VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

INFORMATICS MASTER'S STUDY PROGRAMME

Master's Thesis

Simulation-Based Analysis of Flow and Delays in Dynamic Service Systems Using Discrete Event Simulation

Simuliacija Pagrįsta Dinaminių Paslaugų Sistemų Srautų ir Vėlavimų Analizė Naudojant Diskretinių įvykių Simuliaciją

Muhammad Wadeed Madni

Supervisor: Prof. Dr. Linas Laibinis
Reviewer: Dr. Julius Andrikonis

Spring 2026

Table of Contents

Table of Figures

List of Tables

Abstract

This thesis analyses flow and delays in a dynamic public transport service system using discrete-event simulation. A simplified hub-based bus service model was implemented in AnyLogic, where passengers arrive stochastically, wait in a queue, and are served by buses with limited capacity. The initial model was extended with probabilistic disruption, dynamic bus allocation policies, fairness constraints, and cold-weather sensitivity.

The main objective is to evaluate how disruptions affect passenger time outcomes and whether dynamic bus allocation can reduce the probability of excessive delays. Several scenarios were compared, including normal operation, disrupted operation, greedy allocation, fairness-protected allocation, and weather-adaptive allocation. Passenger time samples were collected from simulation runs and analysed using Python through summary statistics, probability distributions, cumulative distribution functions, percentiles, and threshold exceedance probabilities.

The results show that disruption significantly worsens passenger time outcomes. In the disrupted baseline scenario, the mean passenger time increased to 60.70 minutes, and 52.0% of samples exceeded 60 minutes. When dynamic allocation was accepted, the mean decreased to 25.43 minutes, and the probability of exceeding 60 minutes decreased to 0.0%. The results demonstrate that dynamic service adjustment can reduce extreme delays, but fairness and weather constraints are necessary to avoid negatively affecting passengers on other routes.

Keywords: discrete-event simulation, AnyLogic, public transport, dynamic bus allocation, passenger delay, probability distribution.

Santrauka

Šiame baigiamajame darbe analizuojami srautai ir vėlavimai dinaminėje viešojo transporto paslaugų sistemoje, taikant diskrečiųjų įvykių modeliavimą. AnyLogic aplinkoje buvo sukurtas supaprastintas autobusų paslaugos modelis su centrine stotele, kuriame keleiviai atvyksta stochastiškai, laukia eilėje ir yra aptarnaujami ribotos talpos autobusų. Pradinis modelis buvo išplėstas įtraukiant tikimybinį trikdžių mechanizmą, dinaminio autobusų paskirstymo politikas, sąžiningumo apribojimus ir jautrumą šalto oro sąlygoms.

Pagrindinis darbo tikslas yra įvertinti, kaip trikdžiai veikia keleivių laiko rezultatus ir ar dinaminis autobusų paskirstymas gali sumažinti pernelyg ilgų vėlavimų tikimybę. Buvo palyginti keli scenarijai: įprastas veikimas, sutrikdytas veikimas, gobšus paskirstymas, sąžiningumu pagrįstas paskirstymas ir oro sąlygoms prisitaikantis paskirstymas. Simuliacijos metu surinkti keleivių laiko mėginiai buvo analizuojami naudojant Python, taikant aprašomąją statistiką, tikimybių skirstinius, sukauptąsias tikimybių funkcijas, procentilius ir slenkstinių reikšmių viršijimo tikimybes.

Rezultatai parodė, kad trikdžiai reikšmingai pablogina keleivių laiko rodiklius. Sutrikdyto bazinio scenarijaus atveju vidutinis keleivio laikas padidėjo iki 60,70 minutės, o 52,0% mėginių viršijo 60 minučių. Kai dinaminis paskirstymas buvo leidžiamas, vidurkis sumažėjo iki 25,43 minutės, o tikimybė viršyti 60 minučių sumažėjo iki 0,0%. Rezultatai rodo, kad dinaminis paslaugos koregavimas gali mažinti ekstremalius vėlavimus, tačiau sąžiningumo ir oro sąlygų apribojimai yra būtini siekiant nepabloginti situacijos keleiviams kituose maršrutuose.

