A broader stream of research addresses indoor navigation from the perspective of localization and user guidance systems rather than explicit building information models. Vision-based indoor navigation is one clear example. Khan et al. [2022] reviewed the state of the art in vision-based indoor navigation and showed that indoor navigation systems may also be organized around image recognition, camera-based localization, marker-based guidance, or wearable/mobile-device feedback rather than pre-existing architectural models. From this perspective, the navigation representation is not always a conventional building graph extracted from spatial data; instead, it may be embedded in localization pipelines, visual landmarks, or sensor fusion frameworks. Such systems are highly relevant for real-world navigation services, but they address a somewhat different problem from the one studied in this thesis, which focuses on deriving a reusable navigation graph directly from building-scale spatial data (see table 2.1).

Taken together, these non-BIM approaches show that indoor navigation can be supported by multiple representational paradigms. Floor-plan-based methods are widely applicable when only drawings are available; trajectory-based methods are data-driven and behavior-sensitive; point-cloud-based methods are suitable for as-is reconstruction; and vision-based

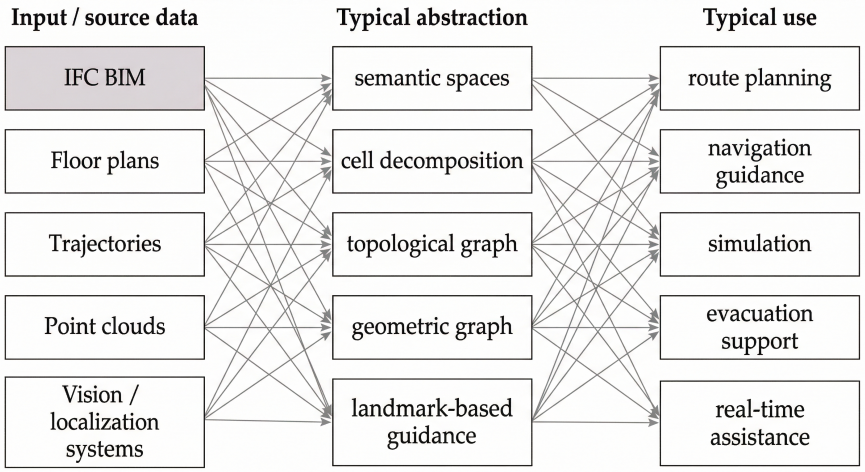


Figure 2.2.: Indoor navigation representations across data sources and abstraction levels

methods are closely tied to real-time localization and user interaction. However, these alternatives also clarify the specific niche of the present thesis. This work is concerned with the transformation of semantically rich IFC BIM into a simulation-ready 2.5D graph representation. Therefore, the goal is not to argue that BIM-based approaches are universally superior, but to show that, for a complex multistorey station environment with heterogeneous circulation elements and the need for graph-based pedestrian simulation, IFC provides a particularly strong basis for building a semantically enriched navigation model.

2.2. Navigation Graph Modelling

After a suitable data source has been selected, the next challenge is to transform indoor space into a representation that is usable for routing and movement analysis. This section reviews graph abstraction strategies, semantic modelling choices, and the extraction of traversable space from building geometry.

2.2.1. Indoor Graph Abstraction and Semantic Modelling

Most indoor navigation methods eventually rely on some form of graph abstraction (see table 2.2). The reason is practical: raw building geometry is too detailed for efficient routing, while purely geometric models often do not encode the relations that navigation actually needs. A graph offers a compact structure in which nodes and edges can represent spaces, transitions, and movement constraints in a form suitable for path computation and simulation [Open Geospatial Consortium, 2020; Liu et al., 2021a].

The main design question is not whether to use a graph, but how much of the indoor environment should be preserved in that graph. Some models are strongly topological and represent rooms, corridors, and doors as abstract connections. These models are efficient, but they may oversimplify movement in large open areas or around complex circulation elements. Other models are more geometric and retain visibility, orientation, or local route

Table 2.1.: Typical characteristics of indoor navigation inputs discussed in the literature.

Input type	Main available information	Typical additional processing	Common limitation for navigation modelling	Example references
IFC BIM	Building geometry and object semantics	Semantic filtering and graph abstraction	Construction-oriented structure is not directly navigable	[Liu et al., 2021a; Isikdag et al., 2013; Diakite et al., 2022]
Floor plans	2D layout geometry and room boundaries	Connectivity inference and transition detection	Limited native semantics and weak vertical representation	[Han et al., 2023; Lewandowicz and Lisowski, 2019]
Trajectories	Observed movement traces and implicit route usage	Coverage analysis and graph reconstruction	Incomplete spatial coverage and dependence on trajectory density	[Fu et al., 2021]
Point clouds	As-is 3D geometry	Reconstruction, segmentation, and obstacle interpretation	Weak native semantics and heavy preprocessing	[Nikoohemat et al., 2020]
Vision / localization systems	Landmarks, images, and user position cues	Localization, scene interpretation, and guidance logic	Often focused on assistance rather than reusable spatial graph construction	[Khan et al., 2022]

structure. For example, Zhou et al. [2022] proposed a hierarchical indoor visibility-based graph to improve route guidance in multi-storey buildings, partly to address the limited geometric precision of simpler topological models.

IndoorGML formalizes this idea by representing indoor environments through cells, boundaries, and their topological relations, while also allowing semantic interpretation and multi-layer modelling [Open Geospatial Consortium, 2020]. This is important because navigation is rarely defined by adjacency alone. In practice, the same physical connection may have different meanings depending on use conditions: a stair and an escalator both connect floors, but they do not impose the same movement cost, directionality, or accessibility constraints. Graph semantics therefore matter as much as graph topology (see figure 2.3).

Recent work has made this point more explicit. Hamzei et al. [2024] argued that indoor graph models often separate route computation from richer spatial understanding, and proposed a model that captures both route and configurational information. In a similar vein, semantics-based network construction methods emphasize that a navigable indoor graph should encode not only connectivity, but also the functional role of spaces and transitions [Zhu et al., 2025]. For navigation tasks beyond static shortest paths, this becomes even more important.

For this thesis, graph abstraction is treated as a modelling step rather than a neutral conversion step (see figure 2.4). The aim is not simply to extract connectivity from IFC, but to construct a graph that remains close enough to pedestrian movement conditions while still being computationally manageable. This is why the graph must carry semantic information about walkable areas, vertical connectors, gates, directionality, and other movement-relevant constraints. In other words, the graph is not only a routing structure; it is also the basis for later pedestrian simulation.

Table 2.2.: Typical graph abstraction strategies in indoor navigation research.

Representation	What is typically represented	Typical advantage	Typical limitation	Example references
Room-level topological graph	Rooms or spaces as nodes, transitions as edges	Compact structure for route search	Coarse treatment of local circulation and open areas	[Open Geospatial Consortium, 2020; Kang and Li, 2017]
Visibility / geometric graph	Geometric route structure, visibility, or local line-of-sight relations	Better support for local route realism	More complex construction and higher geometric sensitivity	[Zhou et al., 2022; Hamzei et al., 2024]
Grid / raster model	Regular cells or voxels representing traversable space	Simple movement discretization and local update logic	High computational cost at fine resolution	[Xie et al., 2024]
Semantically enriched 2.5D graph	Walkable floor structure with explicit connectors and movement semantics	Balances route structure, semantics, and simulation usability	Requires explicit modelling decisions during preprocessing	[Zhu et al., 2025; Liu et al., 2021a]

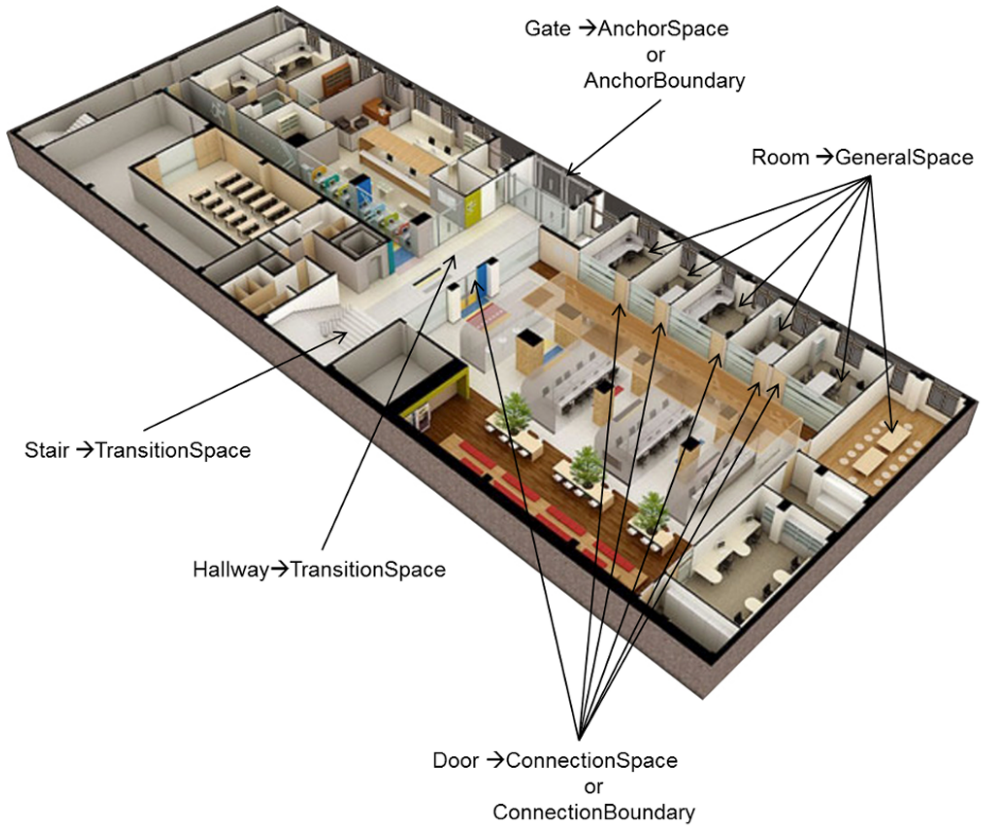


Figure 2.3.: Indoor space mapped to IndoorGML Navigation module classes [Open Geospatial Consortium, 2020]

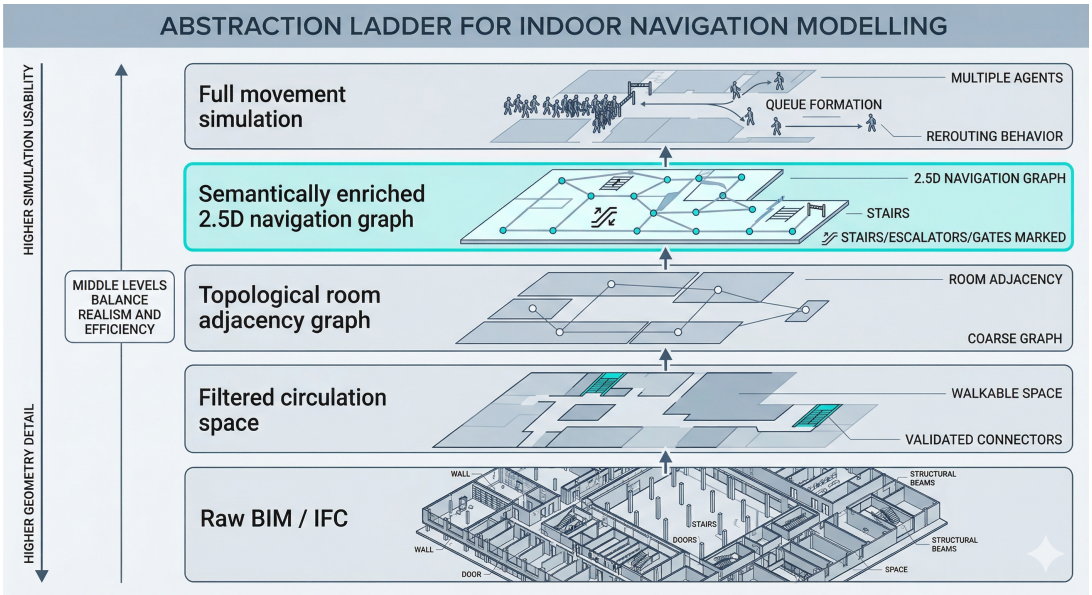


Figure 2.4.: Abstraction ladder from building model to simulation-ready graph

2.2.2. Traversable Space Extraction and Walkability Constraints

A navigation graph is only useful if it represents space that is actually traversable. For this reason, indoor navigation research often includes an explicit step that converts building geometry into walkable or free space. This step is not trivial. A building model contains many elements that define space geometrically but should not become part of the navigation network, such as façade skins, overhead components, technical rooms, or furniture-like obstacles. The key task is therefore not only to detect connectivity, but to distinguish navigable space from non-navigable space.

Several studies have approached this problem through free-space extraction. Diakite and Zlatanova [2016] argued that navigation should be based on the extraction of free space from BIM rather than on the direct use of building elements alone. Their work shows that walkable space is better understood as the residual space left after relevant obstacles and enclosing structures have been considered. This idea is important because it shifts the modelling focus from objects to movement.

A related issue is that walkability depends on human use conditions rather than geometry alone. Zhen et al. [2020] proposed a human-oriented method for generating comfortable navigable space by considering user dimensions and interactions with indoor objects. Their work makes a useful distinction: a path may be geometrically possible but still too narrow or inconvenient for realistic movement. In other words, traversability is constrained by clearance, not just by topological reachability.

Obstacle treatment is therefore central to indoor navigation modelling. Xu et al. [2017] showed that indoor path planning can change substantially when obstacles are explicitly considered instead of relying on simplified open-floor assumptions. This matters particularly in complex interiors, where local objects and partitions influence route availability

and detour cost. Similar observations appear in point-cloud-based reconstruction workflows, where obstacle detection is necessary before a navigation model can be derived from scanned geometry [Nikoohemat et al., 2020].

These studies are directly relevant to this thesis. The graph used here is not generated from all IFC elements equally. It is built from a filtered representation of public circulation space, together with movement-relevant barriers and connectors. This means that walkability must be defined through several constraints: whether a storey or subspace is publicly accessible, whether a building element acts as a barrier at pedestrian level, whether sufficient clearance is available, and whether doors, stairs, escalators, or gates provide valid transitions. Traversable space extraction is therefore a core modelling step in the pipeline, not a minor preprocessing detail.

2.3. Pedestrian Movement Analysis

Once a navigation graph has been constructed, it can be used not only for shortest-path routing, but also for graph-based pedestrian simulation and adaptive routing under changing local conditions. This section reviews work on pedestrian routing, simulation, and congestion-aware replanning.

2.3.1. Pedestrian Routing and Graph-based Simulation

After a navigation graph is constructed, it can be used in two related but different ways. In the simpler case, the graph supports route computation between an origin and a destination. In the broader case, it becomes the movement structure for pedestrian simulation, where multiple agents move simultaneously, compete for space, and generate local congestion. In this setting, the quality of the graph is not judged only by connectivity, but also by whether it supports plausible movement patterns under load (see algorithm 2.1).

Recent work shows a clear shift from static indoor routing toward more dynamic and simulation-oriented applications. For example, Valizadeh et al. [2024] developed an end-to-end indoor navigation framework based on BIM, GIS, and augmented reality, where a graph-based route network is generated from the building model for pedestrian guidance. Their work still focuses mainly on route generation and guidance, but it reflects the broader trend that indoor graphs are increasingly treated as reusable operational structures rather than one-off geometric outputs. Similarly, Zhang et al. [2025] proposed a BIM- and AR-based indoor navigation system for pedestrians and argued that semantic information is important for practical route planning and instruction generation.

In parallel, recent evacuation and safety studies have increasingly coupled graph-based routing with dynamic building conditions. Wong and Lee [2023] presented an indoor navigation and information-sharing system for collaborative fire emergency response, where route guidance is linked to changing conditions and participant roles. Tan et al. [2025] proposed a digital-twin-enabled evacuation guidance framework that updates evacuation routes in real time according to the current hazard state. These studies are not identical to the present thesis, but they show that indoor routing is moving beyond fixed shortest-path analysis toward adaptive route support under evolving conditions.

Recent simulation work points in the same direction. Senanayake et al. [2024] reviewed pedestrian evacuation studies from 2013 to 2023 and showed that agent-based models are now widely used to represent heterogeneous behaviour, local interactions, and changing environments. At the same time, newer modelling frameworks increasingly connect routing, sensing, and digital representations of buildings. For example, Xie et al. [2024] proposed a voxel-based 3D indoor model for evacuation simulation that explicitly represents navigable surfaces and vertical connectivity, while Bhardwaj et al. [2024] introduced a 3D indoor navigation framework for fire evacuation dynamics that combines dynamic hazards with indoor route planning.

For this thesis, the main point is more limited. The graph is not used only to compute one shortest route, but also to support repeated movement simulation under different scenarios. Agents are assigned origins and destinations, move on the graph in discrete time, and may experience waiting or rerouting when local congestion emerges. This remains a simplified form of pedestrian simulation, but it is sufficient for the purpose of this work: to test whether a semantically enriched 2.5D graph derived from IFC can support both routing and congestion-aware movement analysis in a complex station environment.

Algorithm 2.1: Conceptual workflow for constructing a simulation-ready indoor graph from building data

Input: Indoor building data (e.g. IFC BIM, floor plan, or reconstructed geometry)
Output: Navigation graph suitable for routing and movement analysis

1 Identify walkable and non-walkable space;
2 Detect transitions such as doors, stairs, escalators, and gates;
3 Filter irrelevant or inaccessible elements;
4 Construct a navigation graph at a chosen abstraction level;
5 Attach movement-relevant semantics to nodes and edges;
6 Use the graph for route computation or simulation under scenario-specific conditions;

2.3.2. Congestion-aware Routing and Dynamic Replanning

Static indoor routing assumes that the best path can be determined in advance from a fixed network and a fixed cost model. This is often sufficient for simple wayfinding, but it is less suitable once pedestrian density changes over time or local bottlenecks appear. In such cases, the shortest geometric path may no longer be the fastest or most reliable one. Recent work therefore treats indoor routing as a dynamic problem in which route cost depends on current conditions rather than geometry alone [Liu et al., 2021b; Wong and Lee, 2023; Tan et al., 2025].

A useful distinction is between hazard-aware rerouting and congestion-aware rerouting. In emergency studies, route updates are often driven by fire, smoke, or other environmental risks. Recent digital-twin-based systems follow this logic and update evacuation guidance in real time according to changing building conditions [Tan et al., 2025]. In everyday pedestrian navigation, however, the main issue may be crowding rather than hazard. Here, rerouting is triggered by queues, blocked passage, or anticipated delay at downstream connectors (see figure 2.5).

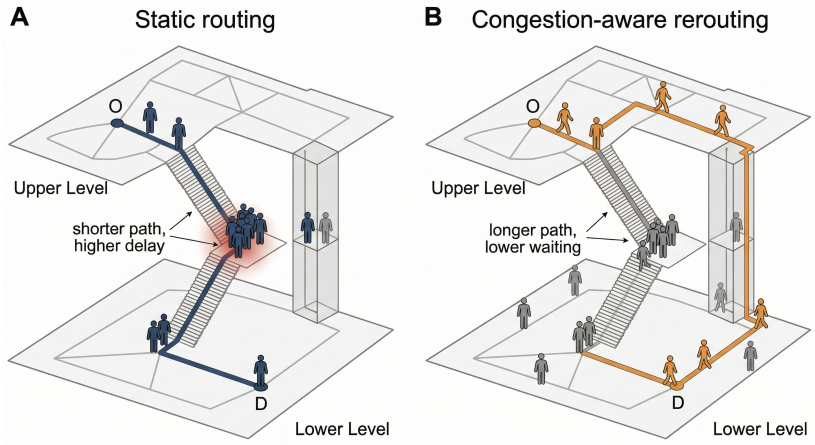


Figure 2.5.: Static routing versus congestion-aware rerouting on an indoor graph

This second line is closer to the focus of this thesis. Liu et al. [2021b] studied crowd-aware indoor path planning and showed that indoor paths should be evaluated by travel time under crowd conditions rather than by length alone. The general idea is consistent across this kind of studies: routing should respond to conditions on the network, not only to its static layout.

For this thesis, congestion-aware replanning is used in a limited but practical sense. Agents are first assigned routes on the 2.5D graph, but route choice can be revised when local congestion makes the current path less attractive. The purpose is not to model full behavioural adaptation. It is to test whether a semantically enriched graph derived from IFC can support basic dynamic rerouting and whether this improves overall movement performance compared with static routing. This question is directly linked to the experimental comparison later in the thesis.

2.4. Research Gap

The literature shows clear progress in indoor navigation from several directions. BIM- and IFC-based studies have demonstrated that building models provide useful geometry and semantics for deriving indoor navigation structures. Other work has shown that indoor routing can also be built from floor plans, trajectories, point clouds, or localization systems. In parallel, recent studies on dynamic routing and evacuation have moved beyond static shortest-path computation and started to consider changing indoor conditions.

However, three gaps remain.

First, many BIM-based studies still focus on connectivity extraction, model conversion, or route generation, while giving less attention to how the resulting graph supports later movement analysis. A navigation graph is often treated as an end product, rather than as a simulation-ready structure with explicit movement constraints.

Second, the transformation from BIM to navigable graph is often under-specified. In complex buildings, this transformation requires more than extracting doors and room adjacency.

It also requires filtering public circulation space, handling barriers and overhead objects, validating connectors, and deciding how stairs, escalators, gates, and other circulation elements should be represented in the graph. These modelling choices can affect route quality and simulation outcomes, but they are not always discussed in a systematic way.

Third, recent work on dynamic indoor routing has shown the value of real-time or condition-aware path updates, yet this line of work is often separated from BIM-based graph construction. As a result, there is still limited work that connects semantically filtered IFC input, explicit 2.5D graph modelling, and congestion-aware pedestrian simulation within one reproducible workflow.

This thesis addresses these gaps by developing a pipeline that starts from IFC BIM, constructs a semantically enriched 2.5D indoor navigation graph, and then evaluates this graph through routing and congestion-aware pedestrian simulation in a complex station environment. The focus is not only on extracting connectivity, but on testing whether the resulting graph is suitable for movement analysis under realistic circulation constraints.

3. Methodology

This chapter presents the methodology used in this thesis. Its purpose is to define the general workflow for transforming construction-oriented IFC BIM data into a semantically enriched 2.5D indoor navigation graph and for using this graph as the basis for graph-based pedestrian simulation and congestion-aware replanning.

The chapter does not report experimental results. Instead, it establishes the methodological basis for the validation work in Chapter 5 and the results in Chapter 6. The central idea is that the navigation graph is not treated as the final output of the thesis. Rather, it is an intermediate representation that enables movement analysis. The methodology therefore links three components: semantic interpretation of IFC data, construction of a 2.5D navigation graph, and graph-based pedestrian simulation under static and congestion-aware routing strategies.

3.1. Methodological Framework

The proposed workflow can be understood as a transformation from building information to movement analysis. The input is a multi-file IFC BIM model, which contains detailed geometric and semantic information about the station environment. The intended output is a semantically enriched 2.5D indoor navigation graph that can support routing, agent-based pedestrian simulation, and congestion-aware replanning.

At a high level, the workflow consists of six linked stages. First, the IFC model is interpreted and filtered from a navigation perspective. This stage identifies relevant building levels, public circulation areas, barriers, and vertical connectors. Second, walkable geometry is extracted from the traffic-relevant subset of the BIM data. Third, walkable nodes are sampled on each selected public level. Fourth, a 2.5D navigation graph is constructed by connecting nodes within each level and adding explicit inter-level connector edges. Fifth, the graph is used for origin–destination routing and graph-based pedestrian simulation. Sixth, static and congestion-aware routing strategies are compared through controlled experiments.

Figure 3.1 illustrates the overall methodological framework. The left part of the workflow concerns representation and graph construction, while the right part concerns routing, pedestrian movement simulation, and evaluation. This distinction is important because the thesis is not only concerned with whether a graph can be generated from IFC BIM, but also with whether the generated graph can support meaningful movement analysis.

The methodology is deliberately modular. Each stage produces intermediate outputs that can be inspected before the next stage is executed. This is necessary because errors in IFC interpretation, walkable-space extraction, or connector modelling can propagate into routing and simulation results. The modular structure also supports reproducibility: the same input data, configuration settings, and processing scripts can be reused to regenerate the navigation graph and simulation setup.

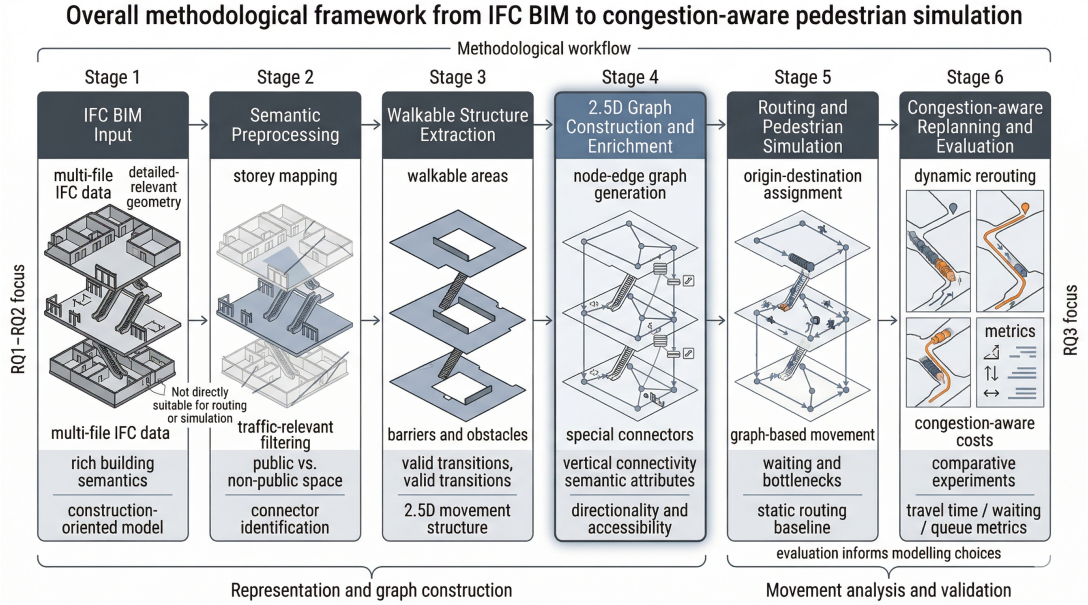


Figure 3.1.: Overall methodological framework from IFC BIM to congestion-aware pedestrian simulation. The workflow converts construction-oriented BIM data into a semantically enriched 2.5D navigation graph and then uses this graph for routing, simulation, and comparative validation.

3.1.1. IFC semantic interpretation

IFC BIM provides detailed object-level information about building components, but it is not directly organized as a pedestrian navigation model. Building elements such as walls, slabs, stairs, spaces, gates, and doors are represented according to construction and asset-management logic rather than according to pedestrian traversability. Therefore, the first methodological step is to reinterpret IFC semantics from a movement perspective.

This thesis distinguishes between four navigation-relevant semantic categories. First, *storey information* defines the vertical organization of the building and provides the basis for separating movement into floor-based layers. Second, *walkable surfaces* such as public floor slabs and circulation spaces define where horizontal movement may occur. Third, *barriers and restricted areas* such as walls, technical rooms, track areas, and inaccessible spaces define where movement should be blocked. Fourth, *connectors* such as stairs, escalators, elevators, gates, and platform doors define transitions between floors or between functional movement zones.

This semantic reinterpretation is necessary because the presence of an IFC object does not automatically indicate its navigation role. A slab may be geometrically present but not publicly accessible; a wall may act as a hard obstacle; a gate may act as a controlled transition rather than a simple barrier; and a stair or escalator must be represented as an inter-level connector rather than as ordinary geometry. Similar issues are discussed in previous BIM-based indoor navigation studies, where IFC is treated as a rich but construction-oriented source that must be transformed into a navigation-oriented representation before routing can be performed [Isikdag et al., 2013; Liu et al., 2021a; Diakite et al., 2022].

The output of this step is therefore not yet a graph. It is a filtered and movement-oriented interpretation of the IFC model, in which building elements are classified according to their role in pedestrian movement. This semantic interpretation provides the basis for walkable-space extraction and graph construction in the following steps.

3.1.2. 2.5D abstraction

The navigation model in this thesis is represented as a 2.5D graph. This means that horizontal movement is modelled on floor-based walkable planes, while vertical movement is represented through explicit inter-level connectors. The model is therefore more detailed than a purely topological room-adjacency graph, but less complex than a full 3D free-space navigation model (see figure 3.2).

The choice of a 2.5D abstraction follows from the movement structure of multi-level indoor environments. Pedestrians normally move on floors, platforms, concourses, and corridors, and change height only through specific elements such as stairs, escalators, and elevators. Representing the whole building as full 3D navigable space would introduce unnecessary geometric complexity for graph-based routing and simulation. Conversely, representing the station as a simple 2D or room-level graph would lose important information about vertical circulation, connector bottlenecks, and local route choices.

In this thesis, the 2.5D abstraction is defined as follows. Public levels are represented as two-dimensional walkable layers. Stairs and escalators are represented as sloped connector chains between layers. Elevators are represented as vertical connectors. Other transition elements, such as gates or platform doors, are represented as semantic graph elements that may affect route cost, accessibility, or local capacity. This interpretation is consistent with the broader idea that indoor navigation requires a graph abstraction that preserves both topology and movement-relevant semantics [Kang and Li, 2017; Zhou et al., 2022; Hamzei et al., 2024].

From 3D IFC geometry to 2.5D walkable layers

A key step in the methodology is the conversion from 3D IFC geometry to 2.5D walkable graph layers. The IFC model contains three-dimensional building elements organized according to construction logic. The navigation graph, however, requires a movement-oriented representation in which horizontal walking is defined on floor surfaces and vertical movement is allowed only through valid connectors. This conversion is therefore not a simple dimension reduction, but a geometric and semantic filtering process.

The first step is storey interpretation. IFC storeys are used to identify the vertical organization of the station. Each selected public storey is assigned a representative elevation z_l, where l denotes the level. In the station case, the public movement levels are treated as separate horizontal graph layers, while non-public or technical levels are excluded from ordinary pedestrian movement unless they are needed for interpreting connector continuity. This preserves the vertical order of the station while avoiding a full 3D free-space search.

The second step is geometric projection. For each selected level, movement-relevant floor surfaces are projected to the horizontal xy-plane. A point on a 3D floor surface,

$$p = (x, y, z),$$

is represented in the corresponding walkable layer as

$$p_l = (x, y, z_l),$$

where the horizontal coordinates define the planimetric movement position and z_l stores the level elevation. In this way, the geometry is simplified for graph construction while the level identity and vertical separation are retained.

The third step is walkable-area derivation. For each public level l, a candidate walkable polygon W_l is obtained from the movement-relevant floor or circulation area P_l by removing barriers and restricted regions:

$$W_l = P_l \setminus (B_l \cup R_l),$$

where B_l represents physical barriers such as walls, columns, and obstacle footprints, and R_l represents restricted or non-public areas such as technical rooms, track areas, or other spaces not intended for ordinary pedestrian movement. Elements that are geometrically present in the IFC model but irrelevant to pedestrian-level movement, such as overhead components or facade detail, are ignored in this step.

The fourth step is node sampling and clearance filtering. Candidate nodes are sampled on each walkable polygon using the selected grid resolution. A candidate node is retained only if it lies inside the valid walkable area and satisfies the clearance threshold from barriers and restricted regions. This prevents the graph from placing nodes directly on walls, inside obstacles, or in spaces that are too close to non-traversable geometry. Each retained node stores its horizontal coordinates, level elevation, level identifier, and semantic attributes. The sampling resolution controls the trade-off between geometric fidelity and computational cost. In this thesis, 0.5 m is used as the default sampling interval for the main graph. This resolution is chosen because it provides a human-scale representation of walkable space and connector surroundings while keeping the graph size manageable for repeated simulation experiments. A coarser 0.75 m sparse graph is generated separately to evaluate the sensitivity of the results to graph abstraction.

The fifth step is horizontal edge construction. Candidate edges are created between neighbouring nodes on the same level. An edge is retained only if the line segment between the two nodes stays within the walkable area and does not cross barriers or restricted regions. This line-of-sight validation prevents the graph from connecting nodes through walls or across inaccessible spaces.

The final step is connector integration. Stairs and escalators are not flattened into ordinary floor edges. Instead, they are represented as sloped connector chains between levels. Elevators are represented as vertical connector relations. For each connector, the lower and upper endpoints are projected or snapped to the nearest valid graph nodes on the corresponding levels. The resulting connector edges store connector type, directionality where relevant, level transition, and movement cost. This allows the graph to preserve the 3D function of vertical circulation while remaining a 2.5D movement model.

This procedure makes the 2.5D abstraction reproducible. The horizontal part of the graph is generated from projected and filtered walkable layers, while the vertical part is generated from explicitly modelled connectors. The result is therefore not a generic 2D projection of the IFC model, but a semantically filtered 2.5D graph that preserves the movement-relevant structure of the original 3D station model.

The main advantage of this abstraction is that it balances geometric detail, semantic richness, and computational usability. The resulting graph can represent multi-level movement and connector-specific constraints while remaining compact enough for repeated routing and agent-based simulation. The graph-construction procedure is summarized in Algorithm 3.1. The algorithm abstracts the implementation into semantic filtering, walkable-space extraction, node sampling, connector modelling, and graph validation steps.

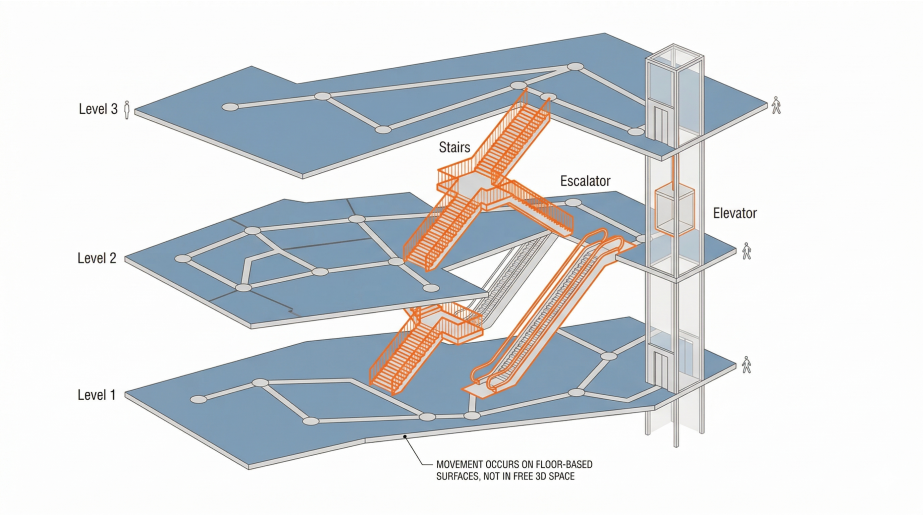


Figure 3.2.: Definition of the 2.5D abstraction used in this thesis. Horizontal movement is represented on floor-based walkable planes, while stairs and escalators are represented as sloped inter-level connectors and elevators as vertical connectors.

3.1.3. Graph-based pedestrian simulation

After the 2.5D navigation graph has been constructed, it is used as the movement substrate for pedestrian simulation. In this thesis, simulation is performed on graph nodes and edges rather than in continuous physical space. Agents are assigned origins and destinations, compute routes on the graph, and move along graph edges in discrete simulation time.

This modelling choice reflects the purpose of the thesis. The goal is not to reproduce detailed microscopic pedestrian behaviour at the body-interaction level. Instead, the simulation is used to test whether a BIM-derived 2.5D graph can support interpretable movement analysis. The main observed phenomena are route use, travel time, waiting time, queue formation, connector load, and route redistribution under congestion.

In the graph-based simulation, each edge represents a possible movement relation. Edge length, connector type, accessibility, and congestion state can influence movement cost. Agents may wait when the next node or connector is occupied or when a connector capacity constraint is reached. This allows the simulation to represent bottlenecks at stairs, escalators, gates, or other constrained transitions without requiring a full continuous-space pedestrian model.

Graph-based and agent-based approaches are widely used in indoor routing and evacuation analysis because they provide a practical link between spatial representation and movement

Algorithm 3.1: IFC-to-2.5D navigation graph construction

Input: Multi-file IFC BIM data, selected public levels, grid resolution, clearance threshold

Output: Semantically enriched 2.5D navigation graph for routing and movement analysis

1 Load IFC files and extract storeys, floor surfaces, obstacles, and connector candidates;
2 Identify public circulation levels and exclude non-public or technical levels from ordinary pedestrian movement;
3 Classify BIM elements into walkable surfaces, barriers, restricted areas, and vertical or access-control connectors;
4 Project selected public floor surfaces to level-based xy-walkable layers while retaining the representative level elevation z_l;
5 Derive walkable polygons $W_l = P_l \setminus (B_l \cup R_l)$ by removing barriers and restricted areas from public floor areas;
6 Sample candidate graph nodes inside each walkable polygon using the chosen grid resolution;
7 Remove candidate nodes that violate clearance or accessibility constraints;
8 Connect neighbouring nodes within each level only if the connecting segment remains inside the valid walkable polygon;
9 Validate connector candidates such as stairs, escalators, elevators, gates, and platform doors;
10 Snap connector endpoints to the nearest valid graph nodes;
11 Add inter-level connector edges and assign connector type, directionality, and movement attributes;
12 Validate graph connectivity and retain the navigation-relevant connected component;
13 Attach semantic attributes to nodes and edges, including level, connector type, accessibility, and movement cost;
14 Return the semantically enriched 2.5D navigation graph;

behaviour [Senanayake et al., 2024; Xie et al., 2024; Bhardwaj et al., 2024]. In this thesis, the graph-based simulator is used as an evaluation layer: it tests whether the constructed navigation graph is useful not only for shortest-path routing, but also for controlled pedestrian movement analysis.

3.1.4. Congestion-aware replanning

The final methodological component is congestion-aware replanning. Static routing computes a route from an origin to a destination using fixed graph costs. This is useful as a baseline, but it does not account for congestion that develops during simulation. In a station environment, fixed shortest-path routing can concentrate many agents on the same stair, escalator, gate, or corridor, causing local waiting and queue formation.

Congestion-aware replanning extends the static routing model by allowing route choice to respond to changing network conditions. In this thesis, congestion is represented through graph-level signals such as node occupancy, connector load, waiting time, and downstream crowding. These signals can increase the temporary cost of congested parts of the graph or trigger route recalculation for selected agents. The replanning mechanism therefore tests whether alternative routes can reduce waiting and queue pressure, even when they are geometrically longer.

This distinction between geometric shortest path and movement-efficient path is important for congestion-sensitive indoor environments. Previous studies have shown that indoor routing should consider crowd conditions and changing local constraints, rather than relying only on static geometric distance [Liu et al., 2021b; Wong and Lee, 2023; Tan et al., 2025]. The present thesis adopts this idea in a limited but controlled way: the aim is not to model full behavioural adaptation, but to evaluate whether a semantically enriched 2.5D graph derived from IFC can support dynamic route updates.

The comparison between static routing and congestion-aware replanning forms one of the main experimental tests in Chapter 5. If congestion-aware replanning reduces waiting time or maximum queue length while maintaining high arrival rates, this indicates that the graph is suitable for more than static connectivity analysis. It can also support movement-performance evaluation under changing local conditions. The graph-based simulation and replanning logic is summarized in Algorithm 3.2. The same 2.5D graph is used for both static and congestion-aware routing; the difference is whether graph costs and agent paths are updated during the simulation.

3.2. Tools and Implementation Environment

The implementation of this thesis was carried out in a Python-based environment. The workflow combines BIM parsing, geometric processing, graph construction, agent-based simulation, and result visualization. The implementation was organized as a modular pipeline so that each processing stage could be inspected separately before being used in the next stage.

Table 3.1 summarizes the main tools used in the thesis and their roles in the workflow.

Algorithm 3.2: Graph-based pedestrian simulation with congestion-aware replanning

Input: 2.5D navigation graph, origin–destination pairs, agent profiles, routing strategy, simulation horizon
Output: Agent trajectories, travel times, waiting times, queue metrics, connector loads, and reroute counts

```
Initialize agents with origins, destinations, departure times, and profile-specific movement parameters;
Compute an initial shortest path for each agent using the static graph cost;
Initialize occupancy, waiting-time, queue-length, and reroute counters;
for each simulation time step do
    Spawn agents whose departure time has been reached;
    for each active agent do
        if the agent has reached its destination then
            Mark the agent as arrived and record travel time;
        else
            Identify the next node or edge on the agent's current path;
            if movement to the next graph position is feasible under occupancy and connector-capacity constraints then
                Move the agent to the next graph position;
            else
                Keep the agent at its current graph position and accumulate waiting time;
            if congestion-aware routing is enabled and a replanning condition is triggered then
                Update temporary graph costs using local occupancy, connector load, or downstream congestion;
                Recompute the path from the agent's current position to its destination;
                Increment the agent's reroute counter;
    Record network occupancy, queue length, waiting time, connector load, and arrival status;
Export trajectories and aggregate simulation metrics for experimental comparison;
```

Table 3.1.: Main tools and implementation environment used in the thesis.

Category	Tool or environment	Role in the thesis
Programming environment	Python, Visual Studio Code	Main development environment for the IFC-to-graph pipeline, pedestrian simulation, and result processing.
BIM processing	IfcOpenShell	Reading IFC files, accessing BIM entities, extracting storeys, elements, and geometry-related information.
Geometry processing	Shapely, NumPy	Processing floor surfaces, obstacles, walkable areas, clearance checks, and geometric relationships.
Graph modelling	NetworkX	Constructing the 2.5D navigation graph, checking graph connectivity, and computing shortest paths.
Simulation	Custom Python scripts	Implementing graph-based pedestrian movement, occupancy checks, waiting behaviour, and congestion-aware replanning.
Data storage	JSON, GeoJSON, CSV	Storing intermediate outputs, graph data, semantic anchors, trajectories, and experimental metrics.
Visualization	Matplotlib, QGIS, exported figures	Producing graph visualizations, route maps, occupancy summaries, queue plots, and case-study illustrations.
Writing and documentation	Overleaf, LaTeX	Thesis writing, figure integration, table formatting, and bibliography management.

The implementation was developed iteratively. First, a simplified two-floor model was used to test the basic logic of geometric extraction, node sampling, graph construction, and routing. After the workflow became stable, the method was transferred to a more complex real station case. This reduced the risk of directly debugging the full workflow on a large and irregular BIM dataset.

AI-assisted tools were used in a controlled way during implementation, writing, and figure preparation. GitHub Copilot supported coding and debugging in Visual Studio Code. ChatGPT supported language polishing, structural clarification, LaTeX formatting, figure-caption improvement, and result-interpretation checking. AI-assisted image generation was used for some schematic explanatory figures, while Gemini was used for presentation preparation and visual-communication brainstorming. A detailed statement is provided in Appendix C. All code, figures, experimental outputs, and interpretations were checked and revised by the author.

3.2.1. Code availability and reproducibility material

The implementation scripts, configuration files, selected processed outputs, and figure-generation materials are documented in the project repository and online demonstration page:

`https://github.com/SueWang291826/tud-thesis-showcase`
`https://suewang291826.github.io/tud-thesis-showcase/index.html`

The repository and demonstration material are intended to support inspection of the processing pipeline, experiment configuration, and result-generation workflow. They include the main workflow logic for graph construction, routing, simulation, experiment execution, and visualization where redistribution is allowed.

The raw Dadongmen Station BIM/IFC data are not redistributed because they originate from professional project data and were used with permission for academic research only. Therefore, the public material does not provide the full raw station model. Instead, the thesis provides methodological descriptions, configuration summaries, selected processed non-sensitive outputs, and figures sufficient to document the workflow and experimental results.

3.3. Chapter Summary

This chapter defined the methodological basis of the thesis. It described how construction-oriented IFC BIM is reinterpreted from a navigation perspective, abstracted into a semantically enriched 2.5D graph, and then used as the substrate for graph-based pedestrian simulation and congestion-aware replanning.

Two conceptual algorithms were provided to make the workflow explicit: one for IFC-to-2.5D graph construction and one for graph-based simulation with congestion-aware replanning. The chapter also summarized the implementation environment and clarified the controlled use of AI-assisted tools for coding support and writing refinement.

The next chapter presents the simplified pilot case used to check the internal correctness of the workflow before the full station-scale experimental setup is introduced in Chapter 5.

4. Case Study

This chapter presents the simplified two-floor model used as the pilot case study in this thesis. The role of this case is not to provide the final experimental validation, but to support the development and checking of the IFC-to-2.5D graph workflow under controlled conditions. The model is used to test whether the pipeline can extract walkable space, sample valid nodes, generate graph edges, represent vertical connection, and compute origin–destination routes before the method is applied to the full station case in the following chapters.

4.1. Source Data

The source data for this chapter consist of a simplified two-floor indoor model derived from the author's previous undergraduate design assignment. The model was selected because it contains the minimum spatial structure required for testing the proposed workflow: two connected floor levels, indoor boundaries, walkable areas, obstacles, and a vertical connector.

The model is not intended to represent a real metro station or to support final quantitative conclusions. Instead, it is used as a pilot case for methodological testing. Its smaller scale and clearer geometry make it suitable for visual inspection of intermediate outputs such as extracted geometry, sampled nodes, graph edges, and cross-level routes.

Table 4.1 summarizes the source data used in this chapter.

Table 4.1.: Source data of the simplified pilot case.

Item	Description
Model source	Previous undergraduate design assignment by the author.
Model type	Simplified two-floor indoor building model.
Main spatial elements	Two walkable floor levels, indoor boundary geometry, obstacles, and one vertical connector.
Role in the thesis	Pilot case for testing the IFC-to-2.5D graph workflow before applying it to the full station case.
Use in final validation	Not used as the final experimental case or as the source of final quantitative results.
Main outputs	Extracted geometry, sampled walkable nodes, horizontal graph edges, vertical connector edges, semantic OD anchors, and example routes.
Data availability	Internal academic design model; used only for methodological testing and thesis illustration.

The simplified model is therefore positioned as a methodological pilot. It provides a controlled environment for checking the internal correctness of the workflow before the same logic is transferred to a more complex station model.

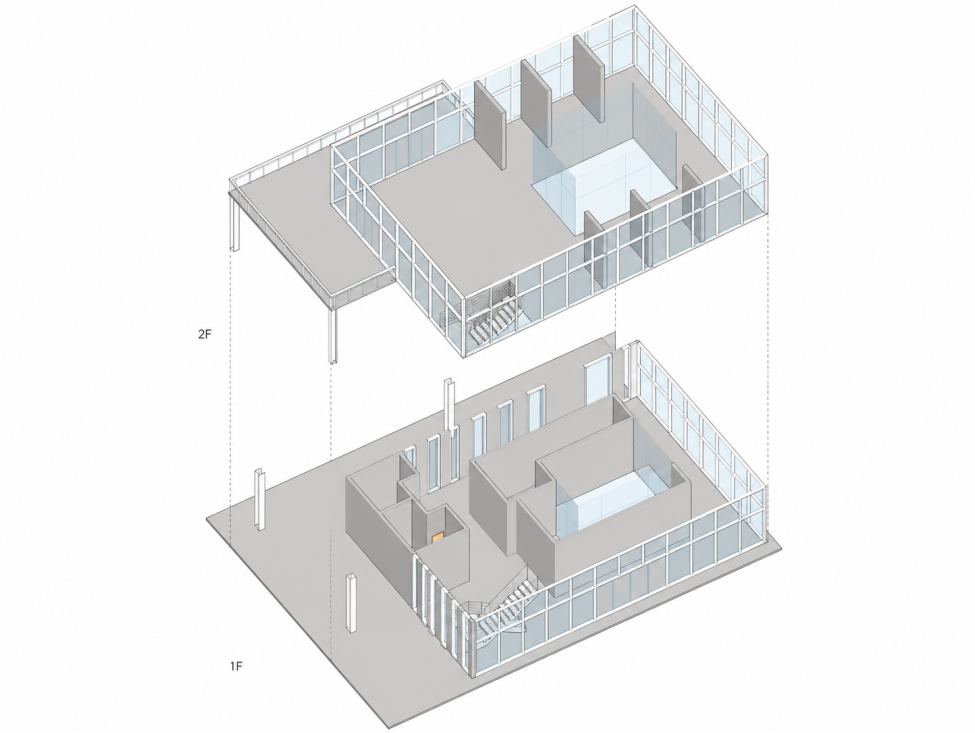


Figure 4.1.: Simplified two-floor model used as the pilot case study. The model provides a controlled environment for testing walkable-space extraction, vertical connection modelling, and 2.5D graph construction.

4.2. Case Selection Rationale

The simplified two-floor model was selected because it provides a controlled setting for testing the proposed workflow. At the early implementation stage, directly applying the method to a large and irregular station BIM model would make it difficult to identify whether an error came from the algorithm, the source geometry, the semantic interpretation, or the complexity of the case. A smaller model makes the behaviour of each processing step easier to inspect.

The pilot case was selected according to four criteria. First, it contains more than one floor, so the workflow must handle vertical movement rather than only single-floor routing. Second, it includes indoor obstacles and walkable areas, so the workflow must distinguish traversable and non-traversable space. Third, it is simple enough for manual checking of graph outputs. Fourth, its design origin is known, which makes interpretation of the model easier.

Table 4.2 summarizes the selection logic.

Table 4.2.: Rationale for using the simplified two-floor model as a pilot case.

Selection criterion	Reason	Methodological relevance
Multi-floor structure	The model contains two connected levels.	Tests whether the workflow can represent vertical movement in a 2.5D graph.
Walkable and non-walkable space	The model contains movement areas, boundaries, and obstacles.	Tests whether the workflow can separate traversable and non-traversable space.
Controlled complexity	The model is smaller and cleaner than a full station BIM model.	Allows intermediate outputs to be visually inspected and debugged.
Known design origin	The model was created in a previous undergraduate design assignment by the author.	The spatial layout and design intent are known, making interpretation more reliable.
Pilot role only	The model is not used for final quantitative validation.	Prevents the pilot case from being over-interpreted as a real-world simulation result.

The simplified model is therefore used to verify the basic workflow logic. It is not used to claim real-world station performance. The full station case and final validation are introduced in the following chapters.

4.3. Spatial Scope

The spatial scope of the simplified model is limited to two connected indoor floors. Each floor is treated as a horizontal movement layer, while the stair or vertical connection is treated as an explicit inter-level connector. This follows the same 2.5D modelling principle defined in Chapter 3, but in a smaller and more controlled spatial setting.

The simplified model includes three main types of spatial elements. The first type is walkable floor area, which provides the candidate space for graph node sampling. The second type is obstacle or boundary geometry, which constrains movement and prevents invalid graph connections. The third type is the vertical connector, which links the two floor layers and allows cross-level routing.

Table 4.3 summarizes how the main spatial components are interpreted.

Figure 4.2 shows the main intermediate outputs of the pilot workflow. These outputs are used to verify whether the workflow behaves correctly at each stage: geometry extraction, node sampling, graph generation, and cross-level routing.

The purpose of this spatial simplification is not to create a realistic passenger-flow model. Instead, it allows the assumptions of the 2.5D graph representation to be tested in a transparent way.

Table 4.3.: Spatial scope of the simplified two-floor model.

Spatial component	Interpretation	Use in the pilot workflow
Floor 1	Lower horizontal movement layer.	Used for walkable-space extraction and graph node sampling.
Floor 2	Upper horizontal movement layer.	Used for walkable-space extraction and graph node sampling.
Walkable area	Candidate pedestrian movement space.	Converted into valid graph nodes after filtering.
Walls or obstacles	Non-traversable indoor geometry.	Used to remove invalid nodes and prevent graph edges crossing blocked space.
Stair or vertical connector	Inter-level movement element.	Represented as connector nodes or connector edges between the two floor layers.
Non-essential architectural detail	Geometry not relevant to pedestrian movement.	Ignored or simplified during pilot graph construction.

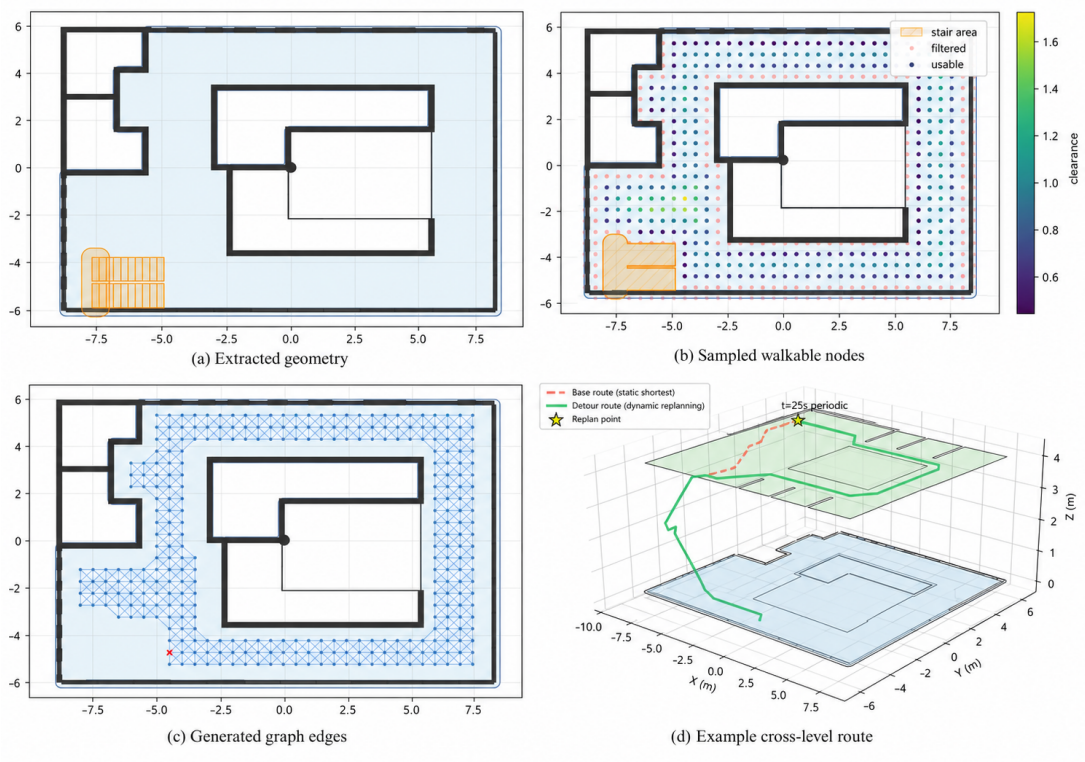


Figure 4.2.: Intermediate outputs of the simplified two-floor pilot workflow: (a) extracted floor geometry and obstacles, (b) sampled walkable nodes after filtering, (c) generated graph nodes and edges, and (d) an example cross-level route on the 2.5D representation. These outputs were used to check the internal correctness of the workflow before applying it to the full station case.

4.4. Origin–Destination Semantics

Origin–destination semantics were defined in the simplified model to test whether graph routing can operate on meaningful movement targets rather than arbitrary coordinates. In the pilot case, the origins and destinations are represented by simple semantic anchors placed on the two floor layers. These anchors represent entry points, exit points, or target areas in the indoor layout.

The OD anchors are not treated as independent route geometry. Instead, each semantic anchor is snapped to the nearest valid graph node before route computation. This ensures that routing is performed on the generated navigation graph rather than on manually drawn lines. The same principle is later used for the full station case, where entrances, exits, platforms, concourses, and connector landings are mapped to graph nodes.

Table 4.4 summarizes the OD semantics used in the pilot case.

Table 4.4.: Origin–destination semantics in the simplified pilot model.

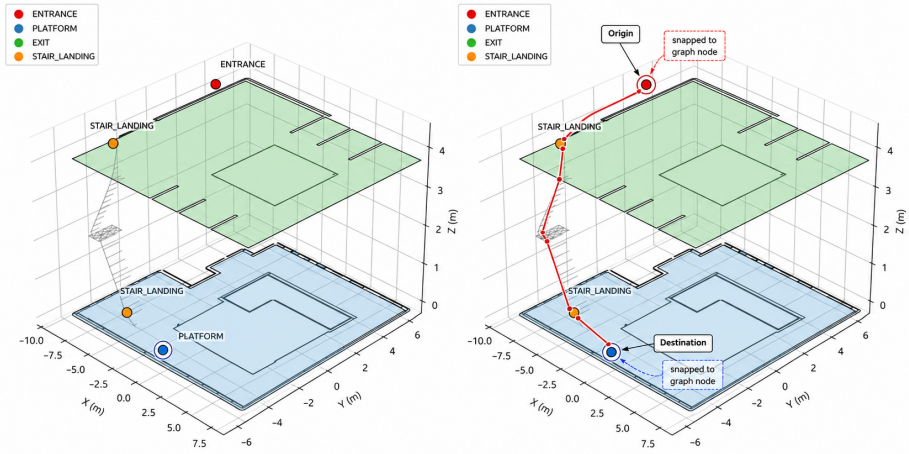
Semantic anchor	Meaning in the pilot model	Role in workflow testing
Lower-floor origin	Starting area on Floor 1.	Tests route generation from the lower movement layer.
Upper-floor destination	Target area on Floor 2.	Tests cross-level routing through the vertical connector.
Upper-floor origin	Starting area on Floor 2.	Tests reverse or alternative cross-level movement.
Lower-floor destination	Target area on Floor 1.	Tests whether routing works in both movement directions.
Connector anchor	Graph node near the stair or vertical connector.	Tests whether horizontal graph layers are correctly linked through the connector.

Figure 4.3 illustrates the use of semantic anchors and graph-based cross-level routing in the simplified model.

The OD setup in the simplified model is deliberately minimal. It is used to verify basic routing correctness: whether a route can be found, whether the route remains inside walkable space, whether obstacles are avoided, and whether cross-level movement uses the intended connector. More complex OD demand, agent spawning, and scenario comparison are defined later in the experimental design chapter.

4.5. Pilot Outputs and Relation to the Full Workflow

The simplified model produced several intermediate outputs that were useful for method development. These include extracted walkable areas, sampled graph nodes, horizontal graph edges, connector edges, semantic anchors, and example routes. These outputs were used to check whether the pipeline behaved correctly before it was transferred to the full station case.