Raktiniai žodžiai: diskrečiųjų įvykių modeliavimas, AnyLogic, viešasis transportas, dinaminis autobusų paskirstymas, keleivių vėlavimas, tikimybių skirstinys

1. Introduction

The world has seen a notable increase in its total population over the past several decades. Starting from around 2.5 billion people in 1950, the global population has grown to more than 8 billion today [Uni24]. Current projections from the United Nations suggest that this population growth is likely to continue for a number of years, with the median estimate indicating a peak around 10.4 billion in the 2080s before stabilizing or slightly declining towards the end of the century (Figure 1).

This continuing population growth, combined with the well-established global trend of urbanization, meaning more people are choosing to live in cities, creates significant and increasing demands on city infrastructure and public services. As urban areas become home to more inhabitants, the need for adequate housing, reliable water and energy supplies, and particularly, efficient transportation systems, becomes more pressing. This trend of a growing global population residing more in urban centers directly underscores the increasing importance of creating effective and sustainable urban mobility solutions. In this context, the crucial role of public transport services within a successful transportation system is widely acknowledged [Ver82], [May95]. This importance is rooted in several key factors. A significant reason is public transport's contribution to long-term sustainability, particularly concerning resource consumption and environmental protection [Ban97]. Equally vital is its function in providing an essential transportation option for individuals who do not have access to a private motor vehicle. Furthermore, public transport offers a valuable travel alternative for commuters, which can help reduce pressure on existing road infrastructure. These combined contributions provide a strong motivation for prioritizing and developing robust public transport services in urban areas.

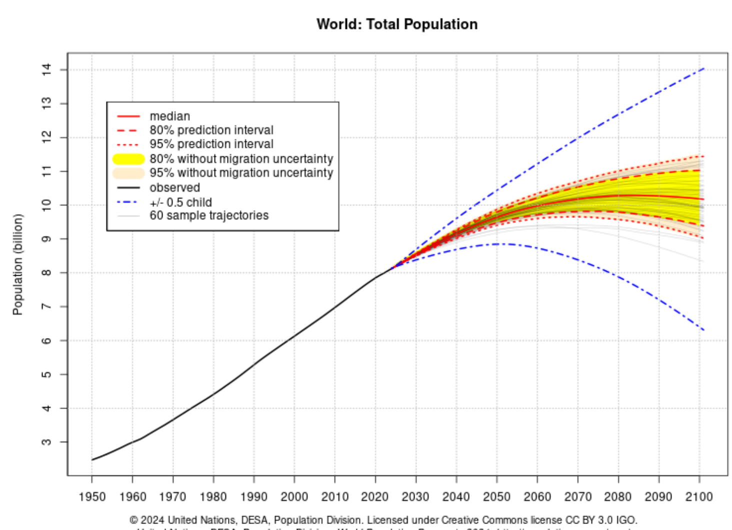

Figure 1 World Total Population: Observed (1950-2024) and Probabilistic Projections to 2100. Source: United Nations (2024): World Population Prospects 2024

As city populations grow, people naturally need easy access to various activities such as businesses, schools, jobs, and places for leisure. Deciding where to place these important services and ensuring there is good transportation infrastructure, including things like major roads, extensive public transit networks, and enough parking, is a central part of good urban planning. The way a transportation system is designed and managed has a major effect on how a region develops, its economic health, its impact on the environment, and the quality of life for the people living there. It makes sense, then, that government bodies often invest significant resources in planning and creating better transportation services to address these many urban requirements [Mur98]. In this context, making sure that public transportation is efficient, easy for everyone to use, and able to meet changing demands is a very important goal.