(a) Semantic OD anchors (b) Snapped cross-level route

Figure 4.3.: Origin–destination semantics in the simplified pilot model: (a) semantic anchors placed on the 2.5D representation, and (b) an illustrative cross-level route after snapping the origin and destination anchors to valid graph nodes.

The pilot case helped verify four aspects of the workflow. First, walkable-space extraction should produce nodes only in valid movement areas. Second, obstacle treatment should prevent graph edges from crossing non-traversable space. Third, the vertical connector should correctly link the two floor layers. Fourth, routing should produce a continuous path between semantic origins and destinations.

Table 4.5 summarizes the pilot outputs and their relation to the full workflow.

The simplified model is therefore used as a pilot case, not as the final validation case. Its value lies in making the workflow inspectable and reducing implementation risk before the method is applied to a larger and more complex station model.

4.6. Chapter Summary

This chapter introduced the simplified two-floor model used as the pilot case study. The model was selected because it provides a controlled setting for testing walkable-space extraction, node sampling, 2.5D graph construction, vertical connection modelling, and basic origin–destination routing.

The chapter also clarified the role of the pilot case in the thesis. The simplified model is not used for final quantitative validation. Instead, it is used to check the internal correctness of the workflow before applying the same methodology to the full station case. The next chapter defines the experimental design and validation strategy used for the main evaluation.

Table 4.5.: Pilot outputs from the simplified model and their relation to the full workflow.

Pilot output	What was checked	Relation to the full workflow
Walkable-space layer	Whether movement areas were extracted correctly.	Supports the extraction of public walkable areas in the full station case.
Sampled graph nodes	Whether nodes were generated only in valid locations.	Provides the same node-sampling logic used on public movement levels.
Horizontal graph edges	Whether local movement relations were valid within each floor.	Provides the same edge-construction logic used on each movement level.
Vertical connector edges	Whether two floors were correctly connected.	Extends to stairs, escalators, and elevators in the full station case.
Example OD routes	Whether cross-level routing produced continuous paths.	Supports later routing between entrances, concourses, platforms, and exits.
Basic simulation test	Whether agents could move along graph paths without structural failure.	Forms the basis for static and congestion-aware simulation experiments.

5. Experimental Design and Validation

This chapter defines the experimental design used to validate the proposed IFC-to-2.5D graph workflow and the graph-based pedestrian simulation model, as summarized in Figure 5.1.

The experimental design is aligned with the research questions introduced in Chapter 1. First, the experiments test whether the generated 2.5D graph is structurally usable for indoor routing and simulation, which supports the transformation question in RQ1. Second, they examine how modelling choices, congestion thresholds, and graph assumptions influence navigability and simulation behaviour, which supports RQ2. Third, they compare fixed static routing with congestion-aware replanning, which directly addresses RQ3. Fourth, they evaluate pedestrian-profile effects through both single-profile and mixed-agent simulations, which supports RQ4.

In addition to these research-question-driven experiments, the chapter includes a capacity-oriented stress test. This test does not claim to estimate the official passenger capacity of the real station. Instead, it examines how the generated station graph and simulation logic behave when pedestrian demand increases. This provides a robustness check for the proposed workflow.

The experiments are based on the full station graph generated from the Dadongmen Station BIM model. The simplified two-floor model introduced in Chapter 4 is used only as a pilot case for workflow checking. The final experimental validation in this chapter and the results in Chapter 6 are based on the station-scale graph.

5.1. Validation Strategy

The validation strategy follows the research questions of the thesis. The experiments are designed to test not only whether a 2.5D graph can be generated from IFC BIM, but also whether the generated graph is suitable for routing, pedestrian simulation, congestion-aware replanning, and heterogeneous-agent analysis.

The validation is organized into five levels. The first level concerns graph validity: the graph must represent public walkable levels, vertical connectors, and OD reachability in a coherent way. The second level concerns routing validity: the graph must support meaningful route computation between semantically defined origins and destinations. The third level concerns simulation validity: agents must be able to move, wait, queue, and arrive under controlled settings. The fourth level concerns comparative movement performance: congestion-aware replanning should produce interpretable differences compared with static routing. The fifth level concerns robustness: algorithmic thresholds and increasing demand should reveal the sensitivity and capacity limits of the proposed workflow.

Table 5.1 summarizes how the validation levels correspond to the thesis questions.

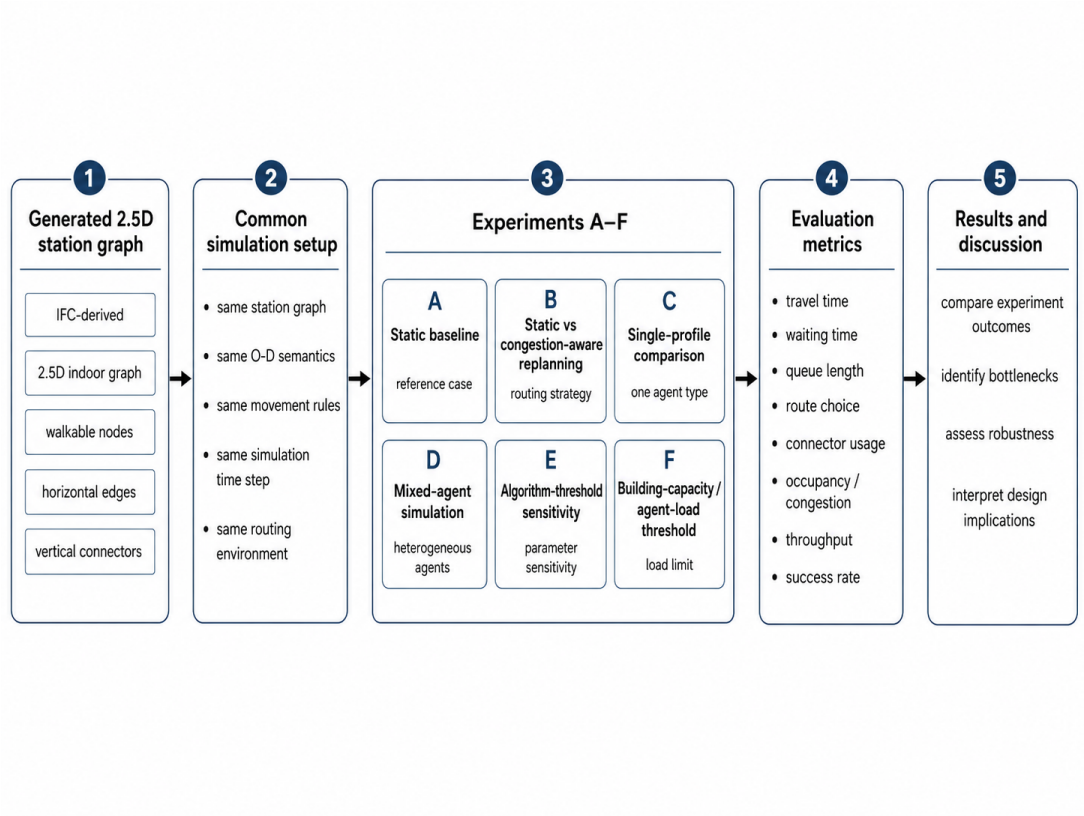


Figure 5.1.: Overview of the experimental design. The same station-scale 2.5D graph is used as the input for controlled routing and pedestrian simulation experiments.

Table 5.1.: Validation strategy and relation to the research questions.

Validation level	Main question	Related research question
Graph validity	Does the generated 2.5D graph provide a coherent routing substrate?	RQ1
Routing validity	Can OD pairs be routed through public levels and valid connectors?	RQ1 and RQ2
Simulation validity	Can agents move, wait, queue, and arrive under controlled settings?	Main RQ and RQ1
Comparative performance	Does congestion-aware replanning change travel time, waiting, queue length, or connector load?	RQ3
Pedestrian heterogeneity	Do different pedestrian profiles produce different travel-time, waiting, and congestion patterns?	RQ4
Threshold and capacity robustness	How sensitive are the algorithm and station graph to threshold settings and increasing demand?	RQ2, RQ3, and robustness extension

The experiments are interpreted as controlled methodological validation rather than as fully calibrated operational passenger-flow prediction. No observed passenger trajectories are used for calibration. Therefore, the results should be read as evidence that the workflow can support structured routing and congestion-aware movement analysis, not as a direct prediction of real passenger behaviour in the station.

5.2. Experimental Setup

All experiments use a common simulation setup unless stated otherwise. The setup defines the graph input, agent parameters, routing strategies, and output recording procedure. This common setup allows the experiments to isolate specific modelling choices, such as routing logic, pedestrian profile, replanning threshold, or demand level.

5.2.1. Graph input

The graph input for the experiments is the semantically enriched 2.5D navigation graph generated from the Dadongmen Station BIM model of Hefei Metro Line 1 (see figure 5.2 and 5.3). The original model is professional project data from the author's previous internship-related design project. It is used with permission for academic research, but the raw BIM model is not redistributed.

For the experimental setup, the station is represented as a multi-level public transport environment. The model contains four relevant levels: Transport, Concourse, Equipment, and Platform. The Transport, Concourse, and Platform levels are treated as public pedestrian movement levels. The Equipment level is not used as an ordinary public movement level, but it may remain relevant for interpreting vertical connector continuity.

Table 5.2 summarizes the station levels used by the experimental graph.

Table 5.2.: Station levels represented in the experimental graph input.

Level ID	Level name	Public movement level	Role in the experimental graph
F4	Transport	Yes	Upper public circulation and station entrance/exit movement.
F3	Concourse	Yes	Main concourse, circulation, and access-control movement.
F2	Equipment	No	Non-public technical level; excluded from ordinary pedestrian simulation.
F1	Platform	Yes	Platform-side movement and train-access related destinations.

The graph contains horizontal movement edges on the selected public levels and explicit connector edges for stairs, escalators, elevators, and other relevant transition elements. The

graph is used as a shared base representation for all experiments. Differences between experiments are introduced through routing logic, pedestrian profile, replanning thresholds, or demand level rather than by changing the underlying building graph, except where a sensitivity test explicitly modifies a modelling assumption.

The default graph uses a 0.5 m sampling grid on the walkable layers. This value is selected as a practical compromise between human-scale spatial detail and computational feasibility. A finer grid would represent local geometry more densely, but would also increase the number of nodes and edges and make repeated simulation experiments more expensive. A coarser grid would reduce computational cost, but could miss local circulation details around stairs, escalators, gates, and platform-related transitions. The 0.5 m setting is therefore used as the main graph resolution because it is fine enough to capture station-scale connector geometry and local movement constraints, while remaining feasible for repeated routing, replanning, profile, and demand-load experiments.

To test whether this graph-resolution choice affects the results, a sparse semantic-light graph variant is also generated using a coarser 0.75 m grid and a reduced neighbourhood rule. This variant is not used as the main graph, but as a graph-abstraction sensitivity case in Experiments E and F.

Figure 5.4 illustrates the semantic origin–destination setup used in the experiments. The OD regions are attached to valid graph nodes so that routing and simulation are performed on the generated 2.5D graph rather than on arbitrary point coordinates.

The online project showcase is used as supplementary visual material for presenting the graph and simulation outputs. It is not treated as a source of scientific evidence separate from the thesis results. The formal experimental setup and result interpretation are defined in this chapter and Chapter 6.

5.2.2. Agent settings

Agents represent pedestrians moving on the 2.5D station graph. Each agent is assigned an origin, a destination, a departure time, a pedestrian profile, and a routing strategy. The simulation is updated in discrete time steps. At each time step, active agents either move to the next graph position, wait because of local occupancy or connector-capacity constraints, or replan if congestion-aware routing is enabled.

The main controlled experiments use 200 agents within a simulation horizon of 600 s, corresponding to a 10-minute simulation window. This number should not be interpreted as a direct estimate of the real passenger volume of Dadongmen Station. Publicly available passenger-flow reports only provide a rough indication of the order of magnitude of metro use. They indicate that Dadongmen Station is a relevant high-use metro station in the Hefei metro network, but they do not provide time-resolved station-level counts for the specific entrances, platforms, directions, or 10-minute intervals used in this thesis [Hefei Yaohai District Government, 2018; China Metro, 2024; Sina Finance, 2024]. Therefore, it would be unrealistic to distribute a daily or long-period passenger total evenly over the full operating day.

In reality, metro demand is strongly time-dependent. A substantial share of passengers is expected to arrive during the morning and afternoon peak periods, while off-peak periods have much lower demand. For this reason, the experiments do not treat the 200-agent setting

Figure 5.2.: General location of Dadongmen Station in Hefei. The figure provides spatial context for the full station case without disclosing detailed project drawings.

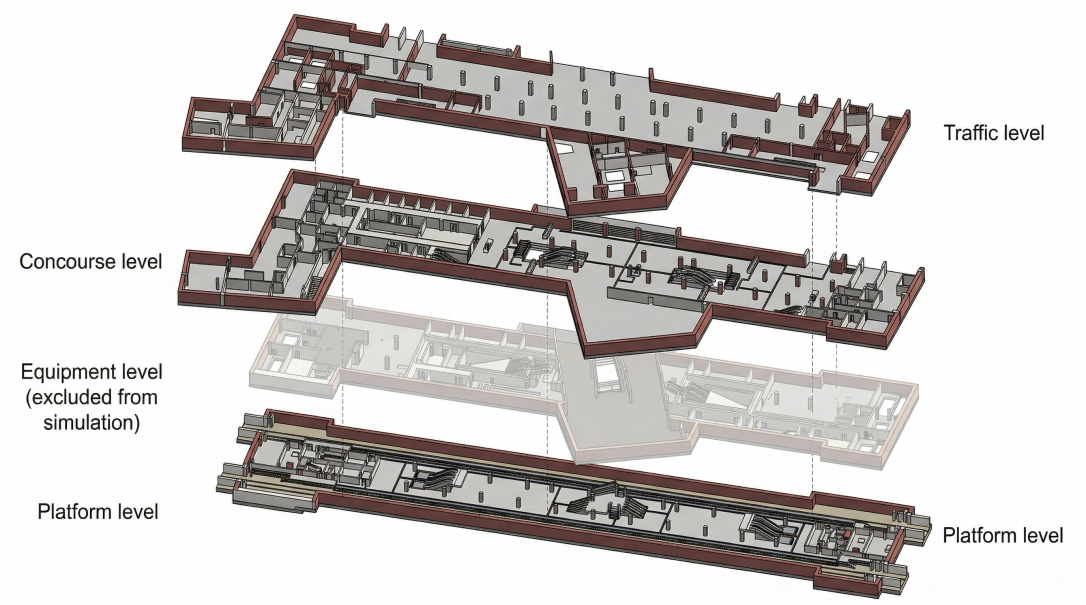


Figure 5.3.: Exploded-level representation of the station graph input. Public circulation levels are retained for pedestrian simulation, while the Equipment level is excluded from ordinary pedestrian movement but may remain relevant for connector interpretation.

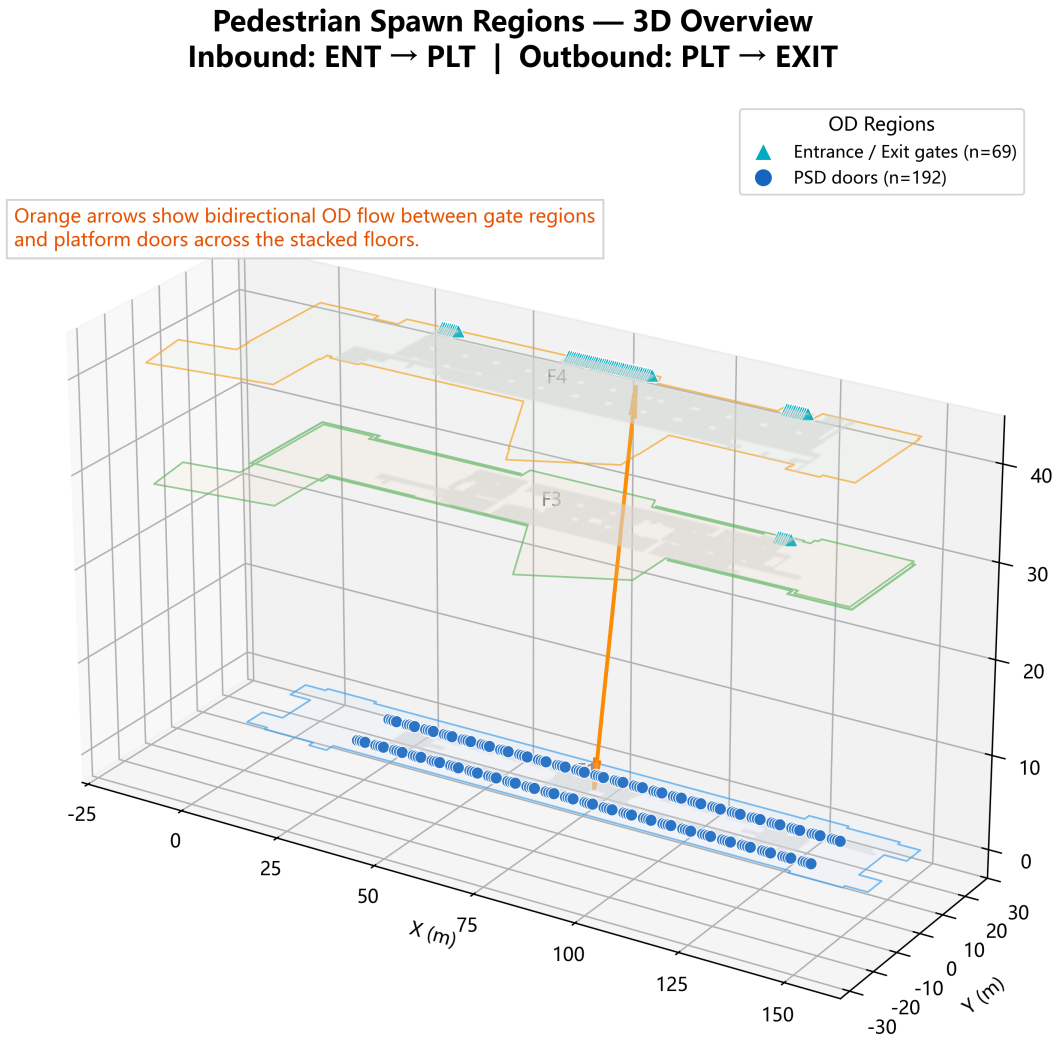


Figure 5.4.: Semantic origin–destination setup used for the station-scale experiments. Origins and destinations are represented as semantic graph regions rather than arbitrary coordinate points, allowing agents to be sampled from entrance, exit, and platform-related graph nodes.

as a calibrated estimate of actual passenger flow. Instead, it is used as a controlled 10-minute demand sample for methodological validation. The purpose is to create enough interaction, waiting, and connector queueing to test the graph-based simulation and congestion-aware replanning logic, while avoiding immediate saturation of the model.

Public reports are therefore used only to contextualize the importance of the case study. They are not used to calibrate the exact simulation demand. Such calibration would require time-resolved automatic fare-collection data, entrance-level counts, platform counts, connector queue observations, or video-derived pedestrian trajectories, which were not available for this thesis.

The demand levels used in the experiments should therefore be interpreted as controlled scenario loads. Experiments A, B, and E use 200 agents as the main comparison setting. Experiments C and D use 50 agents to isolate pedestrian-profile effects under lower demand. Experiment F uses 50, 100, 200, 300, and 500 agents to test how the graph responds to increasing demand. This demand sweep is used to examine model behaviour and robustness, not to claim an official operational capacity of the station.

Table 5.3.: Rationale for the selected agent demand levels.

Agent count	Role in the thesis	Rationale
50 agents	Quick or low-demand check	Used to verify graph connectivity, routing logic, profile implementation, and simulation stability with low computational cost.
200 agents	Main controlled demand level	Represents a controlled 10-minute scenario load, not a calibrated real passenger-flow estimate. It is large enough to produce observable waiting, queueing, and connector-load effects while remaining feasible for repeated experiments.
300–500 agents	Capacity or stress-test range	Used in the agent-load threshold experiment to examine how waiting time, queue length, not-completed agents, and runtime change under increasing scenario demand.

The experiments use several pedestrian profiles. The exact numerical parameters are kept explicit in the experiment configuration so that profile-based comparisons can be repeated. Table 5.4 summarizes the agent profiles used in the experimental design.

The main comparison experiments keep the agent configuration fixed so that the effect of routing strategy can be isolated. Profile variation is introduced only in the profile-specific experiments. This prevents the static-versus-dynamic comparison from being mixed with unrelated changes in walking speed or accessibility constraints.

Table 5.4.: Pedestrian profiles used in the experimental design.

Agent profile	Movement interpretation	Use in experiments
Normal pedestrian	Default walking behaviour and default graph-edge access.	Baseline static and dynamic routing experiments.
Elderly / slow pedestrian	Lower movement speed, representing elderly passengers or cautious pedestrians.	Single-profile comparison and mixed-agent simulation.
Accessibility-constrained pedestrian	Restricted or penalized use of selected connector types, such as stairs.	Accessibility-related profile comparison if supported by the graph.
Mixed population	Combination of the implemented normal and elderly pedestrian profiles.	Experiment D, used to test interaction between different implemented movement profiles.

5.2.3. Routing strategies

Two routing strategies are evaluated. The first is static routing. In this strategy, each agent computes a shortest path from its origin to its destination using fixed graph costs. Once assigned, the route is followed without adapting to congestion. Static routing provides the baseline condition.

The second strategy is congestion-aware replanning. In this strategy, agents may update their path during the simulation when congestion makes the current route less favourable. Congestion can be represented through local occupancy, connector load, waiting time, or downstream crowding. These signals temporarily increase the cost of congested graph elements or trigger route recomputation.

Table 5.5 compares the two routing strategies.

Table 5.5.: Routing strategies compared in the experiments.

Strategy	Route assignment	Purpose
Static routing	Initial route is computed once from fixed graph costs.	Baseline for route use, travel time, waiting, and queue formation.
Congestion-aware replanning	Route can be updated when congestion indicators exceed selected thresholds.	Tests whether adaptive routing can reduce waiting, queue pressure, or connector overload.

The same base graph is used for both strategies. Therefore, differences in the results can be attributed mainly to routing logic rather than to different spatial representations.

5.2.4. Simulation horizon, time step, and agent status definitions

The simulation is executed in discrete time. In the Chapter 6 experiments, the main simulation horizon is 600 s, corresponding to a 10-minute controlled simulation window. The internal simulation update step is $\Delta t = 0.5\,s$, while aggregate network outputs such as occupancy, queue length, connector load, and summary statistics are recorded at a coarser 5 s interval. This distinction allows the movement process to be updated at a relatively fine temporal resolution while keeping the exported result files compact enough for repeated experiments.

The 200-agent setting used in the main experiments should therefore be interpreted as a 10-minute controlled demand sample, not as a daily passenger volume or an official peak-hour demand estimate. Smaller 50-agent runs are used as low-demand or profile-isolation checks, while Experiment F varies the number of agents to test the demand-load response of the graph.

Each agent has one of the following simulation states:

- *not spawned*: the agent has been defined but its departure time has not yet been reached;
- *active*: the agent has entered the graph and is moving or waiting;
- *waiting*: the agent is active but cannot move to the next graph position because of local occupancy, connector-capacity constraints, or routing-state constraints;
- *completed*: the agent has reached its assigned destination before the end of the simulation horizon;
- *failed* or *not completed*: the agent has not reached its assigned destination before the end of the simulation horizon, or no valid route to the destination can be maintained under the current simulation state.

The term *failed agent* is used only as a simulation accounting term. It does not represent a real pedestrian accident or physical failure. In most cases, it means that the fixed simulation horizon ended before the agent reached its destination, or that the agent became locally blocked or unable to maintain a valid route under the implemented graph-based movement rules.

Table 5.6 summarizes the main temporal and demand settings used in the experiments.

5.2.5. Output recording

During each simulation run, the system records both individual-agent outputs and network-level outputs according to the temporal settings defined in Section 5.2.4.

Table 5.7 summarizes the recorded outputs.

These outputs are stored as structured files and used to generate the tables and figures reported in Chapter 6.

Table 5.6.: Simulation horizon, temporal resolution, and demand settings used in the experiments.

Experiment group	Main setting	Interpretation
Experiments A and B	200 agents, 600 s horizon	Main controlled 10-minute demand sample for comparing static and congestion-aware routing.
Experiment C	50 agents for single-profile runs; 200-agent mixed reference	Low-demand profile-isolation runs plus reference to the main controlled demand level.
Experiment D	50 agents	Prototype mixed-profile run used to test interaction between implemented pedestrian profiles.
Experiment E	200 agents, 600 s horizon	Fixed demand level used to test sensitivity to congestion-aware replanning thresholds.
Experiment F	Variable agent numbers	Demand-load test used to identify model-based saturation behaviour under increasing pedestrian demand.
Internal update step	$\Delta t = 0.5\,s$	Simulation state is updated at each time step.
Output recording interval	5 s	Aggregate network metrics are exported at regular intervals for analysis and visualization.

Table 5.7.: Outputs recorded during simulation.

Output type	Recorded data	Use in evaluation
Agent trajectory	Sequence of graph positions over time.	Route visualization and movement checking.
Agent timing	Departure time, arrival time, travel time, and waiting time.	Movement-performance evaluation.
Agent profile	Pedestrian type and profile-specific parameters.	Profile comparison and mixed-agent analysis.
Replanning log	Replan events and updated routes.	Evaluation of dynamic routing behaviour.
Network occupancy	Node, edge, or connector occupancy over time.	Congestion and queue analysis.
Connector load	Use and queueing at stairs, escalators, elevators, or gates.	Bottleneck identification.
Runtime and completion	Runtime, arrival rate, Not-completed agents, and possible deadlocks.	Robustness and capacity evaluation.

5.3. Evaluation Metrics

The evaluation metrics are grouped into four categories: graph-level metrics, movement-performance metrics, congestion and queue metrics, and robustness and capacity metrics. This grouping separates structural graph validity from simulation performance and avoids treating all outputs as equivalent.

5.3.1. Graph-level metrics

Graph-level metrics 5.8 describe whether the generated station graph is structurally suitable for routing and simulation. They are used before interpreting movement results.

Table 5.8.: Graph-level metrics.

Metric	Meaning	Interpretation
Number of nodes	Number of sampled or connector graph nodes.	Indicates graph size and movement resolution.
Number of edges	Number of horizontal and vertical movement relations.	Indicates graph connectivity.
Connected components	Number and size of graph components.	Checks whether public movement areas are connected.
Connector count	Number of stair, escalator, elevator, or gate-related connections.	Checks vertical and access-control representation.
OD reachability	Whether selected OD pairs can be connected by graph paths.	Checks route feasibility.

5.3.2. Movement-performance metrics

Movement-performance 5.9 metrics describe how efficiently agents move through the graph.

5.3.3. Congestion and queue metrics

Congestion and queue metrics 5.10 describe where and how movement becomes constrained.

In this thesis, a longer route is not automatically interpreted as worse. Under congestion-aware routing, a geometrically longer route may be preferable if it reduces waiting time, queue length, or connector overload.

5.3.4. Robustness and capacity metrics

Robustness and capacity metrics 5.11 are used in the threshold experiments. They distinguish between algorithmic sensitivity and building-capacity behaviour.

Table 5.9.: Movement-performance metrics.

Metric	Meaning	Interpretation
Arrival rate	Percentage of agents reaching their destination.	Indicates whether the simulation completes successfully.
Mean travel time	Average time from departure to arrival.	Measures overall movement efficiency.
Median and percentile travel time	Distribution-aware travel-time statistics.	Shows whether delays affect only a few agents or the majority.
Path length	Length of assigned or travelled graph path.	Measures geometric routing cost.
Profile-specific travel time	Travel time grouped by pedestrian type.	Used for pedestrian-profile comparison.

Table 5.10.: Congestion and queue metrics.

Metric	Meaning	Interpretation
Mean waiting time	Average accumulated waiting time per agent.	Measures congestion exposure.
Maximum queue length	Largest observed queue at any connector or graph location.	Indicates bottleneck severity.
Connector load	Number of agents using each vertical or access-control connector.	Identifies dominant movement corridors.
Peak occupancy	Maximum local occupancy during simulation.	Indicates local crowding pressure.
Reroute count	Number of route recalculations.	Measures intensity of dynamic adaptation.

Table 5.11.: Robustness and capacity metrics.

Metric	Meaning	Interpretation
Runtime	Computational time required for a simulation run.	Measures algorithmic feasibility.
Not-completed agents	Number of agents not reaching their destination.	Indicates simulation or capacity failure.
Deadlock count	Number of unresolved local blocking situations.	Indicates robustness limits.
Threshold response	Change in outputs when algorithmic thresholds are modified.	Measures sensitivity of replanning rules.
Demand saturation point	Demand level where waiting or failure increases sharply.	Estimates building graph capacity under the model assumptions.

5.4. Experimental Design

The experimental design consists of six experiments (see Figure 5.5). Their order follows the logic of the research questions. Experiment A first establishes whether the generated graph can support baseline static routing and simulation. Experiment B then compares fixed routing with congestion-aware replanning. Experiments C and D evaluate pedestrian heterogeneity: Experiment C tests the implemented profiles separately, while Experiment D combines the implemented normal and elderly profiles in one mixed-agent scenario. Experiments E and F test two different kinds of thresholds: algorithmic thresholds and demand-load thresholds.

This distinction between the two threshold experiments is important. Experiment E evaluates the sensitivity of the replanning algorithm by varying parameters such as waiting-time trigger, occupancy threshold, congestion weight, or replanning interval. Experiment F evaluates the response of the station graph to increasing pedestrian demand by varying the number of agents or spawn intensity. In other words, Experiment E tests the behaviour of the algorithm, while Experiment F tests the behaviour of the building graph under increasing scenario load.

Table 5.12 summarizes the relationship between the experiments and the research questions.

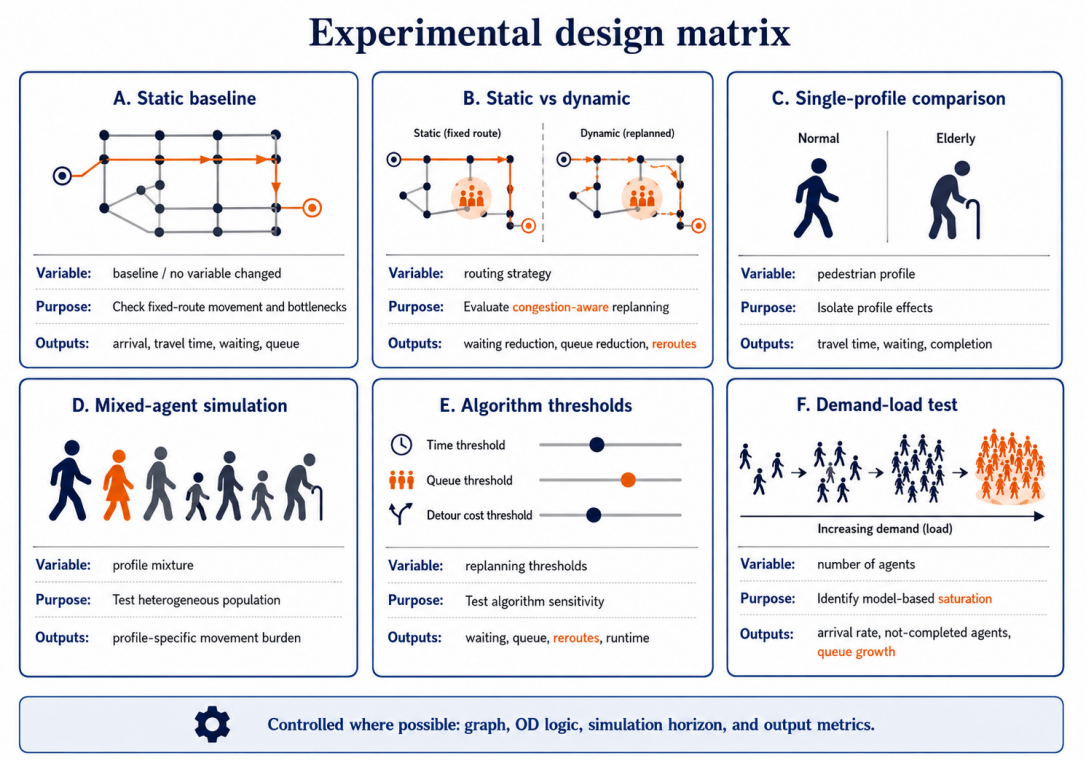


Figure 5.5.: Overview of the six experiments used in the validation. Experiments A–D evaluate routing and pedestrian-profile behaviour, while Experiments E–F test algorithmic thresholds and demand-load response.

Table 5.12.: Relationship between experiments and research questions.

Experiment	Main purpose	Related research question
A	Establish a static-routing and simulation baseline on the generated graph.	RQ1
B	Compare fixed static routing with congestion-aware replanning.	RQ3
C	Test each implemented pedestrian profile separately.	RQ4
D	Test the implemented normal and elderly pedestrian profiles together in a mixed-agent scenario.	RQ4
E	Test the sensitivity of congestion-aware replanning to algorithmic thresholds and graph abstraction.	RQ2 and RQ3
F	Test the response of the station graph to increasing controlled scenario demand.	RQ2, RQ3, and robustness extension

5.4.1. Experiment A: Baseline Static Routing and Simulation

Experiment A 5.13 establishes the baseline condition. All agents use static shortest-path routing on the generated station graph. Routes are computed once before or at departure, and no congestion-aware replanning is allowed during movement.

The purpose of this experiment is to answer a basic validation question: what movement pattern emerges when agents follow fixed shortest paths on the generated 2.5D graph? The results from this experiment provide the reference condition for all later comparisons.

Table 5.13.: Design of Experiment A.

Item	Setting
Graph input	Full station-scale 2.5D navigation graph.
Routing strategy	Static shortest-path routing.
Replanning	Disabled.
Agent profile	Default pedestrian profile unless otherwise specified.
Main outputs	Arrival rate, travel time, waiting time, queue length, connector load, and route pattern.
Purpose	Establish baseline movement behaviour and identify bottlenecks under fixed routing.

5.4.2. Experiment B: Static versus Congestion-aware Replanning

Experiment B 5.14 is the main comparative experiment for congestion-aware routing. It compares static routing with congestion-aware replanning under the same graph, OD demand,

agent number, and pedestrian profile settings. The purpose is to isolate the effect of dynamic route adaptation.

In the static condition, agents keep their initially assigned routes. In the congestion-aware condition, agents may replan when selected congestion indicators exceed predefined thresholds. These indicators may include waiting time, local occupancy, connector load, or downstream congestion.

Table 5.14.: Design of Experiment B.

Item	Setting
Compared conditions	Static routing versus congestion-aware replanning.
Controlled variables	Graph input, OD demand, number of agents, departure pattern, and pedestrian profile.
Main variable	Routing logic.
Replanning indicators	Waiting time, local occupancy, connector load, or downstream congestion.
Main outputs	Travel time, waiting time, maximum queue, connector load, route redistribution, and reroute count.
Purpose	Test whether congestion-aware replanning improves movement performance compared with fixed routing.

The expected improvement is not necessarily a shorter geometric route. A congestion-aware route may be longer but still perform better if it reduces waiting time, queue length, or connector pressure.

5.4.3. Experiment C: Single-profile Pedestrian Comparison

Experiment C 5.15 evaluates the effect of individual pedestrian profiles. Unlike Experiment D, which mixes multiple profiles in one simulation, Experiment C tests each implemented profile separately. This makes it possible to observe the direct effect of profile parameters such as walking speed or connector restrictions.

The purpose of this experiment is to determine whether different pedestrian profiles change route feasibility, travel time, waiting time, or connector use when tested under comparable conditions.

5.4.4. Experiment D: Mixed-agent Simulation with Implemented Pedestrian Profiles

Experiment D 5.16 evaluates the combined effect of all implemented pedestrian profiles in one mixed-agent simulation. This experiment is different from Experiment C. Experiment C isolates one profile at a time, while Experiment D tests whether multiple pedestrian types interact within the same movement scenario.

The mixed-agent setting is closer to a realistic station population than a single-profile simulation. It can show whether slower or accessibility-constrained agents only affect their

Table 5.15.: Design of Experiment C.

Item	Setting
Compared profiles	Each implemented pedestrian profile is tested separately.
Controlled variables	Graph input, OD demand, routing strategy, and demand level.
Main variable	Pedestrian profile parameters.
Possible profiles	Normal pedestrian, slow pedestrian, and accessibility-constrained pedestrian if implemented.
Main outputs	Arrival rate, travel time, waiting time, path length, connector use, and profile-specific bottlenecks.
Purpose	Test how individual pedestrian profiles influence movement outcomes.

own travel times, or whether they also influence global congestion, queue formation, and connector use.

Table 5.16.: Design of Experiment D.

Item	Setting
Agent composition	Combination of the implemented normal and elderly pedestrian profiles.
Controlled variables	Graph input, OD demand, routing strategy, and total demand level.
Main variable	Population composition and interaction between profiles.
Main outputs	Travel time by profile, waiting time by profile, queue exposure, connector use, arrival rate, and global congestion pattern.
Purpose	Test whether mixed pedestrian types change collective movement outcomes.

This experiment supports the pedestrian heterogeneity component of the thesis. It is not intended to fully calibrate real passenger demographics, but to test whether the graph-based simulation can represent profile-dependent movement effects.

5.4.5. Experiment E: Algorithm-threshold Sensitivity Analysis

Experiment E 5.17 tests how selected algorithmic thresholds influence congestion-aware replanning behaviour. This experiment supports RQ2 because threshold choices are part of the modelling design of the graph-based simulation, and it also supports RQ3 because these thresholds directly affect whether dynamic replanning improves movement performance.

The tested thresholds concern the decision rules used by the congestion-aware algorithm. Possible parameters include waiting-time trigger, occupancy threshold, downstream look-ahead distance, congestion-weight coefficient, connector-load threshold, and replanning interval. These parameters determine when a route is considered congested enough to justify replanning and how strongly congestion modifies temporary route cost.

Table 5.17.: Design of Experiment E.

Item	Setting
Experiment type	Algorithm-threshold sensitivity analysis.
Tested object	Congestion-aware replanning rules.
Possible variables	Waiting-time trigger, occupancy threshold, congestion weight, look-ahead distance, connector-load threshold, and replanning interval.
Controlled variables	Graph input, OD demand, agent profile, and total demand level.
Main outputs	Reroute count, waiting time, travel time, maximum queue, runtime, and route stability.
Purpose	Identify whether the replanning algorithm is stable or overly sensitive to threshold choices.

A threshold setting is considered problematic if it causes excessive replanning, unstable route switching, high runtime, or no meaningful reduction in waiting and queue pressure. The purpose is therefore not to find a universally optimal parameter value, but to examine whether the proposed replanning logic is stable within a reasonable parameter range.

5.4.6. Experiment F: Building Capacity and Agent-load Threshold Test

Experiment F 5.18 tests how the generated station graph responds to increasing pedestrian demand. This experiment is included as a robustness and stress-test extension to the main research questions. It does not estimate the official design capacity of Dadongmen Station. Instead, it identifies the demand level at which the graph-based simulation begins to show sharp increases in waiting time, queue length, connector saturation, Not-completed agents, or runtime.

This experiment is different from Experiment E. Experiment E changes algorithmic thresholds while keeping pedestrian demand fixed. Experiment F changes the number of agents or the spawn intensity while keeping the routing and threshold settings fixed. The purpose is to distinguish algorithmic sensitivity from building-load sensitivity.

The threshold identified in this experiment is interpreted as a model-based saturation point under the assumptions of the graph and simulation settings. It depends on graph resolution, connector-capacity assumptions, agent behaviour, and demand definition. It is therefore useful for comparing demand scenarios, but it should not be interpreted as a calibrated passenger-capacity assessment of the real station.

5.5. Chapter Summary

This chapter defined the experimental design used to validate the proposed workflow against the research questions. The experiments use the full station-scale 2.5D graph as their common input and evaluate it through graph-level, routing, simulation, congestion, robustness, and capacity metrics.

Table 5.18.: Design of Experiment F.

Item	Setting
Experiment type	Building capacity and agent-load threshold test.
Tested object	Station graph response under increasing pedestrian demand.
Main variable	Number of agents or spawn intensity.
Controlled variables	Graph input, routing strategy, pedestrian profile composition, and connector-capacity assumptions.
Demand levels	Low, medium, high, and stress-test demand settings.
Main outputs	Arrival rate, mean travel time, mean waiting time, maximum queue, connector saturation, Not-completed agents, deadlock count, and runtime.
Purpose	Identify the demand level at which movement performance degrades sharply.

Experiment A provides the static-routing baseline and supports the validation of the generated graph. Experiment B directly addresses the congestion-aware replanning question by comparing fixed routing with dynamic route adaptation. Experiments C and D address pedestrian heterogeneity by testing profiles separately and in a mixed-agent scenario. Experiment E evaluates the sensitivity of the replanning algorithm to threshold settings, while Experiment F tests the response of the station graph to increasing pedestrian demand. The next chapter reports and discusses the results in the same order.

6. Results and Discussion

This chapter presents and discusses the results of the proposed IFC-to-2.5D graph workflow and the associated pedestrian simulation experiments. The structure follows the validation logic defined in Chapter 5. First, the chapter reports the outputs of the IFC-to-graph pipeline and the resulting station-scale 2.5D navigation graph. It then presents the experimental results for static routing, congestion-aware replanning, pedestrian-profile comparison, mixed-agent simulation, algorithm-threshold sensitivity, and building-capacity testing.

The purpose of this chapter is not only to report numerical outputs, but also to interpret what these outputs indicate about the usability of the generated graph. The results are therefore discussed in relation to the research questions introduced in Chapter 1: whether IFC BIM can be transformed into a traffic-relevant indoor navigation graph, whether graph and semantic modelling choices affect simulation usability, whether congestion-aware replanning improves movement performance, and how pedestrian heterogeneity influences travel time, waiting, and congestion.

The results should be interpreted as controlled methodological validation rather than as calibrated operational prediction. The simulation is not calibrated against observed passenger trajectories, and the raw station BIM data are not publicly redistributed. Therefore, the numerical results are used to evaluate the behaviour of the proposed workflow under controlled assumptions, not to claim official passenger-flow capacity or real-time operational performance for the station.

6.1. Outputs of the IFC-to-Graph Pipeline

The first part of the result analysis concerns the intermediate outputs of the IFC-to-graph pipeline. These outputs are important because the proposed method is not a direct conversion from IFC geometry to a navigation graph. Instead, the workflow proceeds through a sequence of explicit processing steps, including IFC loading, geometry extraction, walkable-space sampling, 2.5D graph construction, semantic OD definition, simulation, and evaluation.

The repository audit confirms that the pipeline is organized as a seven-step workflow. The main scripts are `step0_load.py`, `step1_geometry.py`, `step2_sampling.py`, `step3_graph.py`, `step4_routing.py`, `step5_experiments.py`, and `step6_evaluate.py`. This modular structure makes the workflow inspectable: each stage produces intermediate outputs that can be checked before the next stage is executed.

For the final station case, the pipeline produced a station-scale 2.5D graph, semantic OD regions, and simulation-ready agent samples. The graph was exported as `navigation_graph.gpickle`, while node and edge layers were also exported as GeoJSON files by edge type, including floor, stair, escalator, elevator, fare-gate, platform-door, entrance, and anchor-snap edges.

These outputs indicate that the final graph is not a generic geometric network, but a movement-oriented representation with explicit semantic and connector structure.

Table 6.1 summarizes the main outputs used in the Chapter 6 evaluation. Figure 6.1 provides a visual overview of the same pipeline stages.

Table 6.1.: Main outputs of the IFC-to-graph pipeline used for the Chapter 6 evaluation.

Output category	Value	Role in the workflow
Final graph file	`navigation_graph.gpickle`	NetworkX graph used as the substrate for routing and simulation.
Node export	`nodes_all.geojson`	Spatial representation of all graph nodes with level and type attributes.
Edge exports	`edges_*.geojson`	Edge layers separated by type, including floor, connector, entrance, fare-gate, and platform-door edges.
Public movement levels	F1, F3, F4	Floor-based layers used for ordinary pedestrian movement.
Connector elements	4 stair chains, 6 escalator units, 1 elevator shaft	Explicit inter-level movement elements in the 2.5D graph.
Semantic OD regions	Entrance: 69 nodes; Exit: 69 nodes; Platform: 192 nodes	Origin–destination regions used for route assignment and simulation.
Sampled agents	200 agents	Main simulation demand used in the station-scale baseline and dynamic comparison.
OD flow split	90 entrance-to-platform; 110 platform-to-exit	Directional passenger-flow assumptions used for the main simulation scenarios.

The key result of this stage is that the station model was transformed into an auditable graph-processing pipeline. The intermediate outputs allow errors in semantic filtering, connector representation, OD definition, or graph construction to be detected before simulation. This is important because the later experimental results depend on the structural correctness of the generated graph.

6.2. Resulting 2.5D Navigation Graph

The final output of the graph-construction stage is a station-scale 2.5D navigation graph. The graph represents public pedestrian movement on floor-based layers and preserves vertical circulation through explicit connector edges. In the final default graph, the public movement structure is represented on the Platform level (F1), Concourse level (F3), and Transport level (F4), with additional connector nodes for stairs, escalators, and elevators.

The resulting default graph contains 18,454 nodes and 134,852 edges. Most nodes are floor nodes, but the graph also contains semantic and connector-related nodes, including entrance

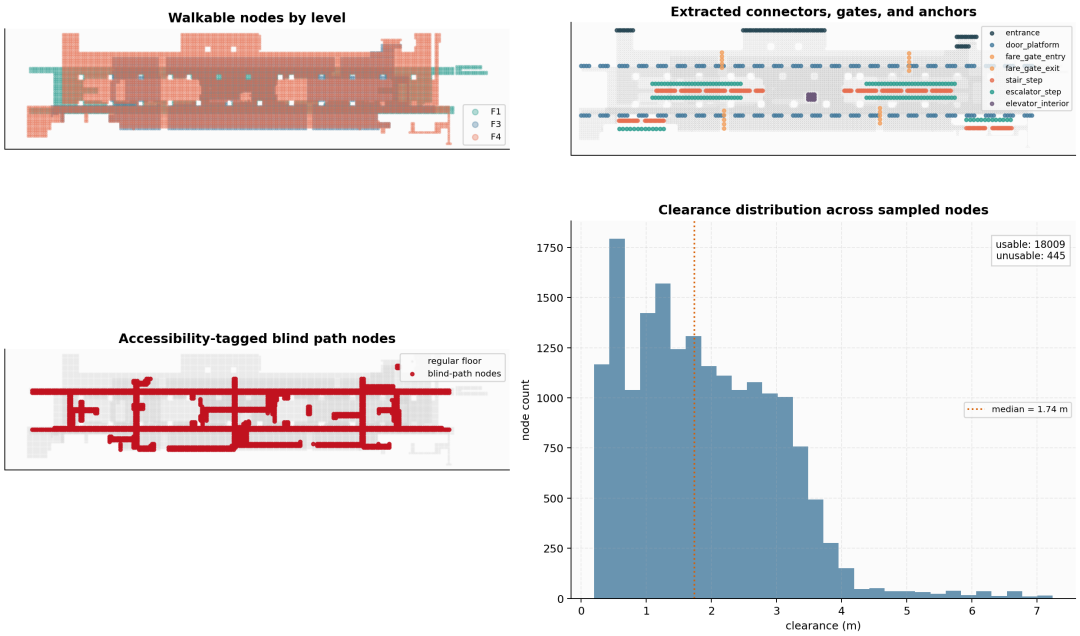


Figure 6.1.: Main intermediate outputs of the station-scale IFC-to-graph pipeline. The workflow converts construction-oriented BIM information into movement-relevant graph layers, connector representations, and semantic OD regions before routing and simulation are performed.

nodes, platform-door nodes, stair-step nodes, escalator-step nodes, elevator nodes, and fare-gate nodes. The edge set is similarly differentiated, with floor edges forming the majority of local movement relations and connector edges representing stairs, escalators, elevators, fare gates, entrances, platform doors, and anchor-snap links.

Table 6.2 summarizes the node distribution by level, while Table 6.3 summarizes the main edge types. Figure 6.2 provides an overall multi-level view of the generated 2.5D station graph and illustrates how the public floor layers are connected by escalators, elevators, and stairs. Figures 6.3 and 6.4 then provide enlarged level-specific views of the default graph and the sparse semantic-light graph variant. Figure 6.5 summarizes the main graph statistics.

Table 6.2.: Node statistics of the resulting default 2.5D navigation graph.

Graph component	Nodes	Interpretation
Platform level (F1)	4,001	Lower public movement layer associated with platform-side movement.
Concourse level (F3)	5,403	Main distribution and access-control movement layer.
Transport level (F4)	8,714	Upper public circulation and entrance/exit movement layer.
Stair nodes	190	Explicit stair-step and stair-chain representation.
Escalator nodes	146	Explicit escalator-step and escalator-chain representation.
Total	18,454	Final retained station-scale default graph.

Table 6.3.: Main edge types in the resulting default 2.5D navigation graph.

Edge type	Edges	Interpretation
Floor	132,614	Horizontal movement relations on public levels.
Anchor-snap	384	Links between semantic anchors and graph nodes.
Platform-door	384	Platform-side access or transition edges.
Stair	388	Stair-based vertical movement edges.
Escalator	304	Escalator-based vertical movement edges.
Elevator and elevator-related	600	Elevator-door, elevator-interior, and elevator movement relations.
Fare gate	40	Access-control transition edges.
Entrance	138	Entrance-related graph connections.
Total	134,852	Complete edge set used for routing and simulation.

The default 0.5 m grid is used as the main graph resolution. The sparse semantic-light

graph, based on a 0.75 m grid, is used as a controlled abstraction variant. The purpose is not to identify a universally optimal resolution, but to test whether reducing graph density and semantic support changes routing and simulation behaviour.

Table 6.4.: Default graph and sparse semantic-light graph variant used in the graph-design sensitivity analysis.

Graph setting	Nodes	Edges	Design interpretation
Default graph	18,454	134,852	0.5 m grid, 8-neighbour floor links, explicit entrance/platform semantics, and blind-path stitching enabled.
Sparse semantic-light graph	8,639	31,672	0.75 m grid, 4-neighbour floor links, no blind-path stitching, and entrance/platform semantics reduced to boundary fallback regions.

Figure 6.2 illustrates the overall 2.5D character of the representation. Horizontal movement is modelled on floor-based graph layers, while vertical movement is constrained to explicit connector elements. This distinguishes the graph from a full 3D free-space model: agents do not move freely through the volume of the station, but move across walkable layers and between levels only through stairs, escalators, and elevators.

The enlarged views in Figure 6.3 show the spatial distribution of the default graph and the sparse graph on each public movement level. These views are included because the complete graph is dense, and a compact overview alone makes it difficult to inspect the local structure of the generated graph.

Figure 6.4 shows the corresponding views for the sparse semantic-light graph. Compared with the default graph, this variant has fewer graph nodes, fewer local floor connections, and smaller semantic OD regions. The comparison visually supports the later graph-variant experiments by showing that the sparse graph is not merely a smaller file, but a different movement substrate.

The default graph is fully connected according to the graph summary, meaning that the retained public movement structure forms a single connected component. This is a necessary condition for simulation because agents must be able to move between semantic origins and destinations without encountering disconnected graph islands. The semantic OD regions also have sufficient node support in the default graph: the entrance and exit regions each contain 69 graph nodes, while the platform region contains 192 graph nodes. These regions provide a basis for sampling agents and assigning route directions.

Overall, the graph statistics indicate that the workflow produces more than a simple abstract topology. The default graph preserves floor-based movement, semantic access elements, explicit vertical connectors, and sufficiently large OD regions for station-scale simulation. The sparse graph remains connected and routable, but it deliberately reduces graph density and semantic support. This contrast provides the basis for testing whether graph construction choices affect simulated movement performance in Experiments E and F.

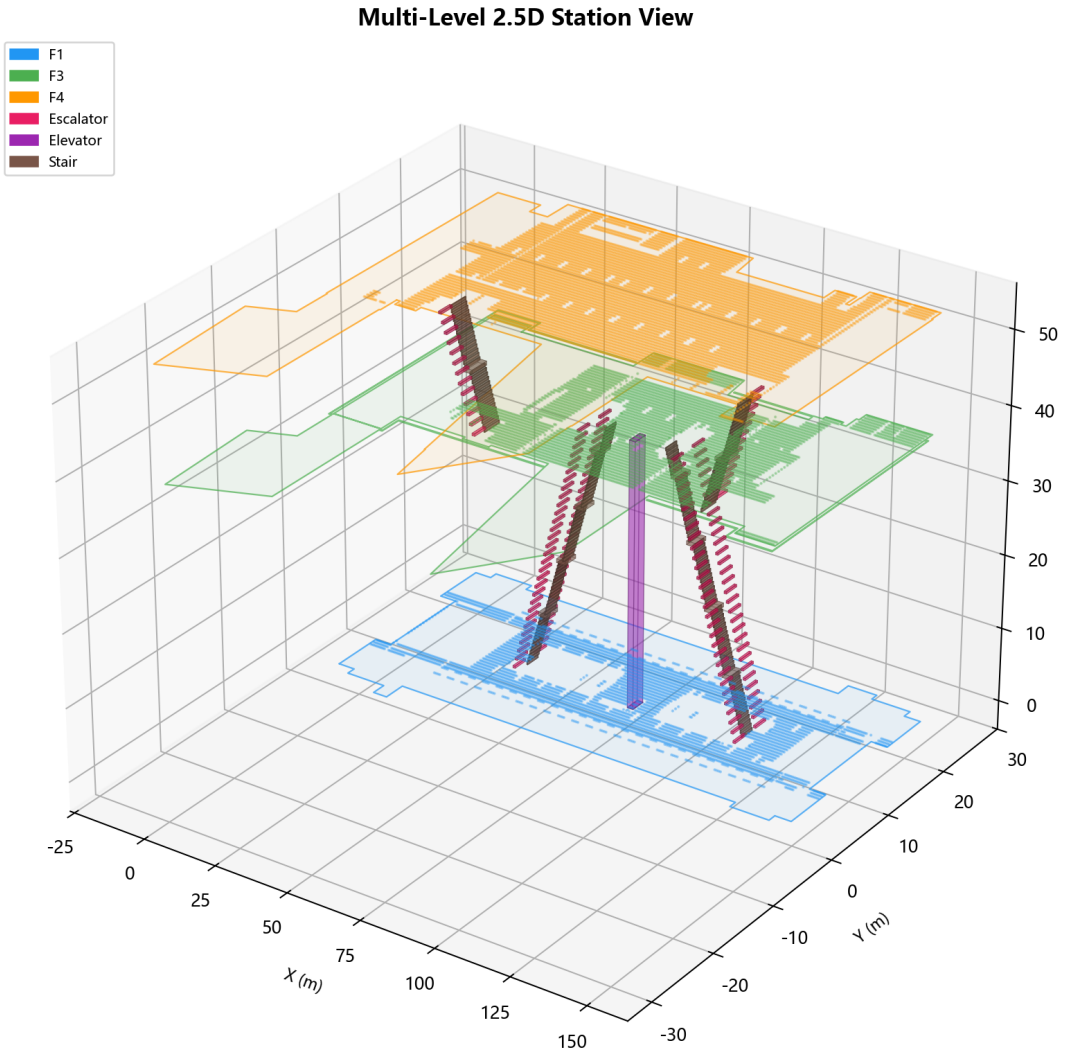
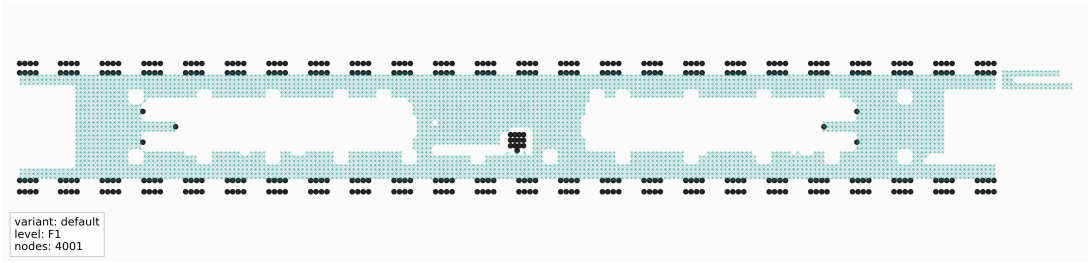
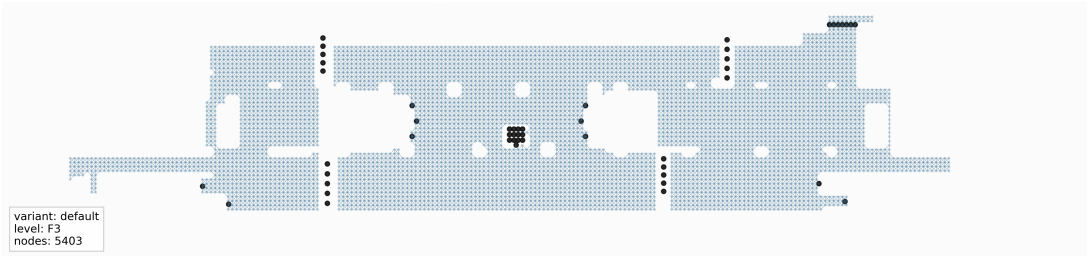


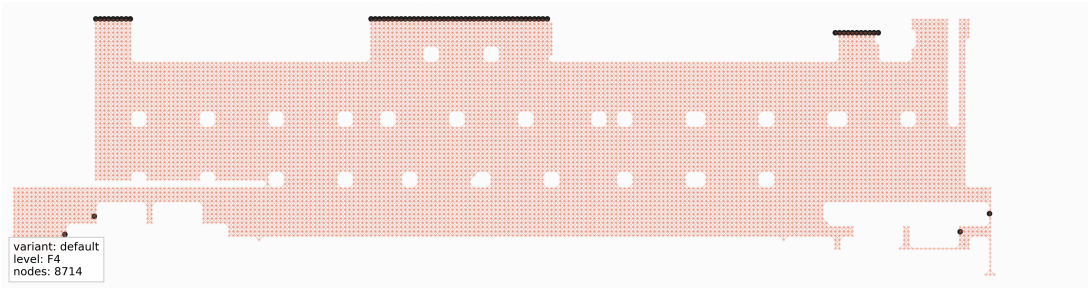
Figure 6.2.: Multi-level 2.5D station view of the generated navigation graph. The figure shows the three public movement levels (F1, F3, and F4) together with the main vertical connector types, including escalators, elevators, and stairs. It illustrates the overall 2.5D representation used for routing and pedestrian simulation.