Despite these clear benefits, a fundamental challenge in managing public transport systems effectively is the highly variable nature of passenger demand. This demand is not evenly distributed; instead, it changes significantly across different locations within the network and at different times of the day or week [Tsa12]. For example, routes serving central business districts typically experience sharp increases in passenger numbers during morning and evening commute periods, often referred to as peak hours, while demand may be substantially lower during midday or on weekends. Furthermore, specific events, such as sporting fixtures or cultural festivals, can cause unpredictable, short-term surges in passenger numbers at particular stations or along certain corridors [Mar24]. This inherent variability, which can be influenced by diverse factors including work schedules, school calendars, and local land use patterns, makes it difficult for transport operators to perfectly match service supply with actual passenger needs at all times using traditional, fixed schedules.

When passenger demand at key locations, such as major bus hubs or city center stops, exceeds the available service capacity, one of the most direct and noticeable consequences for users is an increase in waiting times. Passengers may find themselves queuing for extended periods, often in uncomfortable or exposed conditions, before they are able to board a bus. These prolonged waits are a significant source of frustration and can lead to a negative perception of the overall public transport service quality. Moreover, unpredictable and long waiting times diminish the reliability of the service, making it difficult for passengers to plan their journeys effectively and potentially causing them to be late for work, appointments, or other commitments. If passengers consistently experience such delays, it can ultimately discourage them from using public transport, leading them to seek alternative, often less sustainable, modes of travel [Cos25].

1.1 Problem Statement

While public transport is clearly essential for cities, systems that operate with fixed routes and unchanging schedules often struggle to effectively manage the highly variable passenger demand. This lack of operational flexibility becomes a significant issue, especially during peak hours or at specific busy locations like major hubs. In these situations, the inability of the system to quickly adapt its services when there's a sudden increase in passenger numbers leads to several problems:

1. **Overcrowding at hubs:** Certain stops, particularly hubs, become severely overcrowded. This reduces passenger comfort and can create potential safety issues.
2. **Long and Unreliable Waiting Times:** When passenger demand at these congested points is higher than the scheduled service can handle, people face extended and often unpredictable waits. This causes frustration and a poor overall travel experience.
3. **Inefficient Use of Bus Fleet:** At the same time, buses on other routes or during less busy periods might be operating with very few passengers. This indicates that the available bus fleet and resources are not being used as effectively as possible.

Therefore, the central problem this research addresses is that conventional public transport systems often lack the ability to dynamically adjust to balance passenger loads and manage localized congestion effectively, especially during peak hours at important hubs. This results in a less than ideal experience for passengers at crowded locations and means the system's resources are not deployed in the most efficient manner.

To rectify these issues, more flexible and adaptive operational methods are needed. Discrete Event Simulation (DES) offers a valuable approach for exploring such methods, as simulation environments like AnyLogic [Any26a] provide a powerful and flexible platform to model these complex interactions, test different rerouting rules, and evaluate their impact on system performance before considering real world implementation.

1.2 Research Aim and Objectives

1.2.1 Research Aim

The main aim of this research is to investigate how dynamic bus rerouting strategies can be used to lessen passenger crowding at busy hubs and improve the overall flow of public transport services, particularly during peak hours. This will be explored by developing and testing these strategies within a Discrete Event Simulation model, developed within the AnyLogic framework [Any26b], of a simplified public transport system.

1.2.2 Research Objectives

To achieve this aim, the following specific objectives will be addressed:

1. To create a conceptual model[1] of a basic multi-stop public transport system, and identify its main system parameters such as bus capacities and passenger arrival rate that directly influence its behavior and quantitative characteristics, while paying close attention to how passenger demand increases at a designated hub during peak hours.

[1] visual representation of the different variables in research and the relationship between them.
source: https://web.stanford.edu/group/ncpi/unspecified/student_assess_toolkit/conceptualModels.html

2. To design the rules and logic for at least two different dynamic bus rerouting strategies that are triggered by real-time conditions at the hub, such as long queues or extended waiting times.
3. To build this conceptual model and the proposed rerouting strategies into a working Discrete Event Simulation (DES) environment using the AnyLogic software.
4. To select and define Key Performance Indicators (KPIs) or metrics that can measure how well the system performs. These will include passenger waiting times (especially at the hub), how fully buses are utilized, and how often rerouting occurs.
5. To run a series of simulation experiments within the AnyLogic framework. These experiments will compare the system's performance when using the different dynamic rerouting strategies against how it operates normally (the baseline scenario without any rerouting).
6. To carefully analyze the results from the simulations to determine if, and under what conditions, dynamic rerouting effectively reduces congestion at hubs and to understand any associated trade-offs for the wider system.