(a) F1 / Platform level

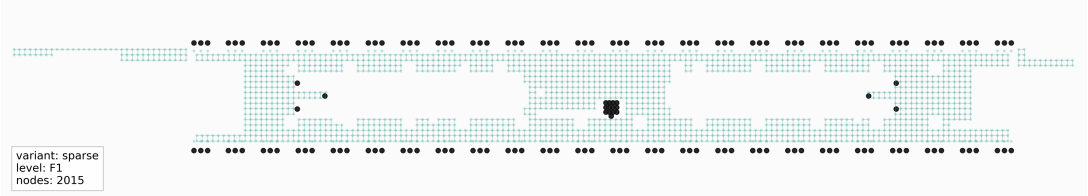


(b) F3 / Concourse level

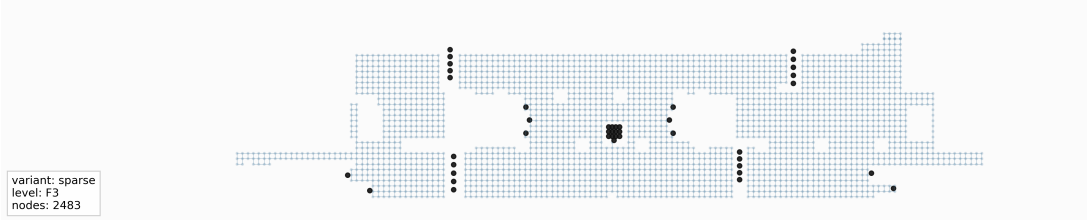


(c) F4 / Transport level

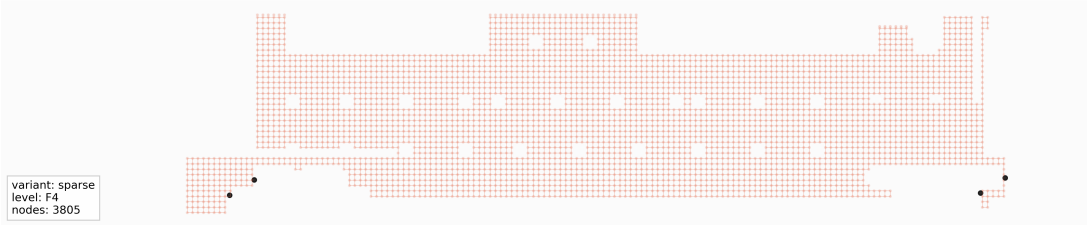
Figure 6.3.: Enlarged level-specific views of the default station-scale 2.5D navigation graph. The views show the spatial distribution of graph nodes, floor-level connections, and connector-related graph structures on the three public movement levels.



(a) F1 / Platform level



(b) F3 / Concourse level



(c) F4 / Transport level

Figure 6.4.: Enlarged level-specific views of the sparse semantic-light graph variant. Compared with the default graph, this variant is coarser and has weaker semantic OD support, which is later reflected in the graph-variant simulation results.

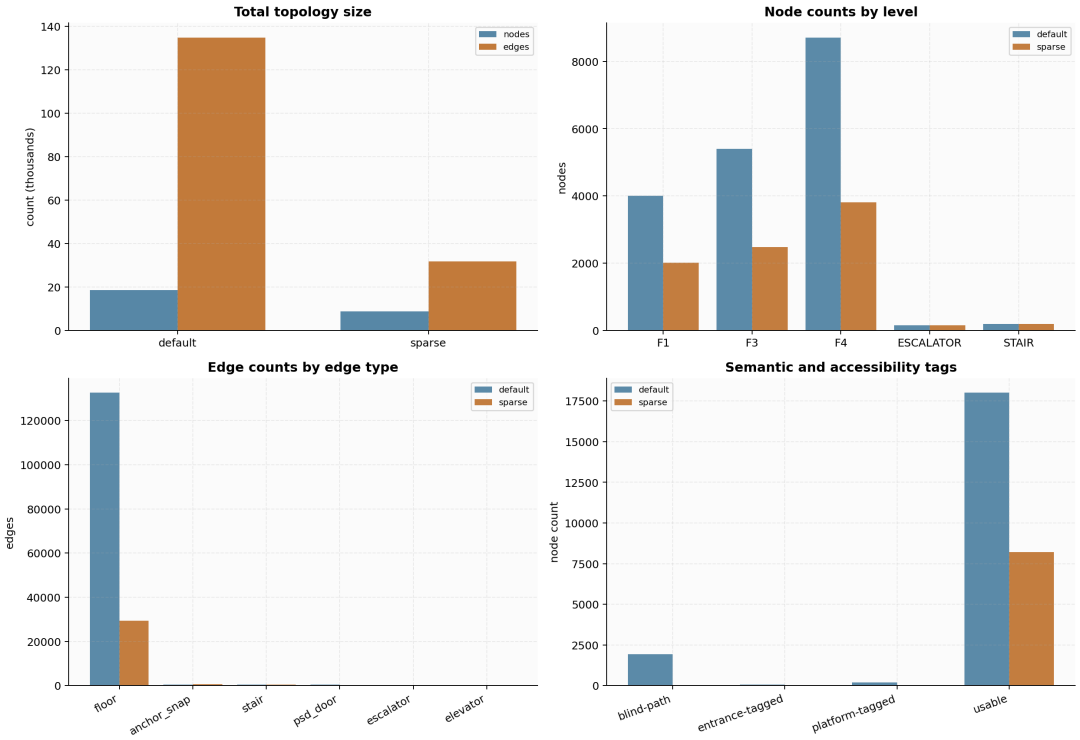


Figure 6.5.: Graph statistics of the resulting 2.5D navigation graph, including node distribution by level, edge-type distribution, and summary graph properties.

6.3. Experimental Results

This section presents the experimental results in the order defined in Chapter 5. The first experiment establishes a static-routing baseline. The second compares static routing with congestion-aware replanning. The third evaluates single-profile pedestrian behaviour. The fourth tests a mixed-agent scenario using the pedestrian profiles implemented in the current simulation prototype.

The results should be interpreted as controlled simulation outputs rather than calibrated operational predictions. The experiments use the generated station graph and the implemented agent-based movement logic to test whether the graph can support interpretable routing, waiting, queueing, replanning, and profile-sensitive movement analysis.

To avoid ambiguity in the interpretation of the experiment tables, several terms are used consistently throughout this chapter. *Completed agents* are agents that reach their assigned destination before the end of the simulation horizon. *Not-completed agents* are agents that do not reach their assigned destination within the fixed simulation horizon, or agents for which no valid route can be maintained under the implemented graph-based movement rules. In some exported files, this category may appear as "failed", but in the thesis it is interpreted only as not-completed within the simulation horizon. It is a simulation accounting category, not a real-world failure event.

Arrival rate is the ratio of completed agents to spawned agents. *Mean travel time* is computed for completed agents and measures the time between departure and arrival. *Mean waiting time* is the accumulated time during which agents are active but unable to move because of occupancy, connector capacity, or routing-state constraints. *Maximum queue* reports the largest observed queue at a connector or constrained graph location. *Reroute events* count the number of path recalculations triggered under congestion-aware routing. *Runtime* refers to computational runtime and should not be confused with the simulated 600 s movement horizon.

6.3.1. Experiment A: Baseline Static Routing and Simulation

Experiment A establishes the static-routing baseline for the station-scale default graph. In this experiment, 200 agents are simulated over a 600 s horizon using static shortest-path routing. This corresponds to the main controlled 10-minute demand sample defined in Chapter 5. Routes are assigned using the initial graph costs and are not updated during the simulation. The purpose of this experiment is to identify the movement pattern produced by fixed shortest-path assignment before congestion-aware replanning is introduced.

The baseline result is summarized in Table 6.5. Out of 200 agents, 195 completed their trips and 5 did not reach their assigned destination within the fixed simulation horizon or could not maintain a valid route under the implemented movement rules. The resulting arrival rate is 97.5%. The mean travel time is 185.47 s, while the 95th percentile travel time reaches 280.05 s. This indicates that the generated graph supports station-scale OD movement, but that part of the simulated population experiences substantially longer trips.

Waiting time is a major component of the baseline movement pattern. The mean waiting time is 68.54 s and the maximum waiting time reaches 321.0 s. The peak connector queue reaches 53 agents. The dominant bottleneck is the connector `stair_881419_F3_F4_r0_s0`,

with 139 agent crossings. This confirms that fixed shortest-path routing concentrates a large share of the demand on a limited number of vertical connectors.

Table 6.5.: Results of Experiment A: baseline static routing and simulation.

Metric	Value	Interpretation
Routing mode	Static	Fixed shortest-path routing.
Number of agents	200	Main controlled 10-minute demand sample.
Completed agents	195	Agents reaching their assigned destinations.
Not-completed agents	5	Agents not reaching their destination within the fixed simulation horizon or unable to maintain a valid route.
Arrival rate	97.5%	High but not complete trip completion.
Mean travel time	185.47 s	Average trip duration for completed agents.
Median travel time	188.74 s	Typical trip duration.
95th percentile travel time	280.05 s	Delay experienced by the slowest completed agents.
Mean waiting time	68.54 s	Average accumulated waiting burden.
Median waiting time	56.50 s	Typical waiting burden.
Maximum waiting time	321.0 s	Worst individual waiting burden.
Peak connector queue	53 agents	Strong local bottleneck formation.
Dominant bottleneck connector	`stair_881419_F3_F4_r0_s0`	Main overloaded vertical connector.
Bottleneck throughput	139 crossings	Number of crossings at the dominant connector.
Total replanning events	0	No dynamic route adaptation.

Figure 6.6 visualizes the baseline congestion pattern. The static-routing condition produces route concentration around the main connector between the Concourse and Transport levels. Figure 6.7 further identifies the connector-level concentration behind the aggregate waiting and queue metrics.

The baseline experiment supports two conclusions. First, the generated default 2.5D graph is usable for station-scale OD movement because most agents successfully reach their destinations. Second, static shortest-path routing creates clear bottleneck effects, which provides a meaningful reference for evaluating congestion-aware replanning in Experiment B.

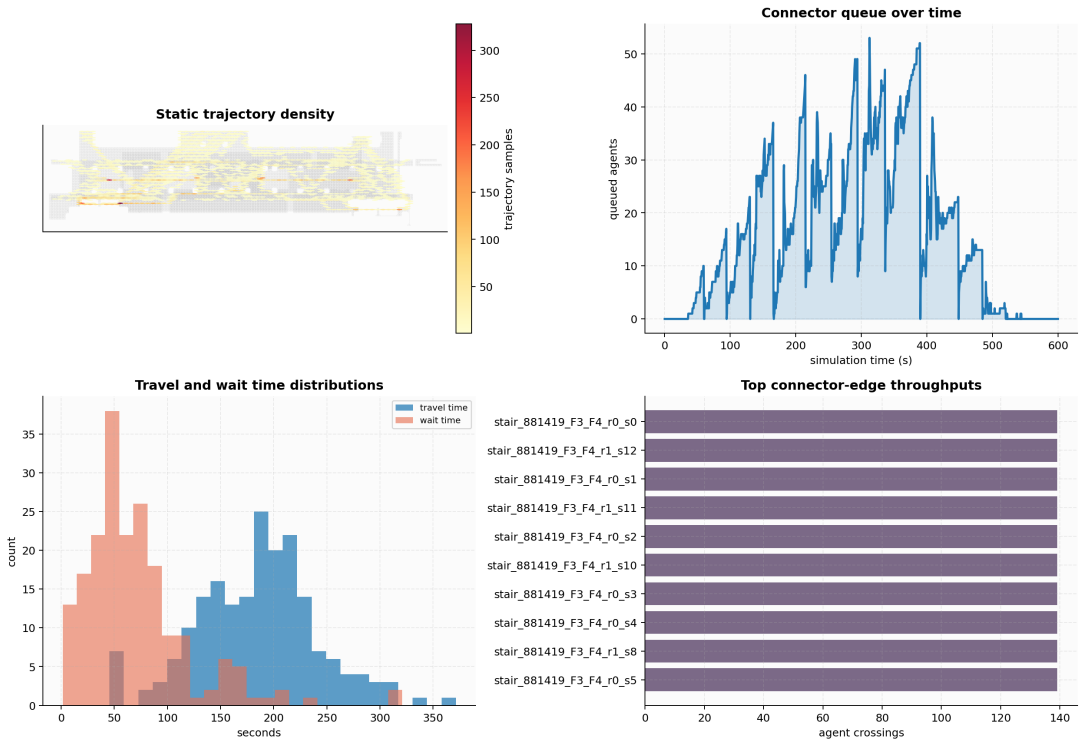


Figure 6.6.: Baseline static-routing result. Fixed shortest-path assignment concentrates movement on the most attractive connector corridors, producing visible queueing around the dominant vertical connector.

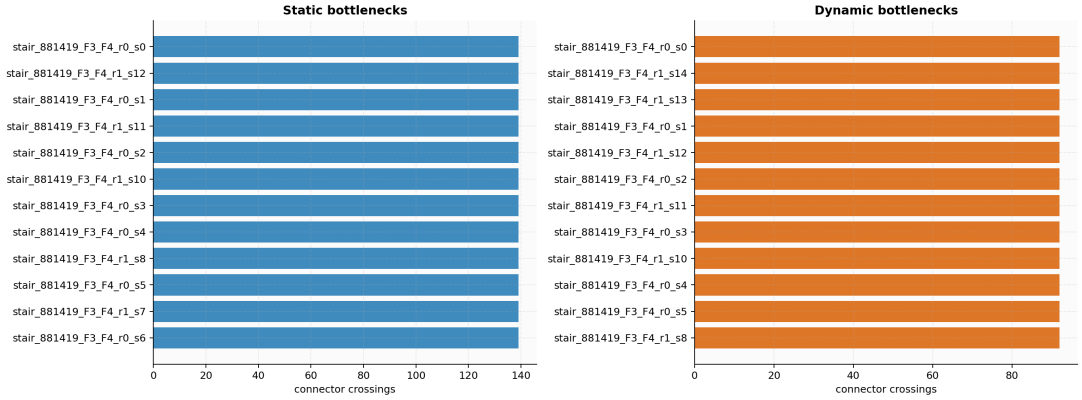


Figure 6.7.: Main bottleneck connectors under the baseline and replanning scenarios. The figure shows which connectors accumulate the highest number of crossings and helps explain why fixed shortest-path routing produces local queue concentration.

6.3.2. Experiment B: Static versus Congestion-aware Replanning

Experiment B compares the static baseline with congestion-aware replanning under the same graph and agent demand. Both conditions use the default graph, 200 agents, and the same 600 s simulation horizon. The static condition keeps the initial route fixed, while the dynamic condition allows route updates when congestion indicators trigger replanning. This experiment directly evaluates whether congestion-aware replanning changes movement performance compared with fixed routing.

Table 6.6 summarizes the comparison. The dynamic condition slightly improves the arrival rate from 97.5% to 98.0%, reducing the number of not-completed agents from 5 to 4. However, mean travel time increases from 185.47 s to 197.84 s, and the 95th percentile travel time increases from 280.05 s to 311.72 s. This means that dynamic replanning does not minimize total travel time under the tested setting.

The main improvement appears in waiting and queue metrics. Mean waiting time decreases from 68.54 s to 52.37 s, a reduction of 23.6%. Median waiting time decreases from 56.5 s to 36.5 s, a reduction of 35.4%. Maximum waiting time decreases from 321.0 s to 276.5 s. The peak connector queue decreases from 53 to 44 agents. These changes show that congestion-aware replanning reduces queueing pressure and waiting burden, even though it increases average route duration.

Table 6.6.: Comparison of Experiment B: static routing versus congestion-aware replanning.

Metric	Static	Dynamic	Change
Number of agents	200	200	–
Completed agents	195	196	+1
Not-completed agents	5	4	-1
Arrival rate	97.5%	98.0%	+0.5 pp
Mean travel time	185.47 s	197.84 s	+6.7%
Median travel time	188.74 s	192.94 s	+2.2%
95th percentile travel time	280.05 s	311.72 s	+11.3%
Mean waiting time	68.54 s	52.37 s	-23.6%
Median waiting time	56.5 s	36.5 s	-35.4%
Maximum waiting time	321.0 s	276.5 s	-13.9%
Peak connector queue	53	44	-17.0%
Reroute events	0	1,996	+1,996

The result does not show a decrease in mean travel time. This requires careful interpretation. If the routing strategy were designed as a global travel-time optimizer, a reduction in mean travel time would be expected. However, the congestion-aware strategy implemented in this thesis does not directly minimize total travel time. Instead, it modifies route choice based on local congestion indicators and penalizes crowded graph elements.

In the station graph, avoiding a congested connector often means choosing a longer alternative path or using a less direct vertical movement sequence. As a result, agents may spend less time waiting in queues but more time walking or rerouting through alternative graph sections. For this reason, the dynamic strategy should be interpreted as a queue-management mechanism rather than as a shortest-time optimizer.

Its benefit is therefore visible mainly in the waiting and queue metrics: mean waiting time, median waiting time, maximum waiting time, and peak connector queue all decrease under dynamic routing. However, this improvement comes with a trade-off in mean and high-percentile travel time. The result is not that dynamic routing is universally faster, but that it redistributes demand and reduces local congestion pressure.

Figure 6.8 compares the main performance metrics, while Figure 6.9 shows how route use changes between static and dynamic routing. The dynamic run generates 1,996 replanning events, which indicates that agents actively adapt to local congestion. However, this adaptation comes at a cost: the dynamic condition trades shorter waiting and smaller queues for longer travel times and more route changes.

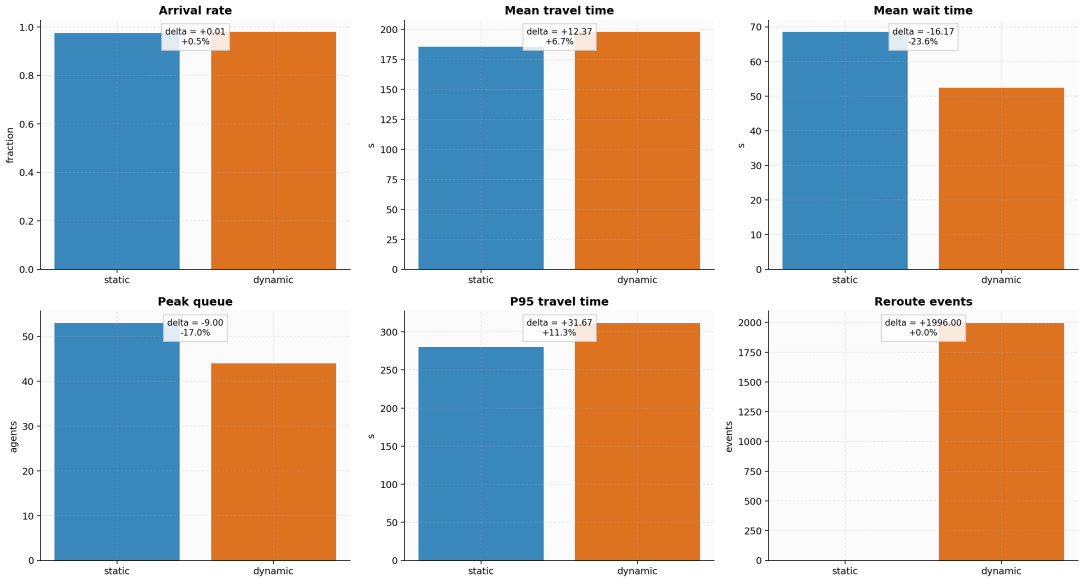


Figure 6.8.: Metric comparison between static routing and congestion-aware replanning. Dynamic replanning reduces waiting time and peak queue length, but increases mean and 95th percentile travel time under the tested setting.

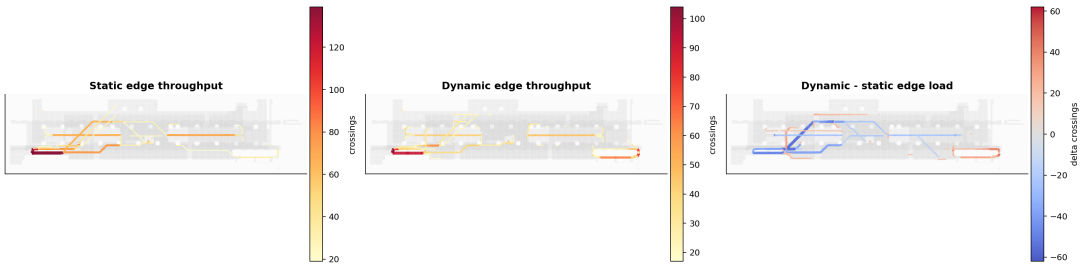


Figure 6.9.: Route-use comparison between static routing and congestion-aware replanning. Dynamic routing redistributes part of the flow away from the strongest bottleneck, reducing queue concentration at the cost of additional route changes.

Overall, Experiment B shows a trade-off between queue-related performance and travel-time

performance. Congestion-aware replanning reduces waiting time and local queue severity, but it does not necessarily reduce mean travel time. This is because the implemented dynamic routing logic avoids crowded connectors rather than solving a global system-optimal assignment problem. For station environments, this distinction is important: reducing severe local queues may be valuable even when some agents take longer alternative routes.

6.3.3. Experiment C: Single-profile Pedestrian Comparison

Experiment C evaluates the effect of pedestrian profile on movement performance. Two implemented profiles are available in the current simulation prototype: normal pedestrians and elderly pedestrians. Normal pedestrians use a speed factor of approximately 0.95–1.05 times the base walking speed, while elderly pedestrians use a lower speed factor of approximately 0.55–0.75 times the base walking speed.

The experiment includes normal-only and elderly-only runs with 50 agents each, and a 200-agent mixed reference scenario. Because the single-profile runs and the mixed reference use different agent counts, their results should not be interpreted as a strict capacity comparison. Instead, the purpose is to identify whether profile assumptions lead to systematic differences in travel time, waiting time, and completion rate under otherwise comparable graph-based movement logic.

Table 6.7 reports the profile comparison. In the static normal-only run, all 50 agents complete their trips, with a mean travel time of 135.91 s and a mean waiting time of 12.28 s. In the static elderly-only run, only 45 of 50 agents complete their trips, giving an arrival rate of 90.0%. The mean travel time increases to 238.56 s, and the mean waiting time increases to 126.99 s. The peak connector queue also doubles from 7 to 14 agents.

Dynamic replanning improves the normal-only waiting result, reducing mean waiting time from 12.28 s to 7.06 s. However, it does not improve the elderly-only run in the current output: the elderly-only static and dynamic results remain identical, with zero replanning events reported for the elderly-only dynamic condition. This suggests that the dynamic replanning trigger was not activated for that profile under the specific low-demand run, or that the available alternative routes did not produce a different route choice.

Table 6.7.: Results of Experiment C: single-profile pedestrian comparison.

Metric	Normal static	Normal dynamic	Elderly static	Elderly dynamic	Mixed static	Mixed dynamic
Number of agents	50	50	50	50	200	200
Completed agents	50	49	45	45	195	196
Not-completed agents	0	1	5	5	5	4
Arrival rate	100.0%	98.0%	90.0%	90.0%	97.5%	98.0%
Mean travel time (s)	135.91	140.68	238.56	238.56	185.47	197.84
Median travel time (s)	145.86	142.52	243.34	243.34	188.74	192.94
P95 travel time (s)	172.85	214.92	333.24	333.24	280.05	311.72
Mean waiting time (s)	12.28	7.06	126.99	126.99	68.54	52.37
Maximum waiting time (s)	28.0	22.5	278.5	278.5	321.0	276.5
Peak connector queue	7	7	14	14	53	44
Reroute events	0	99	0	0	0	1,996
Top bottleneck connector	stair	stair	stair	stair	stair	stair
Bottleneck crossings	36	28	36	36	139	92

Figure 6.10 visualizes the profile-based differences. The dominant pattern is clear: elderly agents experience substantially higher travel times and waiting burdens than normal agents.

This confirms that the simulation framework is sensitive to pedestrian-profile assumptions. It also shows that aggregate metrics can hide important distributional differences between pedestrian groups.

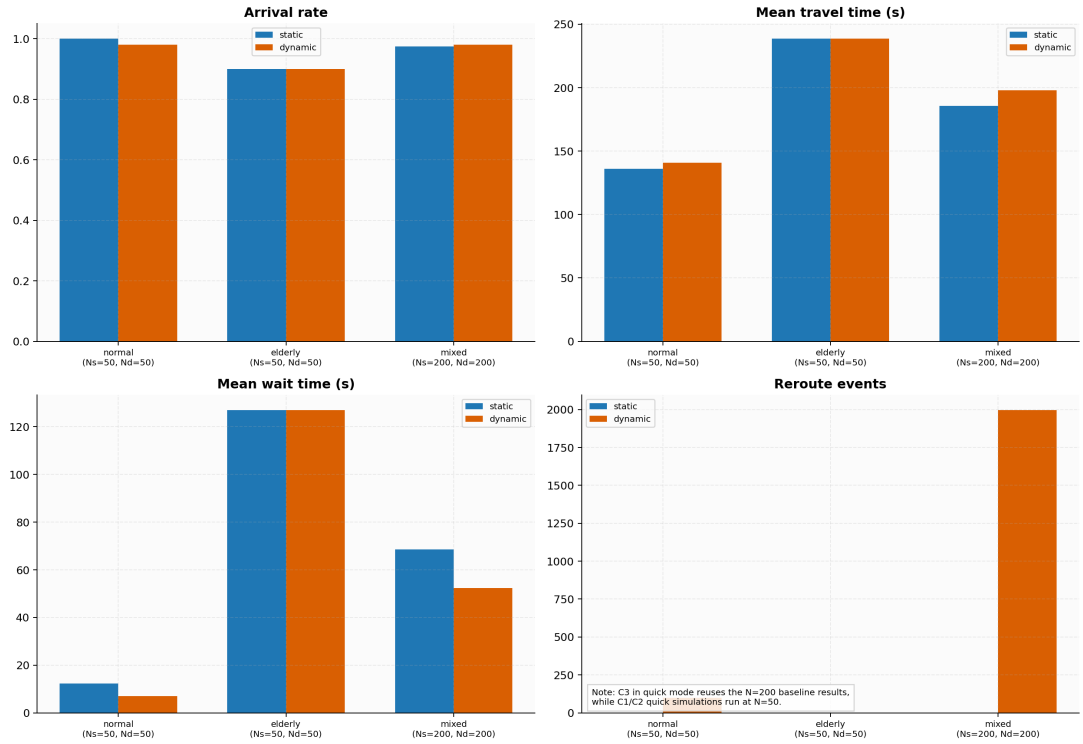


Figure 6.10.: Single-profile pedestrian comparison. Elderly pedestrians experience substantially higher travel times and waiting times than normal pedestrians, indicating that the graph-based simulation is sensitive to profile-specific movement assumptions.

Experiment C supports the fourth research question. It shows that heterogeneous pedestrian assumptions affect not only individual speed but also waiting exposure and completion rate. However, the current profile implementation is limited to normal and elderly pedestrians. Accessibility-constrained or wheelchair profiles were planned but were not fully completed in the current implementation.

6.3.4. Experiment D: Mixed-agent Simulation with Implemented Pedestrian Profiles

Experiment D evaluates a mixed-agent scenario using all pedestrian profiles implemented in the current simulation prototype. Although the experimental design originally allowed for a broader set of pedestrian types, the completed implementation supports two profiles: normal pedestrians and elderly pedestrians. Therefore, the reported mixed-agent experiment should be understood as a mixed normal–elderly simulation rather than a complete accessibility-population model.

The mixed-agent run was executed with 50 agents as a low-demand prototype test of profile interaction. In the static condition, the population consisted of 36 normal pedestrians and 14 elderly pedestrians, corresponding to shares of 72% and 28%, respectively. The same composition was used in the dynamic condition. Because this experiment uses a smaller number of agents than the 200-agent baseline, it should not be interpreted as a station-capacity result. Instead, it tests whether profile-specific movement burdens remain visible when different implemented profiles are simulated together.

Table 6.8 reports both overall and profile-specific results. In the static mixed-agent run, all 50 agents completed their trips, with a mean travel time of 166.20 s and a mean waiting time of 42.87 s. In the dynamic run, all 50 agents also completed their trips. Mean travel time decreased slightly to 163.74 s, while mean waiting time decreased from 42.87 s to 31.95 s. The peak connector queue decreased from 9 to 8 agents, and 141 replanning events were recorded.

At the profile level, elderly pedestrians experienced substantially higher movement burdens than normal pedestrians. In the static run, normal pedestrians had a mean travel time of 149.48 s and a mean waiting time of 23.38 s, while elderly pedestrians had a mean travel time of 209.21 s and a mean waiting time of 93.00 s. In the dynamic run, elderly mean travel time decreased to 199.13 s and mean waiting time decreased to 82.71 s. Normal mean waiting time also decreased from 23.38 s to 12.21 s.

Table 6.8.: Results of Experiment D: mixed-agent simulation with implemented pedestrian profiles.

Metric	Overall static	Overall dynamic	Normal static	Normal dynamic	Elderly static	Elderly dynamic
Number of agents	50	50	36	36	14	14
Profile share	100%	100%	72%	72%	28%	28%
Completed agents	50	50	36	36	14	14
Not-completed agents	0	0	0	0	0	0
Arrival rate	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Mean travel time (s)	166.20	163.74	149.48	149.98	209.21	199.13
Median travel time (s)	162.35	153.10	158.24	141.91	215.64	213.41
P95 travel time (s)	252.43	254.47	189.20	243.72	322.37	279.67
Mean waiting time (s)	42.87	31.95	23.38	12.21	93.00	82.71
Maximum waiting time (s)	203.5	139.5	60.5	28.0	203.5	139.5
Peak connector queue	9	8	N/A	N/A	N/A	N/A
Reroute events	0	141	–	–	–	–
Simulation runtime (s)	3.98	486.79	–	–	–	–

Figure 6.11 summarizes the mixed-agent result. The figure shows that the dynamic condition reduced waiting time for both profiles, but the elderly profile remained more heavily affected. This is an important result: even when all agents arrive, the burden of congestion is not evenly distributed across pedestrian types.

The main conclusion from Experiment D is that mixed-agent simulation can reveal profile-dependent movement burdens that are not visible in aggregate completion rates. Both profiles achieved 100% arrival, but elderly pedestrians experienced much higher waiting and travel-time burdens. The experiment therefore supports the use of profile-specific metrics in addition to overall travel-time and arrival-rate metrics.

At the same time, the result has two limitations. First, the mixed-agent run uses only 50 agents, so it should not be interpreted as a high-demand station-scale capacity result. Second, queue exposure and connector usage by profile are not available in the current output

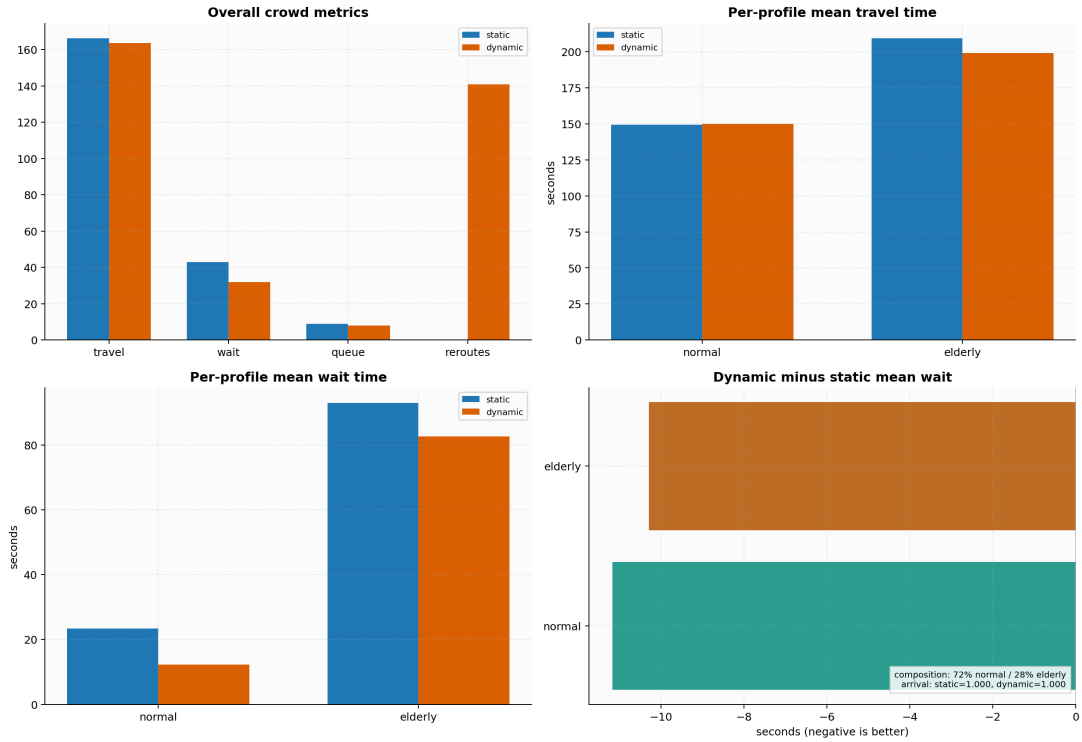


Figure 6.11.: Mixed-agent simulation results for the implemented pedestrian profiles. Both normal and elderly agents completed their trips, but elderly agents experienced higher travel times and substantially higher waiting times. Dynamic replanning reduced waiting time for both groups in this prototype run.

files. The result therefore supports profile-sensitive movement analysis, but not yet detailed profile-specific connector-load analysis.

6.3.5. Experiment E: Algorithm-threshold Sensitivity Results

Experiment E evaluates how congestion-aware replanning responds to algorithmic threshold settings and graph-construction choices. The experiment addresses two related issues. First, it tests the sensitivity of the dynamic routing mechanism to replanning parameters on the default graph. Second, it applies the balanced dynamic setting to the sparse semantic-light graph to examine whether graph abstraction affects movement performance even when the graph remains connected and routable.

Three threshold settings were tested on the default graph with 200 agents. The aggressive setting uses a low waiting trigger and high congestion sensitivity, so agents replan quickly. The balanced setting is the default dynamic configuration used in Experiment B. The conservative setting uses a higher waiting trigger and lower congestion sensitivity, so agents tolerate congestion for longer before replanning. A fourth setting applies the balanced threshold configuration to the sparse semantic-light graph.

Table 6.9 summarizes the results. On the default graph, the aggressive setting completed 198 of 200 agents and produced the lowest mean waiting time, 46.13 s, and the smallest maximum queue, 36 agents. This improvement came with a high rerouting burden: 3,814 reroute events, or 19.26 reroutes per completed agent. The conservative setting also completed 198 agents, but produced a higher mean waiting time, 58.90 s, and the highest maximum waiting time, 334.5 s. It generated only 660 reroute events, showing that fewer replans reduce route instability but leave more congestion unresolved.

The balanced setting sits between these two extremes. It completed 196 agents, produced a mean waiting time of 52.37 s, and generated 1,996 reroute events. It is therefore not the optimum for every metric, but it provides a practical compromise between waiting-time reduction, route stability, and computational effort.

Table 6.9.: Results of Experiment E: algorithm-threshold sensitivity and sparse graph variant.

Setting	Completed	Arrival rate	Mean travel	P95 travel	Mean wait	Max queue	Reroutes	Runtime
Aggressive, default graph	198 / 200	99.0%	196.91 s	325.48 s	46.13 s	36	3,814	386.12 s
Balanced, default graph	196 / 200	98.0%	197.84 s	311.72 s	52.37 s	44	1,996	0.00 s[a]
Conservative, default graph	198 / 200	99.0%	186.80 s	328.37 s	58.90 s	45	660	262.11 s
Balanced, sparse semantic-light graph	199 / 200	99.5%	202.87 s	314.68 s	61.58 s	51	2,330	80.40 s

[a]The balanced default setting reuses the stored dynamic result from Experiment B; therefore, the runtime reported in the exported file is not directly comparable with newly executed runs.

Figure 6.12 visualizes the threshold-sensitivity and graph-variant results. The main pattern is that aggressive replanning reduces waiting and queue length but increases route switching and computational cost. Conservative replanning reduces rerouting frequency but leaves more waiting unresolved. The sparse graph result adds a second interpretation: graph abstraction affects performance even under the same balanced dynamic-routing rule.

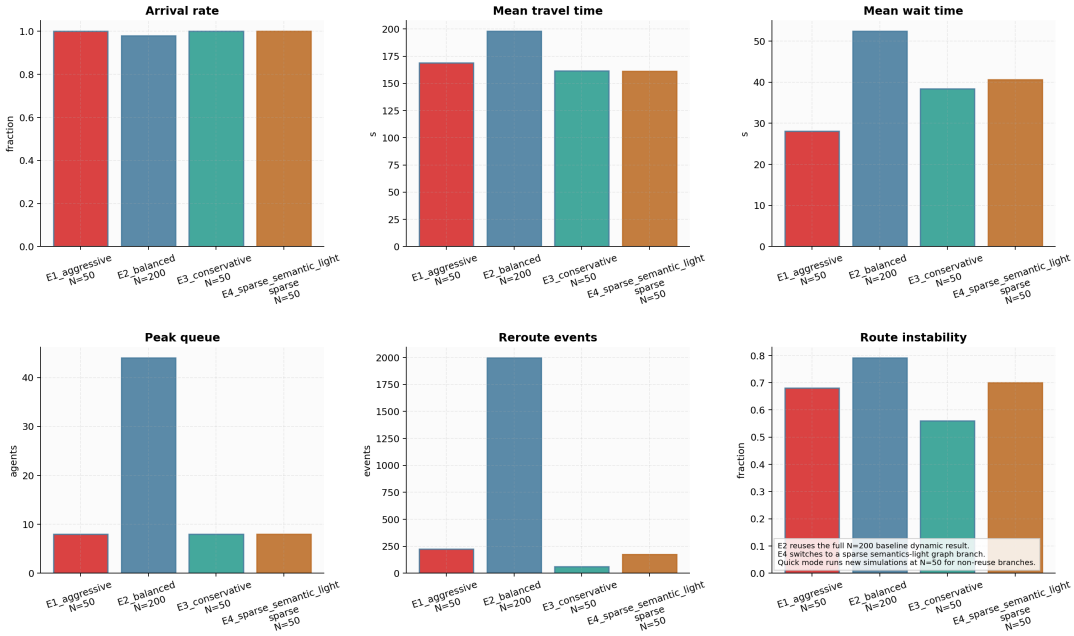


Figure 6.12.: Algorithm-threshold sensitivity and graph-variant result. Aggressive replanning reduces waiting and peak queue length but causes substantially more route changes, while the sparse semantic-light graph produces higher waiting and queue values under the same balanced dynamic-routing setting.

The sparse semantic-light graph has a higher arrival rate than the balanced default graph in this fixed-demand run, but it performs worse on movement-performance metrics: mean travel time increases from 197.84 s to 202.87 s, mean waiting time increases from 52.37 s to 61.58 s, maximum queue increases from 44 to 51 agents, and reroute events increase from 1,996 to 2,330. This indicates that arrival rate or reachability alone is not sufficient for evaluating graph quality. The sparse graph remains connected and mostly routable, but the coarser abstraction and weaker semantic support increase movement burden and rerouting pressure.

Experiment E therefore supports two conclusions. First, congestion-aware replanning is sensitive to threshold design: aggressive, balanced, and conservative settings produce different trade-offs between waiting reduction and route stability. Second, graph construction choices affect simulation behaviour even when the same routing rule is used. This directly strengthens the answer to RQ2.

6.3.6. Experiment F: Building Capacity and Agent-load Threshold Results

Experiment F evaluates how the generated station graph responds to increasing pedestrian demand. Unlike Experiment E, which varies algorithmic threshold parameters at a fixed demand level, Experiment F varies the number of agents while keeping the graph variant and routing mode explicit. The purpose is to identify the approximate demand range in which waiting time, queue length, not-completed agents, or runtime begin to increase sharply.

This experiment is not an official passenger-capacity assessment of Dadongmen Station. The demand threshold reported here is a model-based saturation point under the assumptions of the current graph, OD definition, agent profiles, connector-capacity treatment, and simulation horizon. It is useful for comparing graph and routing settings, but it does not replace calibrated passenger-flow modelling based on time-resolved AFC records or observed pedestrian trajectories.

Four demand-response sequences were tested: default graph with static routing, default graph with dynamic routing, sparse semantic-light graph with static routing, and sparse semantic-light graph with dynamic routing. The tested demand levels are 50, 100, 200, 300, and 500 agents. The 200-agent level corresponds to the main controlled demand level used in Experiments A and B, while the 300- and 500-agent levels are used to stress the graph and identify degradation behaviour.

Table 6.10 reports the static-routing capacity sequence, and Table 6.11 reports the dynamic-routing sequence.

The static-routing sequence shows a clear difference between the default and sparse graph variants. On the default graph, arrival rate remains above 95% even at 500 agents, although waiting time and maximum queue increase steadily. On the sparse graph, however, the arrival rate drops to 90.3% at 300 agents and 74.2% at 500 agents. Mean waiting time also increases more sharply, reaching 200.93 s at 500 agents. This indicates that the sparse graph is less robust under increasing demand.

The dynamic-routing sequence shows that congestion-aware replanning reduces waiting and queue pressure in both graph variants. At 200 agents on the default graph, dynamic routing reduces mean waiting time from 69.84 s to 49.54 s and maximum queue from 63 to 40 agents. On the sparse graph at the same demand level, dynamic routing reduces

Table 6.10.: Experiment F static-routing capacity sequence for the default and sparse graph variants.

Graph and demand	Agents	Completed	Arrival rate	Mean travel	Mean wait	Max queue	Not completed
Default, 50 agents	50	50	100.0%	164.71 s	41.45 s	9	0
Default, 100 agents	100	99	99.0%	174.34 s	55.15 s	27	1
Default, 200 agents	200	195	97.5%	187.86 s	69.84 s	63	5
Default, 300 agents	300	295	98.3%	208.04 s	88.12 s	107	5
Default, 500 agents	500	476	95.2%	215.69 s	102.27 s	166	24
Sparse, 50 agents	50	49	98.0%	179.52 s	69.63 s	15	1
Sparse, 100 agents	100	100	100.0%	207.85 s	96.06 s	43	0
Sparse, 200 agents	200	199	99.5%	224.31 s	114.42 s	84	1
Sparse, 300 agents	300	271	90.3%	237.01 s	144.11 s	138	29
Sparse, 500 agents	500	371	74.2%	275.29 s	200.93 s	301	129

Table 6.11.: Experiment F dynamic-routing capacity sequence for the default and sparse graph variants.

Graph and demand	Agents	Completed	Arrival rate	Mean travel	Mean wait	Max queue	Not completed	Reroutes
Default, 50 agents	50	50	100.0%	171.47 s	36.89 s	8	0	190
Default, 100 agents	100	100	100.0%	171.15 s	36.08 s	18	0	581
Default, 200 agents	200	196	98.0%	194.95 s	49.54 s	40	4	1,966
Default, 300 agents	300	292	97.3%	205.28 s	63.81 s	64	8	3,647
Default, 500 agents	500	461	92.2%	224.21 s	83.23 s	154	39	8,573
Sparse, 50 agents	50	50	100.0%	158.18 s	39.31 s	9	0	169
Sparse, 100 agents	100	100	100.0%	173.18 s	45.30 s	22	0	586
Sparse, 200 agents	200	196	98.0%	211.86 s	63.55 s	56	4	2,337
Sparse, 300 agents	300	281	93.7%	214.34 s	78.31 s	82	19	3,566
Sparse, 500 agents	500	438	87.6%	235.57 s	98.75 s	125	62	7,724

mean waiting time from 114.42 s to 63.55 s and maximum queue from 84 to 56 agents. This confirms that dynamic routing can mitigate congestion on both graph variants.

However, dynamic routing does not eliminate the effect of graph abstraction. At 500 agents, the sparse dynamic run still has a lower arrival rate than the default dynamic run, 87.6% compared with 92.2%. It also has more not-completed agents, 62 compared with 39. Therefore, dynamic replanning improves performance but does not fully compensate for the reduced graph density and weaker semantic support of the sparse graph.

Figure 6.13 visualizes the demand-response curves, while Figure 6.14 separates the static and dynamic sequences by graph variant.

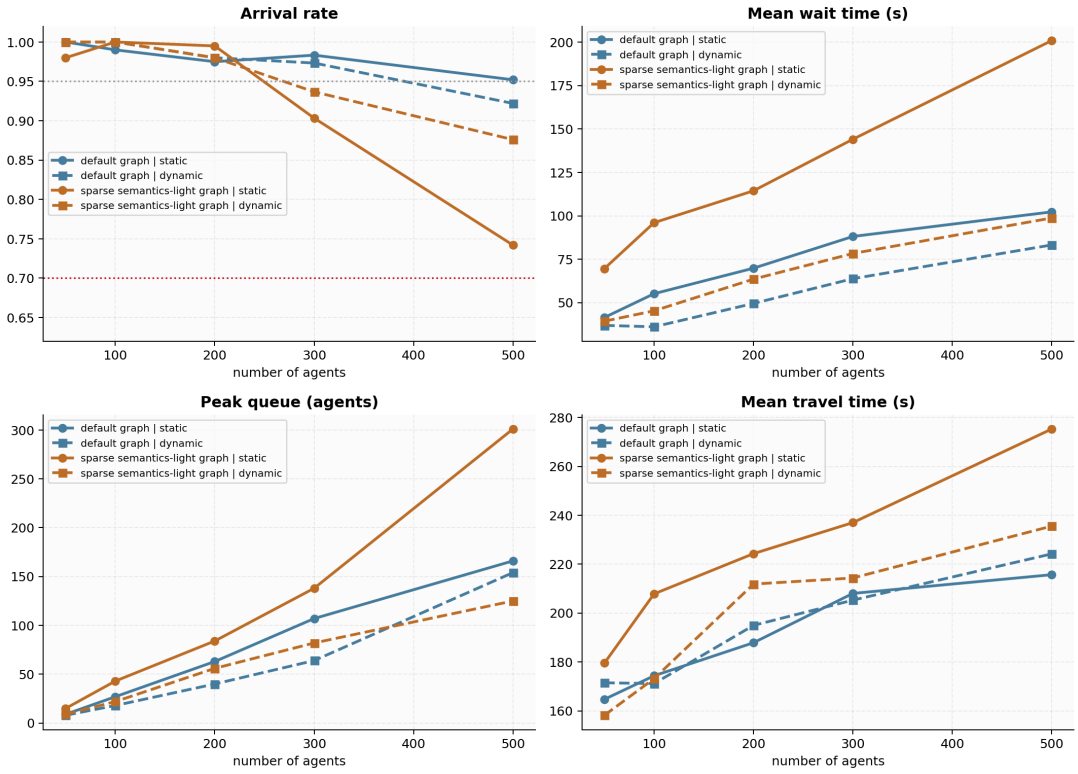


Figure 6.13.: Building capacity and agent-load threshold results for the default and sparse graph variants. Increasing demand produces nonlinear growth in waiting time and queue length, and the sparse graph shows earlier degradation under higher demand.

The capacity experiment has three main implications. First, the 200-agent demand used in the main experiments is a reasonable controlled demand level: it produces visible waiting and queueing while avoiding complete saturation. Second, congestion-aware replanning reduces waiting and queue pressure but cannot create additional structural capacity. Third, graph abstraction strongly affects demand robustness. The sparse graph remains connected, but it degrades earlier under increasing demand. This strengthens the answer to RQ2 by showing that graph construction choices influence movement performance, not only graph size or visual complexity.

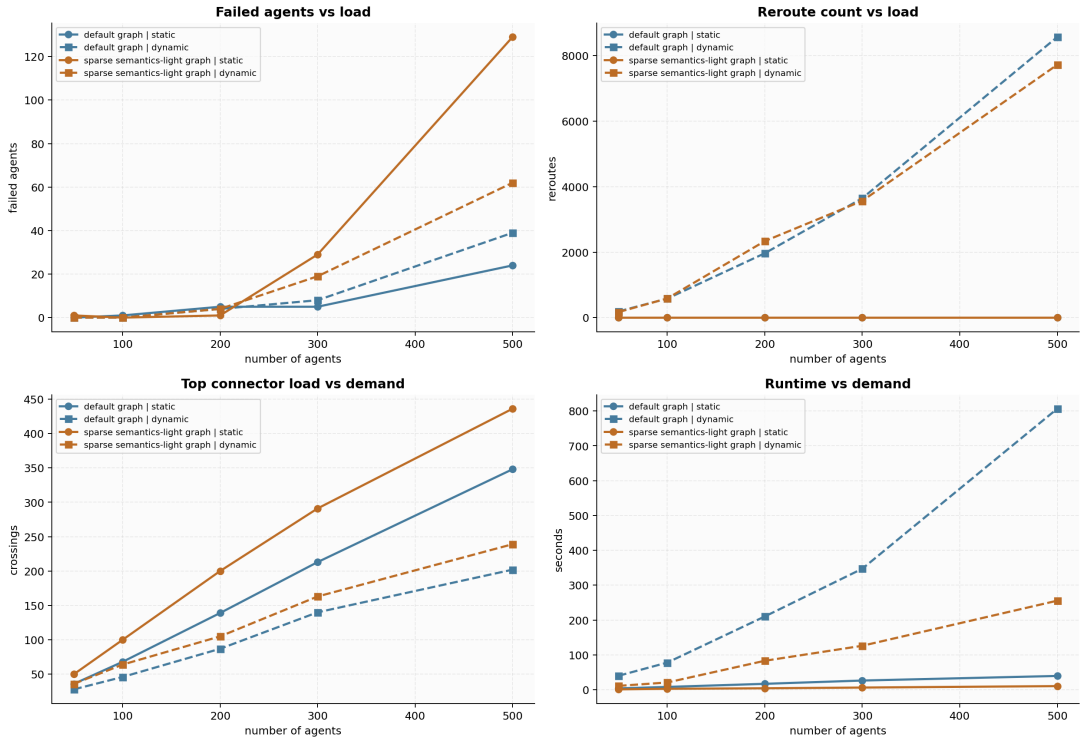


Figure 6.14.: Static and dynamic demand-response breakdown for the default and sparse graph variants. Congestion-aware replanning reduces queueing pressure, but graph abstraction still affects arrival rate, waiting time, and model-based saturation behaviour.

6.4. Overall Discussion

This section synthesizes the experimental results reported in the previous sections. The purpose is not to repeat the numerical tables, but to connect the results back to the research questions. The discussion is organized around five aspects: the feasibility of the IFC-to-2.5D graph workflow, the effectiveness of congestion-aware replanning, the influence of pedestrian heterogeneity, the distinction between algorithmic sensitivity, graph abstraction, and demand-load response, and the limitations of the results.

A central interpretation of the chapter is that a simulation-ready indoor graph cannot be evaluated only by connectivity or graph size. The experiments show that graph construction choices, semantic OD support, routing logic, pedestrian profile, and demand level all affect movement-performance outcomes such as travel time, waiting time, queue length, rerouting, and completion.

6.4.1. Feasibility of the IFC-to-2.5D graph workflow

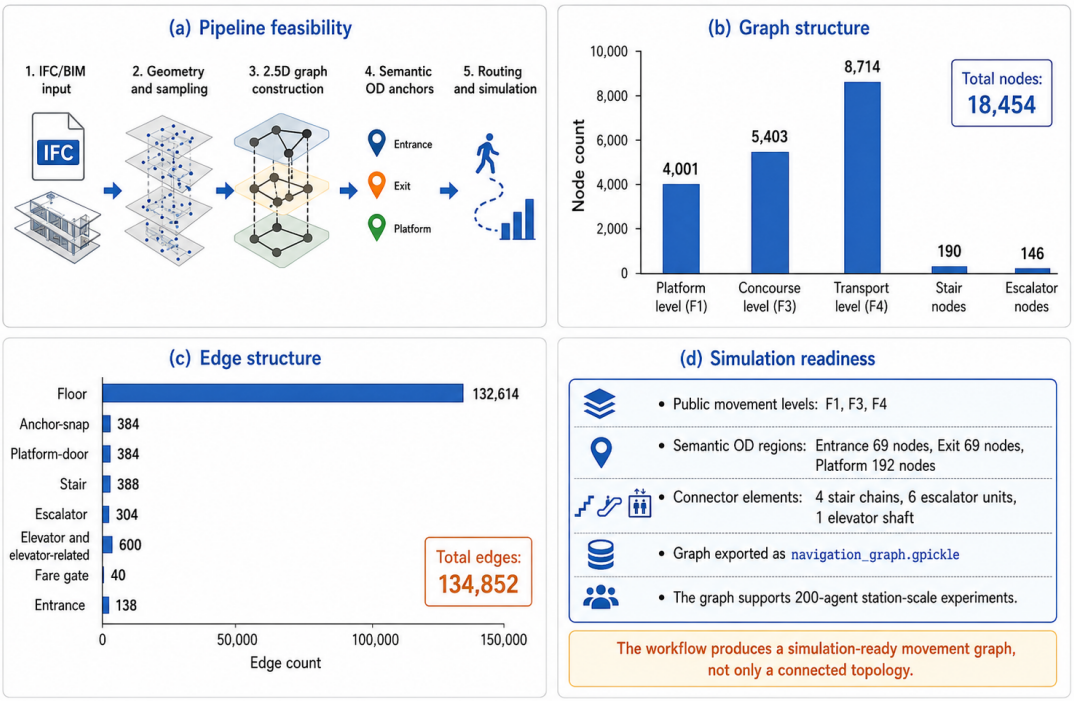


Figure 6.15.: Discussion summary of the feasibility of the IFC-to-2.5D graph workflow. The generated graph preserves floor-based movement, semantic OD regions, and explicit connector structures, making it suitable for routing and simulation rather than only topological inspection.

The results support the feasibility of transforming IFC-derived station information into a simulation-ready 2.5D navigation graph. As summarized in Figure 6.15, the workflow does not directly convert BIM geometry into a graph in one step. Instead, it proceeds through a

staged process: IFC/BIM interpretation, geometric filtering, 2.5D walkable-layer construction, graph generation, semantic OD anchoring, and routing or simulation. This staged structure is important because it makes the transformation process inspectable.

The default graph contains 18,454 nodes and 134,852 edges. These numbers alone do not prove that the graph is useful, but they show that the workflow can produce a station-scale graph rather than only a small pilot network. More importantly, the graph is semantically structured. It includes floor movement edges, stair edges, escalator edges, elevator-related edges, fare-gate edges, entrance edges, platform-door edges, and anchor-snap edges. This means that the graph preserves movement-relevant station semantics instead of reducing the station to an undifferentiated spatial mesh.

The node distribution also supports the 2.5D modelling strategy. Public movement is represented on the Platform level (F1), Concourse level (F3), and Transport level (F4), while stairs, escalators, and elevators explicitly represent inter-level movement. This confirms that the graph is neither a purely 2D floor-plan graph nor a full 3D free-space model. It is a layered movement graph designed for pedestrian routing and simulation.

The semantic OD regions further strengthen this interpretation. Entrance, exit, and platform regions are represented by groups of graph nodes rather than by arbitrary coordinates. This allows simulated agents to be spawned and routed through meaningful station functions. The fact that the same graph supports the 200-agent routing and simulation experiments indicates that the workflow produces a usable movement substrate, not merely a connected graph.

The sparse semantic-light graph adds an important qualification to this conclusion. It remains connected and can still support routing and simulation, but it has substantially fewer nodes and edges, smaller OD regions, and weaker semantic support. The later experiments show that this structural simplification changes movement outcomes. Therefore, feasibility should not be judged only by whether a graph is connected or whether agents can find routes. A simulation-ready indoor graph also needs sufficient spatial density, connector representation, and semantic anchoring to produce stable movement-performance results.

Therefore, the main contribution of the graph-construction stage is not only connectivity. The workflow produces a semantically enriched graph that is structured enough to support route computation, waiting analysis, queueing, dynamic replanning, profile-based simulation, and graph-variant comparison. This directly supports the first research question and also provides evidence for the second research question concerning graph construction choices.

6.4.2. Effectiveness of congestion-aware replanning

The comparison between static routing and congestion-aware replanning shows that dynamic routing should be interpreted as a congestion-management mechanism rather than as a shortest-time optimizer. Figure 6.16 summarizes this trade-off. Under the dynamic condition in Experiment B, the arrival rate increases slightly from 97.5% to 98.0%, but mean travel time increases from 185.47 s to 197.84 s. The 95th percentile travel time also increases from 280.05 s to 311.72 s. If travel time were the only evaluation criterion, the dynamic strategy would therefore appear less effective.

However, the waiting and queue metrics tell a different story. Mean waiting time decreases from 68.54 s to 52.37 s, corresponding to a 23.6% reduction. Median waiting time decreases

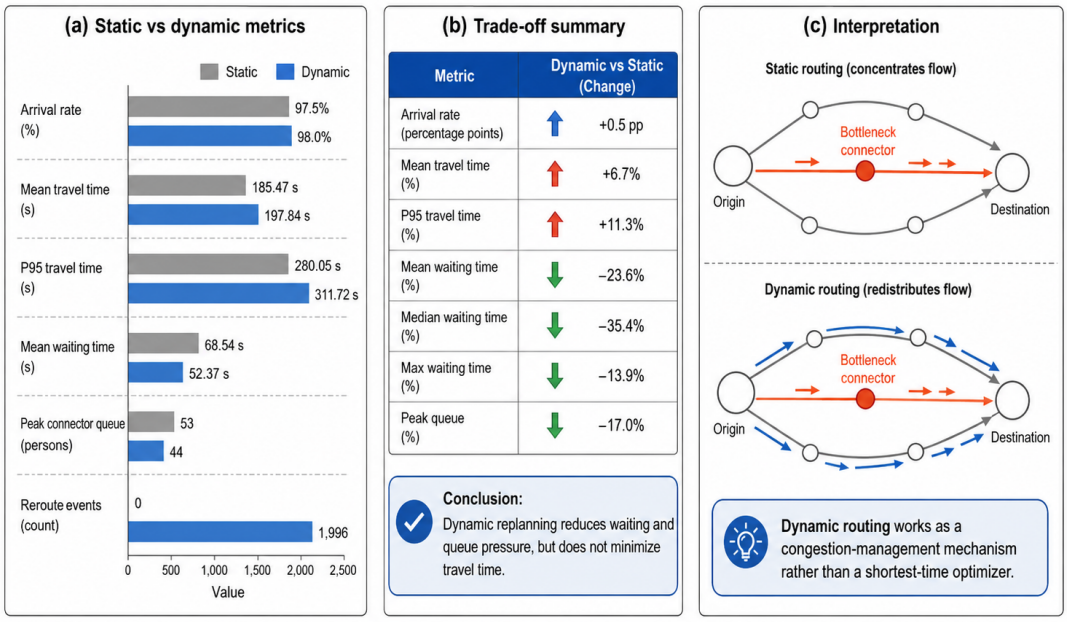


Figure 6.16.: Discussion summary of the effect of congestion-aware replanning in the 200-agent experiment. Dynamic replanning reduces waiting and queue pressure, but does not minimize mean or high-percentile travel time.

by 35.4%, maximum waiting time decreases by 13.9%, and peak connector queue decreases from 53 to 44 agents. These results indicate that dynamic replanning reduces local congestion exposure even though it increases average travel duration.

This also explains why the mean travel time does not decrease in Experiment B. If the routing objective were calibrated to minimize total travel time directly, a decrease in mean travel time might be expected. However, the implemented method reacts to congestion by increasing the cost of locally crowded graph elements. This can send agents to longer but less congested alternatives. The observed result is therefore consistent with a congestion-avoidance strategy: queueing decreases, but route directness may also decrease. The main improvement is not a shorter average trip, but a reduction in waiting burden and connector queue concentration.

This trade-off is consistent with the purpose of congestion-aware replanning. In a station graph, the shortest path is not always the most operationally desirable path when many agents compete for the same connector. Static routing concentrates flow on the most attractive connector, creating local queue pressure. Dynamic routing redistributes part of this flow to alternative paths when congestion indicators become high. As illustrated in the conceptual route-sketch panel of Figure 6.16, dynamic routing therefore reduces bottleneck pressure by spreading demand across the graph.

The graph-variant capacity results provide further evidence for this interpretation. At 200 agents on the default graph, dynamic routing reduces mean waiting time from 69.84 s to 49.54 s and maximum queue from 63 to 40 agents. On the sparse graph at the same demand level, dynamic routing reduces mean waiting time from 114.42 s to 63.55 s and maxi-

mum queue from 84 to 56 agents. This shows that congestion-aware replanning can reduce queueing pressure on both graph variants.

At the same time, the sparse graph still produces higher waiting and queue values than the default graph. This means that dynamic replanning improves queue-related performance within the constraints of a given graph, but it does not remove the effect of graph abstraction. The routing algorithm can redistribute demand, but it cannot fully compensate for reduced graph density, weaker semantic anchoring, or structural bottlenecks.

The result also shows why evaluation should not rely on a single metric. Mean travel time, waiting time, peak queue, reroute count, and not-completed agents describe different aspects of the movement process. The dynamic run generated 1,996 replanning events in Experiment B, which indicates active adaptation, but also additional route instability and computational work. Therefore, congestion-aware replanning should be assessed as a multi-objective trade-off: it can reduce waiting and queueing, but may increase route length, travel time, and algorithmic activity.

This finding answers the third research question in a qualified way. Congestion-aware replanning improves movement performance if performance is defined in terms of waiting-time reduction and queue-pressure mitigation. It does not necessarily improve performance if the objective is to minimize mean travel time. For station environments, this distinction is important because severe local queues may be more critical than small increases in route duration.

6.4.3. Influence of pedestrian heterogeneity

The profile-based experiments show that pedestrian heterogeneity has a strong effect on simulation outcomes. Figure 6.17 summarizes the differences between normal and elderly agents in both single-profile and mixed-agent settings. The main pattern is consistent across the experiments: elderly agents experience longer travel times and much higher waiting burdens.

In the single-profile comparison, the normal static run completes all 50 agents, with a mean travel time of 135.91 s and a mean waiting time of 12.28 s. By contrast, the elderly static run completes 45 of 50 agents, with a mean travel time of 238.56 s and a mean waiting time of 126.99 s. The peak connector queue also increases from 7 to 14 agents. This indicates that lower walking speed does not only increase travel time mechanically; it also increases exposure to waiting and queueing.

The mixed-agent results show the same direction. In the static mixed run, normal agents have a mean travel time of 149.48 s and a mean waiting time of 23.38 s, while elderly agents have a mean travel time of 209.21 s and a mean waiting time of 93.00 s. In the dynamic mixed run, waiting time decreases for both groups, but elderly agents remain more affected: their mean waiting time is still 82.71 s, compared with 12.21 s for normal agents.

These results demonstrate why aggregate metrics can be misleading. In Experiment D, the overall arrival rate is 100% for both static and dynamic mixed runs. If only arrival rate were reported, the simulation would appear equally successful for both groups. However, the profile-specific metrics reveal a substantial inequality in movement burden. Elderly agents arrive, but they do so with much higher waiting and delay exposure.

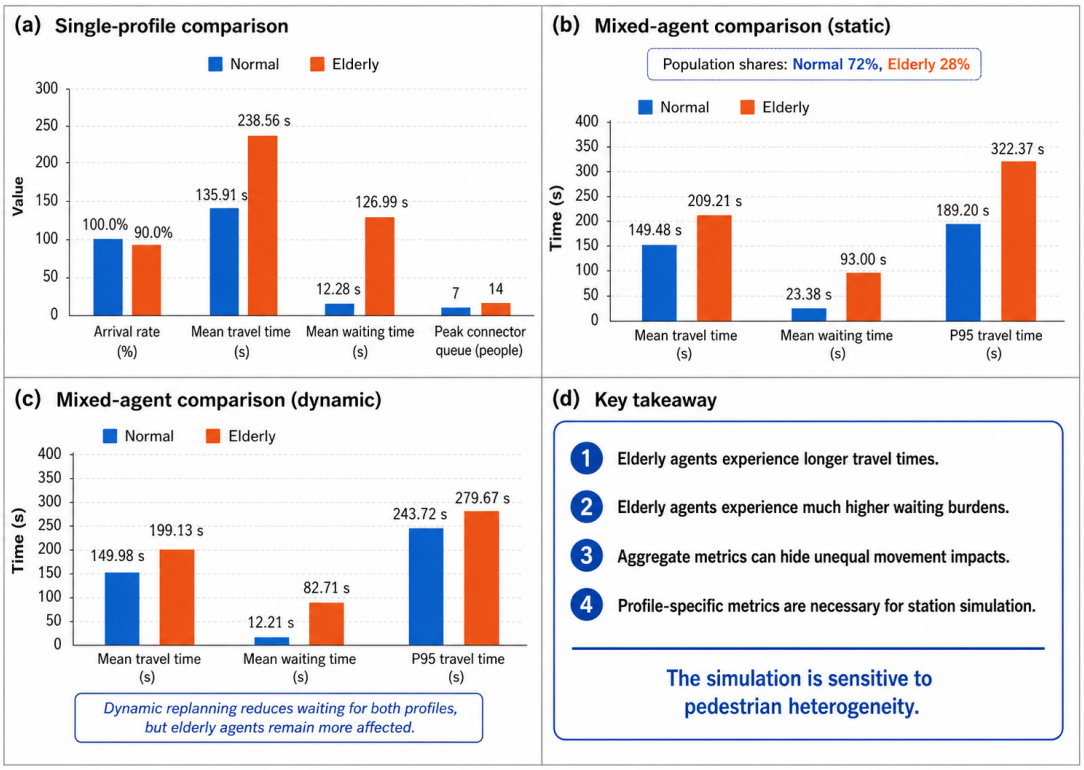


Figure 6.17.: Discussion summary of pedestrian heterogeneity. Elderly agents experience longer travel times and substantially higher waiting burdens than normal agents in both single-profile and mixed-agent settings.

This finding supports the fourth research question. It shows that pedestrian-profile assumptions influence travel time, waiting, completion, and queue exposure. The current implementation is still limited because it includes normal and elderly profiles but not a fully implemented wheelchair or accessibility-constrained route model. Therefore, the results demonstrate profile-sensitive simulation capability rather than a complete accessibility analysis. Future work can extend this by adding explicit accessibility attributes, elevator dependence, stair restrictions, and profile-specific route preferences.

6.4.4. Algorithmic sensitivity versus building capacity

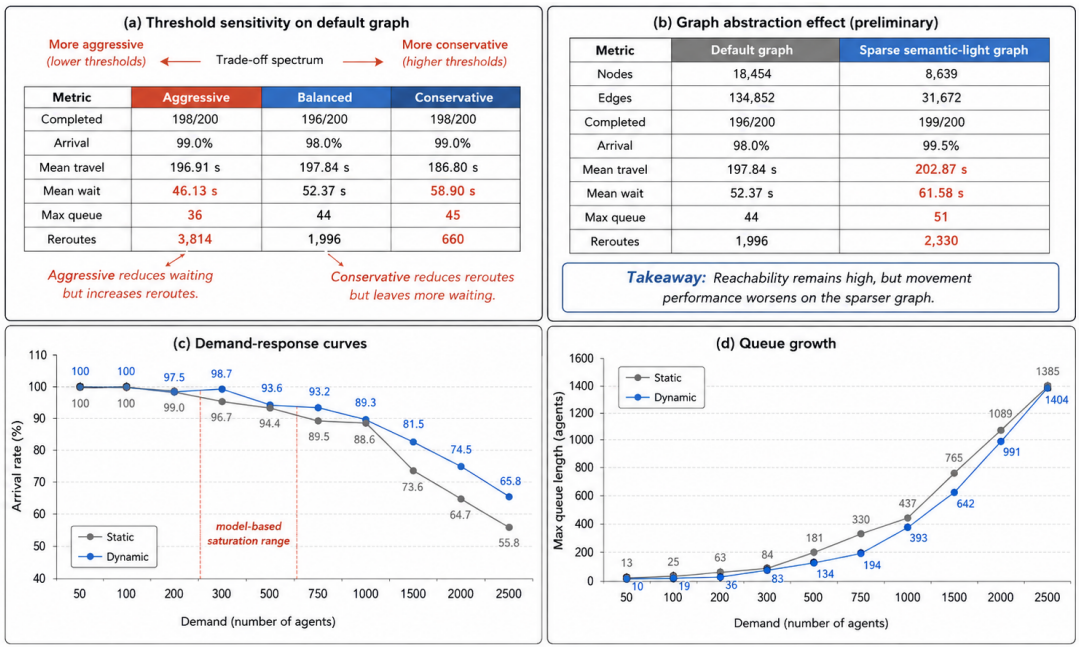


Figure 6.18.: Discussion summary of algorithmic sensitivity, graph abstraction, and building-capacity behaviour. Algorithmic thresholds and graph abstraction affect performance at fixed demand, while increasing demand reveals model-based saturation and structural capacity limits.

The results distinguish between three related but different design questions: algorithmic sensitivity, graph abstraction, and building-capacity response. Experiment E concerns algorithmic sensitivity: it tests how replanning thresholds affect dynamic routing at fixed demand. The graph-variant part of Experiment E concerns graph abstraction: it tests whether the same dynamic rules behave differently on a coarser and semantics-light graph. Experiment F concerns demand-load response: it tests how the graph behaves as the number of agents increases.

The algorithm-threshold experiment shows that replanning parameters strongly affect the result. Aggressive replanning reduces waiting and queueing but produces many more reroute events. Conservative replanning reduces route switching but leaves more waiting unresolved. The balanced setting is therefore a practical compromise rather than an absolute optimum. This means that congestion-aware replanning is not a binary feature that simply

improves or does not improve the simulation. Its effect depends on the decision thresholds used to trigger route updates.

The sparse graph results strengthen the answer to RQ2. At fixed 200-agent demand, the sparse graph produces higher mean travel time, higher waiting time, a larger queue, and more reroute events than the default graph under balanced dynamic routing. Under increasing demand, the sparse graph degrades earlier, especially under static routing. It drops below 95% arrival rate at 300 agents in both static and dynamic capacity tests, while the default graph remains above this threshold at 300 agents and only drops below 95% under dynamic routing at 500 agents.

The capacity experiment shows that increasing pedestrian demand produces nonlinear degradation. The 200-agent setting already reveals meaningful waiting and queueing while retaining high arrival rates, which supports its use as the main controlled demand level. The 300–500 agent range marks clearer model-based saturation, especially for the sparse graph and for dynamic routing at higher demand. In other words, the 200-agent scenario is useful for controlled comparison, while the higher demand levels reveal when the model begins to lose robustness.

Dynamic routing reduces queue pressure at several demand levels, but it does not remove the structural capacity limit. This is an important interpretation. Congestion-aware replanning can redistribute agents and reduce waiting under suitable conditions, but it cannot create additional physical connector capacity. Once demand exceeds the effective capacity of the graph representation, both static and dynamic strategies degrade.

This distinction helps answer RQ2. Graph construction choices, connector representation, semantic OD support, and routing thresholds affect the navigability and realism of the simulation results. A graph that appears acceptable by connectivity metrics may still produce different waiting, queueing, and rerouting behaviour. Therefore, the thesis evaluates graph design choices through movement-performance metrics, not only through graph size, graph connectivity, or visual simplicity.

6.4.5. Limitations of the results

Several limitations should be considered when interpreting the results. Figure 6.19 summarizes the most important ones. These limitations do not invalidate the workflow, but they define the scope of the claims that can be made from the experiments.

First, the simulation is not directly calibrated with observed passenger-flow data. The 200-agent setting is used as a controlled 10-minute scenario load, not as a calibrated estimate of real station demand. Public passenger-flow reports are useful for contextualizing the importance of Dadongmen Station, but they do not provide the time-resolved, entrance-specific, platform-specific, or OD-specific data required to validate the simulated crowding level. Therefore, the thesis does not claim that the simulated crowd density exactly matches the real crowding condition of Dadongmen Station. Such a claim would require time-resolved AFC data, entrance-level counts, platform counts, connector queue observations, or video-derived pedestrian trajectories.

Second, pedestrian behaviour is simplified. Waiting, queueing, connector capacity, and replanning are represented through graph-based rules. This is appropriate for testing the proposed 2.5D graph workflow, but it is less detailed than full microscopic continuous pedestrian dynamics. The simulation can reveal bottleneck tendencies and comparative route

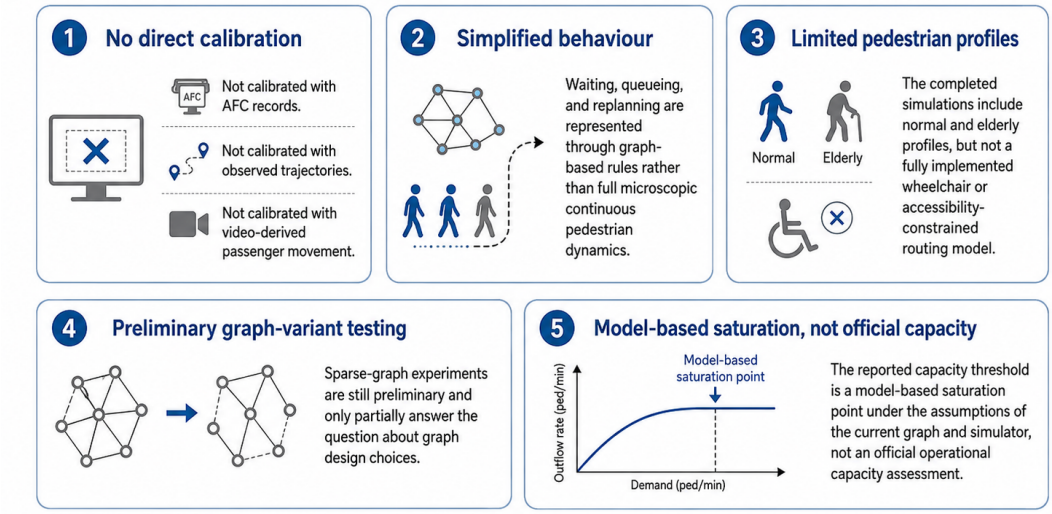


Figure 6.19.: Summary of the main limitations of the results. The experiments support methodological validation of the IFC-to-2.5D graph workflow, but they should not be interpreted as calibrated operational prediction.

effects, but it cannot reproduce all local interactions, body movement, lane formation, or behavioural adaptation found in real pedestrian crowds.

Third, the pedestrian profiles are limited. The completed experiments include normal and elderly profiles. These profiles are sufficient to show that the framework is sensitive to pedestrian heterogeneity, but they do not provide a complete accessibility model. A fully implemented wheelchair or accessibility-constrained profile would require additional connector accessibility attributes, elevator availability assumptions, and route restrictions that were not fully completed in the current prototype.

Fourth, the graph-variant testing uses one specific sparse graph configuration. This variant is useful because it changes grid resolution, neighbourhood rule, blind-path stitching, and semantic OD support, but it does not exhaust all possible graph-abstraction choices. A full graph-design study would require a systematic sweep over resolution, neighbourhood rules, connector modelling, semantic filtering, and OD-region construction.

Fifth, the capacity threshold is model-based. Experiment F identifies a saturation range under the assumptions of the current graph variants and simulator. It should not be interpreted as the official capacity of Dadongmen Station. Real capacity assessment would require calibrated demand data, operational constraints, emergency standards, timetable interactions, and validation against observed station behaviour.

The simulated demand should therefore be interpreted as a scenario load rather than as a measured crowding level. The 200-agent setting creates visible but not fully saturated movement interaction within a 600 s simulation horizon, while the demand-load experiment tests lower and higher loads to observe how the graph and routing logic respond. This design is suitable for comparing model behaviour, graph variants, and routing strategies, but it cannot verify that the simulated crowding matches real station crowding without external validation data.

In addition, the number of not-completed agents depends partly on the fixed 600 s simulation horizon and the implemented graph-based movement rules. It should therefore be interpreted as a model output under controlled assumptions, not as a direct prediction of real passenger failure or safety risk.

The implication is that the thesis provides methodological validation rather than operational certification. The results show that the IFC-to-2.5D graph workflow can support routing, congestion-aware replanning, profile-sensitive simulation, graph-variant comparison, and demand-load testing. However, further calibration and behavioural validation would be required before the method could be used for operational passenger-flow prediction or formal station-capacity assessment.

6.5. Chapter Summary

This chapter presented and discussed the results of the proposed IFC-to-2.5D graph workflow and the associated pedestrian simulation experiments. The workflow produced a station-scale default graph with 18,454 nodes and 134,852 edges, as well as a sparse semantic-light graph variant with 8,639 nodes and 31,672 edges. The default graph preserves floor-based movement, vertical connectors, access-control elements, entrance links, platform-door edges, and semantic origin–destination anchors. The sparse graph variant remains routable, but deliberately reduces graph density and semantic support. This comparison provides a basis for evaluating how graph construction choices affect simulated movement behaviour.

Experiment A confirmed that the default graph can support station-scale static routing and simulation within the 600 s simulation horizon. At the same time, it showed that fixed shortest-path routing concentrates demand on dominant vertical connectors and produces local bottlenecks. Experiment B showed that congestion-aware replanning reduces mean waiting time and peak connector queue, although it increases mean and high-percentile travel time. This indicates that dynamic replanning works mainly as a congestion-management mechanism rather than as a pure shortest-time optimizer.

Experiments C and D showed that pedestrian profiles affect simulation outcomes. Elderly agents experienced longer travel times and substantially higher waiting burdens than normal agents in both single-profile and mixed-agent settings. These results show that aggregate indicators such as overall arrival rate are not sufficient on their own. Profile-specific metrics are necessary to understand unequal movement burdens within the simulated population.

Experiment E showed that congestion-aware replanning is sensitive to algorithmic threshold settings. Aggressive thresholds reduced waiting and queueing but increased rerouting intensity, while conservative thresholds reduced rerouting but left more waiting unresolved. The sparse graph variant in Experiment E further showed that graph abstraction affects movement performance even when arrival rate remains high. In particular, the sparse graph produced higher mean travel time, higher mean waiting time, a larger maximum queue, and more rerouting events than the default graph under the same balanced dynamic-routing setting.

Experiment F extended the graph-variant comparison to increasing demand levels. The results showed that the default graph is more robust under higher demand, while the sparse semantic-light graph degrades earlier in terms of arrival rate, waiting time, queue length,

and not-completed agents. Congestion-aware replanning improves waiting and queue metrics in both graph variants, but it does not fully compensate for reduced graph density or weaker semantic support. This strengthens the answer to RQ2: graph construction and semantic modelling choices influence movement performance, not only graph size, reachability, or visual complexity.

Overall, the chapter shows that the generated 2.5D graph is more than a connected network extracted from BIM. It is a semantically meaningful movement substrate that can support routing, simulation, congestion-aware replanning, profile-sensitive analysis, graph-design comparison, and demand-load testing. However, the results should be interpreted as controlled methodological validation rather than calibrated operational passenger-flow prediction. The simulated demand levels are scenario loads within a fixed 600 s horizon. They are useful for comparing graph variants, routing strategies, pedestrian profiles, and demand-load response, but they do not prove that the simulated crowding matches real station crowding. The next chapter uses these findings to answer the research questions, summarize the contributions and limitations, and outline future work.

7. Conclusions and Future Work

This chapter concludes the thesis by summarizing the main findings, answering the research questions, identifying the main contributions and limitations, and outlining future work. The results should be understood as methodological validation of an IFC-to-2.5D graph workflow rather than as calibrated passenger-flow prediction. Therefore, the simulated crowding should be understood as internally consistent scenario-based crowding rather than empirically matched crowding: it is realistic enough to test whether the graph and routing logic respond plausibly to congestion, but it is not calibrated to the actual temporal distribution of passengers in Dadongmen Station.

7.1. Conclusion

This thesis investigated how IFC BIM can be transformed into a semantically enriched 2.5D indoor navigation graph and used for graph-based pedestrian routing, congestion-aware replanning, and controlled simulation in a multi-level station environment. The work started from the observation that IFC models contain rich geometric and semantic information, but are not directly usable as navigation or simulation models.

The proposed workflow extracts movement-relevant information from IFC-derived data, abstracts the station into floor-based 2.5D graph layers, represents vertical connectors explicitly, and attaches semantic origin–destination anchors to graph nodes. The resulting graph was evaluated through static routing, congestion-aware replanning, pedestrian-profile comparison, algorithm-threshold analysis, graph-variant comparison, and demand-load testing.

The final station-scale workflow produced a default graph with 18,454 nodes and 134,852 edges. This graph supported controlled 600 s simulation runs and allowed travel time, waiting time, queue length, rerouting, and completion to be analysed. The results show that static routing exposes connector bottlenecks, while congestion-aware replanning reduces waiting and queue pressure at the cost of longer travel times and more route changes. The pedestrian-profile experiments further show that elderly agents experience higher travel and waiting burdens than normal agents.

A key finding that emerged during the implementation is that graph connectivity alone is not sufficient. The sparse semantic-light graph variant remained routable, but produced higher waiting, larger queues, and earlier demand-load degradation than the default graph. Therefore, an indoor navigation graph should not be evaluated only by topology, reachability, or visual simplicity. It should also be evaluated through movement-performance metrics.

Overall, the thesis demonstrates that IFC-derived station information can be transformed into a semantically meaningful movement substrate for routing and simulation. At the same time, the results remain controlled methodological validation. They do not constitute calibrated operational prediction of Dadongmen Station passenger flow or real crowd

density. The simulated crowding levels are controlled experimental scenarios used to test graph behaviour, congestion-aware replanning, pedestrian-profile effects, and demand-load response. Matching real station crowding would require time-resolved passenger-flow and trajectory data, which were not available in this study.

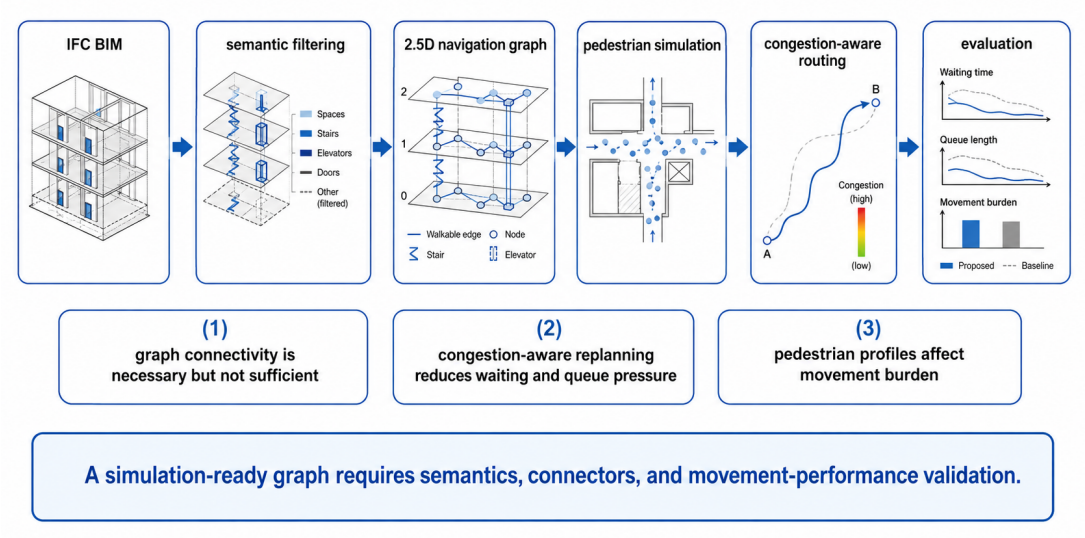


Figure 7.1.: Summary of the thesis conclusion. The workflow transforms IFC-derived building information into a semantically enriched 2.5D graph that supports routing, congestion-aware replanning, graph-variant comparison, and profile-sensitive pedestrian simulation.

7.2. Answers to the Research Questions

Main research question. *How can IFC BIM be transformed into a semantically enriched 2.5D indoor navigation graph and used to support congestion-aware pedestrian routing and simulation in a multi-level station environment?*

The thesis shows that this can be achieved through a staged IFC-to-2.5D graph workflow. Movement-relevant IFC information is filtered, public movement levels are represented as graph layers, vertical connectors are modelled as explicit inter-level edges, and semantic origin–destination regions are attached to valid graph nodes. The resulting graph can then be used for routing, simulation, waiting analysis, queue detection, congestion-aware replanning, and graph-design sensitivity testing.

RQ1. How can multi-file IFC BIM data be transformed into a traffic-relevant and semantically meaningful 2.5D indoor navigation graph?

Multi-file IFC BIM data can be transformed by identifying movement-relevant elements, extracting public walkable levels, projecting movement surfaces into level-based 2.5D layers, sampling valid graph nodes, filtering obstacles and restricted areas, and connecting levels through explicit stair, escalator, and elevator connectors. In the final station case, the default graph contains 18,454 nodes and 134,852 edges, including floor edges, stair edges, escalator

edges, elevator-related edges, fare-gate edges, entrance edges, platform-door edges, and semantic anchor-snap edges.

RQ2. How do graph construction and semantic modelling choices affect the navigability and movement-performance behaviour of the resulting graph?

The results show that graph construction choices affect more than graph size and reachability. The sparse semantic-light graph remains routable, but it produces higher mean travel time, higher mean waiting time, larger maximum queues, more rerouting events, and earlier degradation under increasing demand. This indicates that graph abstraction, grid resolution, neighbourhood rule, connector representation, and semantic OD support influence movement performance even when basic connectivity remains available.

RQ3. To what extent does congestion-aware replanning change pedestrian movement performance compared with fixed-routing simulation on the generated graph?

Congestion-aware replanning reduces waiting and queue pressure, but it does not necessarily reduce travel time. In the 200-agent comparison on the default graph, dynamic replanning reduces mean waiting time and peak connector queue, while increasing mean and 95th percentile travel time. This occurs because the implemented strategy penalizes locally crowded graph elements rather than solving a global total-travel-time optimization problem. Agents may therefore avoid queues by taking longer or less direct alternatives. The main value of the implemented strategy is congestion management and delay redistribution, not guaranteed shortest-time improvement.

The demand-load results show the same pattern: dynamic routing can mitigate congestion in both default and sparse graph variants, but it cannot fully compensate for structural bottlenecks or reduced graph density. This means that congestion-aware replanning should be evaluated through multiple metrics, including waiting time, queue length, route change, travel time, and not-completed agents.

RQ4. How do heterogeneous pedestrian profiles influence travel time, waiting, and congestion patterns in the station environment?

Pedestrian profiles strongly influence simulation outcomes. Elderly agents experience longer travel times and substantially higher waiting burdens than normal agents in both single-profile and mixed-agent settings. This shows that aggregate metrics such as arrival rate or mean travel time are not sufficient on their own. Profile-specific metrics are needed to understand unequal movement impacts within the simulated population. In the current implementation, this conclusion is limited to the implemented normal and elderly profiles.

7.3. Main Contributions

This thesis makes four main contributions, summarized in Figure 7.3.

First, it develops a reproducible workflow for transforming IFC-derived station information into a semantically enriched 2.5D navigation graph. The workflow links building semantics, walkable-layer extraction, graph construction, connector modelling, and semantic origin–destination anchoring.

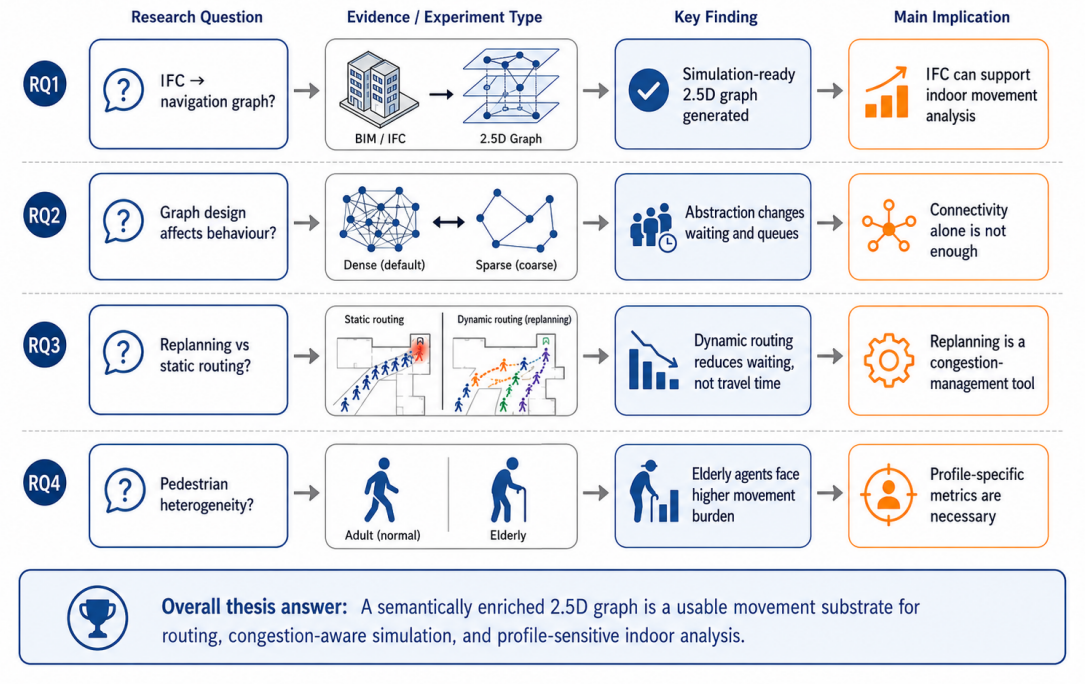


Figure 7.2.: Visual summary linking the research questions to the main experimental evidence and thesis findings.

Second, it demonstrates that a 2.5D graph can support more than static shortest-path routing. The graph can be used for agent-based movement simulation, waiting analysis, queue detection, congestion-aware replanning, and demand-load testing.

Third, it provides an experimental comparison between static routing and congestion-aware replanning. The results show that dynamic replanning reduces waiting and queue pressure, but introduces trade-offs with travel time, rerouting intensity, and computational effort.

Fourth, it shows that graph-abstraction choices and pedestrian-profile assumptions affect simulation outcomes. The comparison between the default graph and the sparse semantic-light graph demonstrates that graph quality should be evaluated through movement-performance metrics, not only graph connectivity. The profile experiments further show that different pedestrian groups can experience unequal movement burdens.

7.4. Limitations

The thesis has several limitations.

First, the simulation is not calibrated with time-resolved AFC data, video-derived pedestrian trajectories, or observed station movement records. The thesis therefore does not claim that the simulated crowd density exactly matches the real crowding condition of Dadongmen Station. Such a claim would require station-level validation data, such as automatic fare-collection records, entrance-level passenger counts, platform counts, connector queue

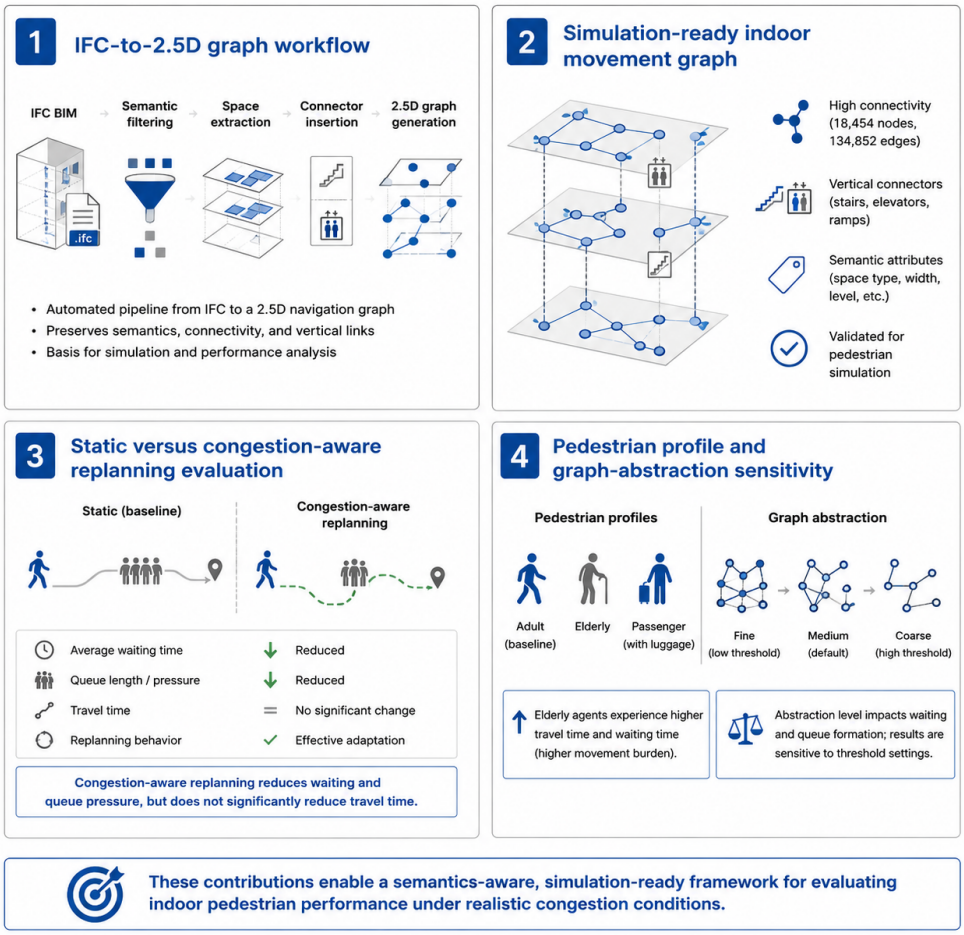


Figure 7.3.: Main contributions of the thesis: IFC-to-2.5D graph construction, simulation-ready graph representation, congestion-aware routing evaluation, and graph/profile sensitivity analysis.

observations, or video-derived pedestrian trajectories. These data were not available for this thesis. The experiments therefore support methodological validation, not operational passenger-flow prediction.

Second, pedestrian behaviour is simplified. Waiting, queueing, connector capacity, and replanning are represented through graph-based rules rather than full microscopic continuous pedestrian dynamics. The model can reveal comparative bottleneck and routing effects, but it cannot reproduce all local interactions found in real pedestrian crowds.

Third, the implemented pedestrian profiles are limited. The experiments include normal and elderly agents, but a complete wheelchair or accessibility-constrained routing model was not fully implemented. Therefore, the thesis demonstrates profile-sensitive simulation capability, but not a full accessibility analysis.

Fourth, the graph-variant testing uses one sparse semantic-light graph configuration. This variant is useful for showing that graph abstraction affects movement behaviour, but it does not cover all possible combinations of grid resolution, neighbourhood rule, connector modelling, semantic filtering, and OD-region construction.

Fifth, the capacity results are model-based. The saturation behaviour identified in Experiment F depends on the current graph variants, agent rules, OD assumptions, connector treatment, and the fixed 600 s simulation horizon. It should not be interpreted as the official capacity of Dadongmen Station or as a verified reproduction of real crowding. The simulated demand levels are scenario loads used to examine model behaviour under increasing pressure. Operational capacity assessment would require calibrated demand data, operational constraints, emergency standards, timetable interactions, and validation against observed station behaviour.

Finally, the number of not-completed agents should be interpreted carefully. It indicates agents that do not reach their assigned destination within the fixed simulation horizon or cannot maintain a valid route under the implemented movement rules. It is a simulation accounting category, not a real-world safety failure.

7.5. Future Work

Future work can develop the thesis in five directions, as summarized in Figure 7.4.

First, the simulation should be calibrated and validated using real passenger data, such as AFC records, platform counts, video-derived trajectories, or observed connector queues. This would allow the method to move from methodological validation toward operational prediction.

Second, the graph-construction process should be tested under more graph abstraction settings. A systematic comparison of grid resolution, neighbourhood rules, connector modelling, semantic filtering, OD-region construction, and connector-cost assumptions would provide a stronger answer to how graph design choices affect routing and simulation.

Third, pedestrian behaviour should be extended. Future versions should include accessibility-constrained agents, wheelchair users, luggage passengers, elevator preference, stair avoidance, group behaviour, and more realistic waiting and queueing rules.

Fourth, the method should be tested on additional station or building cases. Applying the workflow to different station layouts would show whether the method generalizes beyond the Dadongmen Station case.

Fifth, the workflow is well suited for integration with large language models. During the implementation, it became clear that the generated graph, simulation outputs, and bottleneck metrics could become tools for a station-navigation assistant. A future LLM-based system could translate natural-language queries into validated graph operations, such as route planning, accessibility-aware routing, congestion querying, or scenario simulation. For example, a user could ask for the least crowded route from an entrance to a platform, or an operator could ask which connector becomes critical under increased demand. In this direction, the 2.5D graph would act as the structured spatial reasoning layer, while the LLM would provide the user-facing interface for querying, explaining, and comparing navigation options.

This extension should be treated carefully. The LLM should not replace the graph algorithm or simulation model. Instead, it should call validated tools, retrieve graph and simulation outputs, and explain the results in natural language. This would preserve the reliability of the computational workflow while making it easier to use for real station navigation and planning support.

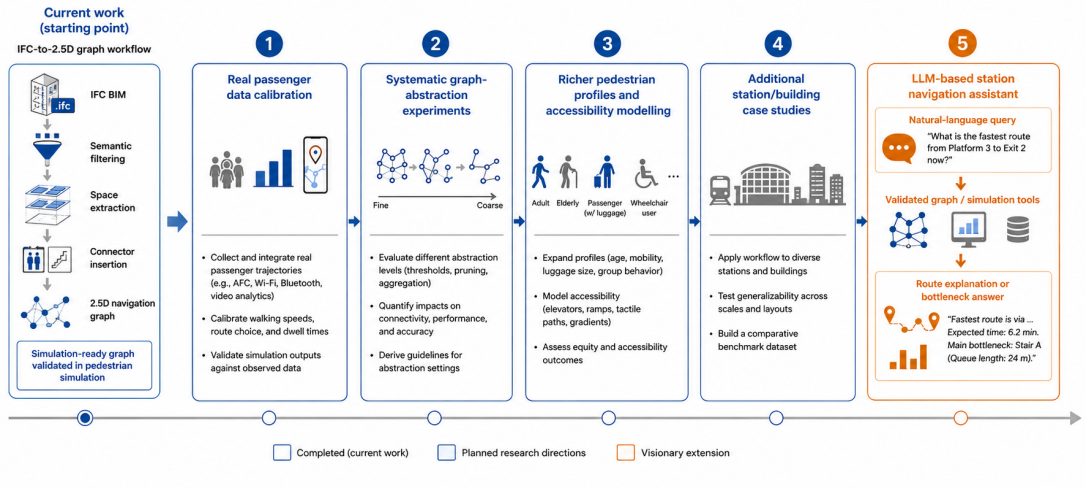


Figure 7.4.: Future work roadmap. The current workflow can be extended through data calibration, systematic graph-abstraction testing, richer pedestrian profiles, additional case studies, and LLM-based station navigation support.

A. Reproducibility self-assessment

This appendix provides a self-assessment of the reproducibility of the thesis. The work includes code-based graph construction, routing, simulation, experiment execution, and figure generation. However, the original Dadongmen Station BIM/IFC data cannot be redistributed because they originate from professional project data and were used with permission for academic research only. Therefore, the thesis is not fully reproducible from raw input data by an external reader, but the workflow, configurations, selected processed outputs, and result-generation logic are documented to support partial reproducibility.

A.1. Marks for each of the criteria

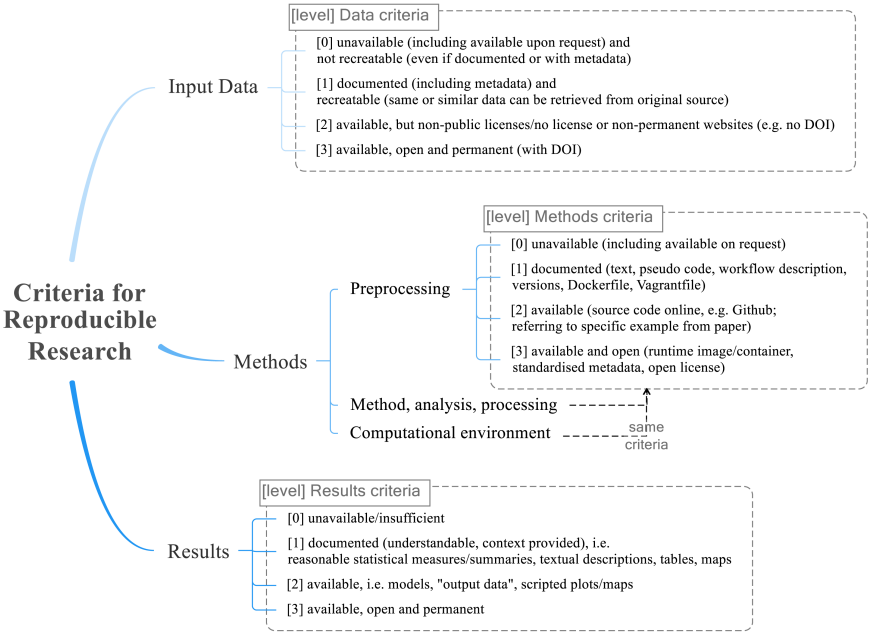


Figure A.1.: Reproducibility criteria to be assessed.

Table A.1 summarizes the self-assessment using the five criteria provided in the thesis template. The marks range from 0 to 3, where 0 means not reproducible and 3 means highly reproducible.

Overall, the reproducibility level of this thesis is assessed as partial but transparent. The main methodological logic and experimental outputs are documented, but full raw-data reproduction is not possible without access to the original station BIM/IFC data.

Table A.1.: Reproducibility self-assessment of the thesis.

Criterion	Mark	Justification
Input data	1 / 3	The raw Dadongmen Station BIM/IFC data cannot be publicly redistributed because of data-permission restrictions. The simplified pilot model, methodological description, selected processed outputs, and figures are available or documented where possible.
Preprocessing	2 / 3	The preprocessing workflow is described in the methodology, including IFC interpretation, level selection, walkable-layer extraction, obstacle filtering, node sampling, clearance filtering, and connector snapping. However, full external reproduction is limited by the unavailability of the raw station model.
Methods	3 / 3	The main methodological steps, algorithms, graph construction logic, routing strategies, congestion-aware replanning logic, and experimental design are described in the thesis. The code and configuration files are documented in the project material where redistribution is allowed.
Computational environment	2 / 3	The main tools and Python libraries are reported, including IfcOpenShell, Shapely, NetworkX, NumPy, SciPy, and Matplotlib. Exact package versions and local machine settings should be checked against the repository or environment file if available.
Results	2 / 3	The main result tables, figures, graph statistics, and experiment outputs are reported. The numerical results are reproducible from the processed graph and experiment configuration, but full end-to-end reproduction from raw IFC is restricted by data-access limitations.

A.2. Self-reflection

The main limitation for reproducibility is the input data. The Dadongmen Station BIM/IFC model was obtained from a professional project context and was used with permission for academic research. Because the model is not public project data, it is not redistributed with the thesis. This means that an external reader cannot fully rerun the complete pipeline from the raw IFC files.

To reduce this limitation, the thesis documents the workflow in detail and reports the intermediate and final outputs used in the experiments. The methodology describes how IFC-derived information is converted into 2.5D walkable layers, how graph nodes and edges are generated, how semantic origin–destination regions are attached, and how routing and simulation experiments are performed. The Chapter 6 results also report graph statistics, experiment settings, and movement-performance metrics so that the reasoning from graph construction to simulation results can be inspected.

The code, configuration files, selected processed outputs, and figure-generation material are documented in the project repository and online demonstration material where redistribution is allowed:

https://suewang291826.github.io/tud-thesis-showcase/index.html

If a separate GitHub repository is made available, it should be cited here as well:

https://github.com/SueWang291826/tud-thesis-showcase

The results should therefore be understood as reproducible at the level of method, configuration, processed outputs, and reported experiments, but not fully reproducible from the original raw BIM data. This is a common constraint when research uses professional or non-public building models. For future work, reproducibility could be improved by applying the workflow to a fully open IFC dataset or by releasing a synthetic station model that preserves the relevant graph and connector structure without disclosing professional project data.

B. Some UML diagrams

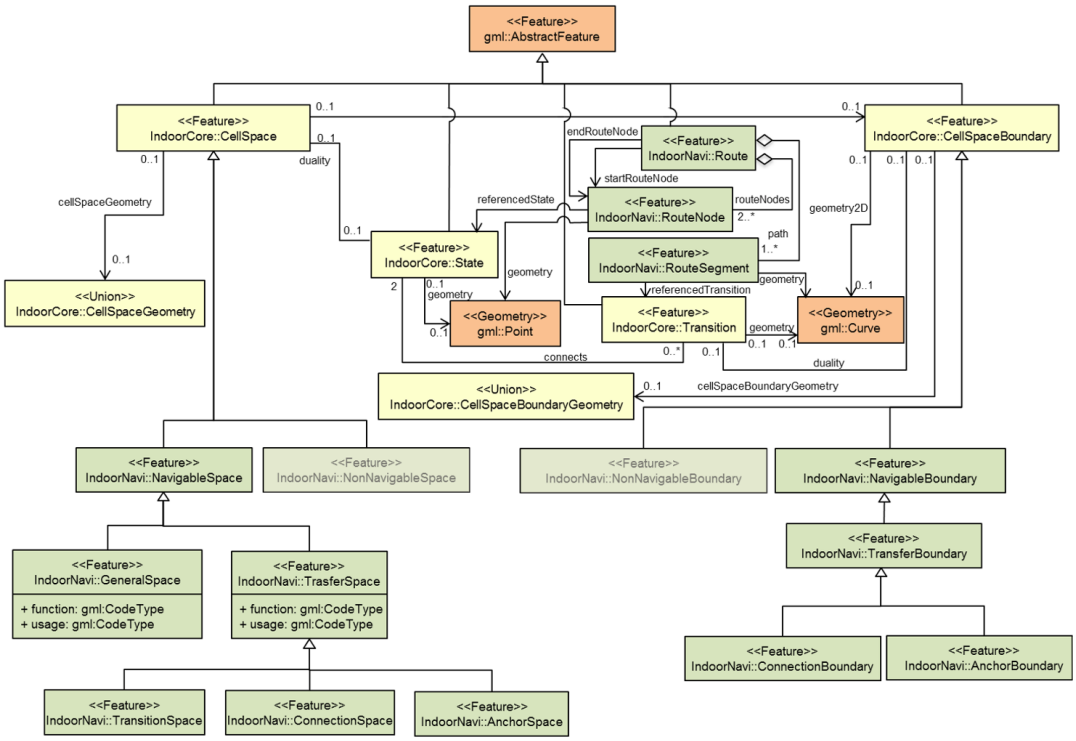


Figure B.1.: UML diagram of IndoorGML Navigation module [Open Geospatial Consortium, 2020]

C. Statement on the Use of AI-assisted Tools

AI-assisted tools were used in a controlled way during the preparation of this thesis. The tools supported implementation efficiency, writing clarity, and figure preparation. They were not used as independent sources of scientific evidence and did not replace the author's responsibility for methodology, experiments, results, or conclusions.

C.1. GitHub Copilot

GitHub Copilot was used in Visual Studio Code as a coding support tool. Its use included code completion, boilerplate generation, debugging suggestions, refactoring support, and assistance with utility scripts for processing outputs and generating figures. Copilot suggestions were manually inspected, modified, and tested before being used. The final implementation logic, parameter choices, experiment configuration, and result interpretation remained the responsibility of the author.

C.2. ChatGPT

ChatGPT was used to support thesis writing and organization. Its use included language polishing, paragraph restructuring, LaTeX formatting suggestions, improvement of figure captions, and clarification of technical explanations. ChatGPT was also used to draft prompts for schematic thesis figures and to help organize experimental results into readable tables and discussion text.

ChatGPT was not used as a source of bibliographic evidence. References suggested by AI tools were manually checked, and unverified or incomplete references were removed. The author checked the consistency of the generated text with the implemented workflow, experimental outputs, and thesis scope.

C.3. Figure-generation support

AI-assisted image generation was used for some schematic and explanatory figures, including workflow diagrams, research-question diagrams, and discussion-summary graphics. These figures were used as visual communication aids rather than as experimental evidence. Quantitative result figures were generated from experiment outputs or manually verified values. Where AI-assisted figures were used, their content, labels, and numerical values were checked and corrected by the author before inclusion.

C.4. Gemini and presentation preparation

Gemini was used only for presentation preparation and visual-communication brainstorming. It was not used to generate experimental results, analyse raw data, or provide thesis evidence.

C.5. Author responsibility

All modelling decisions, code outputs, experimental results, figures, tables, and interpretations were checked and revised by the author. The AI-assisted tools supported drafting, coding efficiency, and communication, but the final responsibility for the thesis content remains with the author.

Bibliography

Bhardwaj, R., Bhargava, A., and Kumar, V. (2024). INFED: Enhancing fire evacuation dynamics through 3D congestion-aware indoor navigation framework. *Simulation Modelling Practice and Theory*, 136:103010. doi: 10.1016/j.simpat.2024.103010.

China Metro (2024). Hefei metro transported 9.7976 million passengers during the 2024 may day holiday. https://www.chinametro.net/index.php?aid=59762&id=539&m=mobilenewscon. Original title in Chinese; accessed 2026-05-18.

Diakite, A. A., Diaz-Vilarino, L., Biljecki, F., Isikdag, U., Simmons, S., Li, K.-J., and Zlatanova, S. (2022). IFC2IndoorGML: An open-source tool for generating IndoorGML from IFC. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XLIII-B4-2022, pages 295–301. doi: 10.5194/isprs-archives-XLIII-B4-2022-295-2022.

Diakite, A. A. and Zlatanova, S. (2016). Extraction of the 3D free space from building models for indoor navigation. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-2/W1, pages 241–248. doi: 10.5194/isprs-annals-IV-2-W1-241-2016.

Fu, X., Zhang, H., and Wang, P. (2021). Automatic construction of indoor 3D navigation graph from crowdsourcing trajectories. *ISPRS International Journal of Geo-Information*, 10(3):146. doi: 10.3390/ijgi10030146.

Hamzei, E., De Cock, L., Tomko, M., Van de Weghe, N., and Winter, S. (2024). Indoor view graph: A model to capture route and configurational information. *Environment and Planning B: Urban Analytics and City Science*, 51(9):2213–2231. doi: 10.1177/23998083241241598.

Han, L., Qiao, H., Li, Z., Liu, M., and Zhang, P. (2023). Navigation-oriented topological model construction algorithm for complex indoor space. *ISPRS International Journal of Geo-Information*, 12(6):248. doi: 10.3390/ijgi12060248.

Hefei Yaohai District Government (2018). Hefei metro passenger flow reached a new high: Hefei south railway station, hefei railway station, and dadongmen station had the highest passenger flows. https://www.hfyaohai.gov.cn/zwdt/mtbd/9849681.html. Original title in Chinese; accessed 2026-05-18.

Isikdag, U., Zlatanova, S., and Underwood, J. (2013). A BIM-oriented model for supporting indoor navigation requirements. *Computers, Environment and Urban Systems*, 41:112–123. doi: 10.1016/j.compenvurbsys.2013.05.001.

Kang, H.-K. and Li, K.-J. (2017). A standard indoor spatial data model—OGC IndoorGML and implementation approaches. *ISPRS International Journal of Geo-Information*, 6(4):116. doi: 10.3390/ijgi6040116.

Khan, D., Cheng, Z., Uchiyama, H., Ali, S., Asshad, M., and Kiyokawa, K. (2022). Recent advances in vision-based indoor navigation: A systematic literature review. *Computers & Graphics*, 104:24–45. doi: 10.1016/j.cag.2022.03.005.

Lewandowicz, E. and Lisowski, P. (2019). A modified methodology for generating indoor navigation models. *ISPRS International Journal of Geo-Information*, 8(2):60. doi: 10.3390/ijgi8020060.

Liu, L., Li, B., Zlatanova, S., and van Oosterom, P. (2021a). Indoor navigation supported by the industry foundation classes (IFC): A survey. *Automation in Construction*, 121:103436. doi: 10.1016/j.autcon.2020.103436.

Liu, T., Li, H., Lu, H., Cheema, M. A., and Shou, L. (2021b). Towards crowd-aware indoor path planning. *Proceedings of the VLDB Endowment*, 14(8):1365–1377. doi: 10.14778/3457390.3457401.

Liu, X., He, C., Zhao, H., Jia, J., and Liu, C. (2021c). Building information modeling indoor path planning: A lightweight approach for complex BIM building. *Computer Animation and Virtual Worlds*, 32(3–4):e2014. doi: 10.1002/cav.2014.

Nikoohemat, S., Diakite, A. A., Zlatanova, S., and Vosselman, G. (2020). Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. *Automation in Construction*, 113:103109. doi: 10.1016/j.autcon.2020.103109.

Open Geospatial Consortium (2020). OGC IndoorGML 1.1. https://docs.ogc.org/is/19-011r4/19-011r4.html. OGC Implementation Standard, OGC 19-011r4; accessed 2026-05-18.

Senanayake, G. P. D. P., Kieu, M., Zou, Y., and Dirks, K. (2024). Agent-based simulation for pedestrian evacuation: A systematic literature review. *International Journal of Disaster Risk Reduction*, 111:104705. doi: 10.1016/j.ijdrr.2024.104705.

Sina Finance (2024). Hefei metro announced that cumulative passenger volume exceeded 1.7 billion. https://finance.sina.com.cn/jjxw/2024-05-06/doc-inauhrwu8886859.shtml. Original title in Chinese; accessed 2026-05-18.

Tan, J. K. N., Zhang, S., Law, A. W.-K., and Cheung, S. H. (2025). Digital-twin enabled evacuation to improve individual and community resilience of building occupants against indoor fires: Framework and route-finding algorithm. *Developments in the Built Environment*, 22:100672. doi: 10.1016/j.dibe.2025.100672.

Valizadeh, M., Ranjgar, B., Niccolai, A., Hosseini, H., Rezaee, S., and Hakimpour, F. (2024). Indoor augmented reality (AR) pedestrian navigation for emergency evacuation based on BIM and GIS. *Heliyon*, 10(12):e32852. doi: 10.1016/j.heliyon.2024.e32852.

Wong, M. O. and Lee, S. (2023). Indoor navigation and information sharing for collaborative fire emergency response with BIM and multi-user networking. *Automation in Construction*, 148:104781. doi: 10.1016/j.autcon.2023.104781.

Xie, R., Zlatanova, S., Aleksandrov, M., and Lee, J. (2024). A voxel-based 3D indoor model to support 3D pedestrian evacuation simulations. *Journal of Building Engineering*, 98:111183. doi: 10.1016/j.jobe.2024.111183.

Xu, M., Wei, S., Zlatanova, S., and Zhang, R. (2017). BIM-based indoor path planning considering obstacles. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-2/W4, pages 417–423. doi: 10.5194/isprs-annals-IV-2-W4-417-2017.

Zhang, W., Li, Y., Li, P., and Feng, Z. (2025). A BIM and AR-based indoor navigation system for pedestrians on smartphones. *KSCE Journal of Civil Engineering*, 29(1):100005. doi: 10.1016/j.kscej.2024.100005.

Zhen, W., Yang, L., Kwan, M.-P., Zuo, Z., Qian, H., and Zhou, S. (2020). Generating comfortable navigable space for 3D indoor navigation considering users' dimensions. *Sensors*, 20(17):4964. doi: 10.3390/s20174964.

Zhou, Z., Weibel, R., Richter, K.-F., and Huang, H. (2022). HiVG: A hierarchical indoor visibility-based graph for navigation guidance in multi-storey buildings. *Computers, Environment and Urban Systems*, 93:101751. doi: 10.1016/j.compenvurbsys.2021.101751.

Zhu, J., Wong, M. O., Xu, J., Nisbet, N., Kelly, T., Zlatanova, S., and Brilakis, I. (2025). Semantics-based connectivity graph for indoor pathfinding powered by IFC-graph. *Automation in Construction*, 171:106019. doi: 10.1016/j.autcon.2025.106019.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.