1.3 Research Questions

This research seeks to answer the following questions:

1. To what extent can implementing dynamic bus rerouting strategies reduce passenger waiting times at a heavily used hub during peak hours?
2. What are the effects of these dynamic rerouting actions on the overall performance of the public transport system, considering factors like potential delays on routes from which buses are diverted and changes in overall bus utilization?
3. Under which specific circumstances (such as different levels of congestion at the hub or varying passenger arrival rates) dynamic rerouting is most beneficial, and are there situations where it might be less effective or even detrimental?
4. If multiple rerouting strategies are examined, which approach offers the best overall balance between improving service quality at the congested hub and maintaining efficient use of resources across the entire system?

1.4 Research Methodology and Tools

This research will adopt a quantitative methodology, with Discrete Event Simulation (DES), using the AnyLogic Software, serving as the primary analytical method to investigate the effectiveness of dynamic bus rerouting. A conceptual model of the transport system is first defined, incorporating the primary system parameters identified in Research Objective 1 (Section 1.2.2) specifically, bus seat capacities and stochastic passenger arrival rates, which serve as the independent variables driving the discrete event logic. This conceptual model, an artificial system, inspired by common real-world scenarios of hub congestion and resource constraints, will then be implemented using

AnyLogic simulation software. The model is inspired by public transport hub behaviour in an urban context such as Vilnius, but it does not reproduce the complete Vilnius public transport network. Instead, it uses a simplified and parameterized structure to analyse congestion, delay, and dynamic service adjustment under controlled simulation conditions.

Simulation experiments will be conducted to compare the performance of different rerouting strategies against a baseline scenario as mentioned in 1.3.2 - 5. Key Performance Indicators (KPIs) or metrics, such as passenger waiting times at the hub, bus utilization, and queue lengths, will be collected and analyzed using appropriate statistical methods.

This approach will allow for a controlled evaluation of the proposed strategies and provide evidence-based insights into their potential to avoid congestion and improve system efficiency before any consideration of real-world application when using the AnyLogic Simulation Software.

1.5 Data Generation and Extraction Protocol

In this study, data collection is engineered as a high-resolution event-logging pipeline. Within the AnyLogic environment, each passenger agent functions as a discrete data carrier. Upon system entry at a hub node, a 'timestamp variable' is assigned to the agent. At the point of system egress (terminal drop-off), a Java-based listener calculates the delta between departure and arrival times. This granular event data, including unique PassengerID, ArrivalTime, and PassengerTime, is streamed in real-time to a structured CSV dataset via the AnyLogic Connectivity Library. This methodology ensures that the quantitative analysis is based on raw, per-agent transaction records rather than estimated system-wide averages.

2. Theoretical Background and Literature Review

This chapter establishes the theoretical and methodological foundation for the research. It is structured into three main sections. The first section provides a general overview of the problem area, characterizing public transport networks as a class of dynamic service systems and exploring them through the lens of Agent-Based Modeling. The second section reviews and compares common methods for system analysis, arguing for the selection of Discrete Event Simulation (DES) and describing its core principles. The final section introduces the specific software tool chosen for this research, AnyLogic, detailing its key features and advantages for this study.

2.1. Dynamic Service Systems

2.1.1. The Nature of Dynamic Service Systems

To properly analyze the flow and delays within a public transport network, it is necessary to first understand the broader class of systems to which it belongs. Many of the operational challenges faced in modern society, from managing patient flow in a hospital emergency room to balancing data loads on a cloud server, can be fundamentally understood as Dynamic Service Systems.

A service system is any configuration of technology, people, and other resources that interact to provide a service to a stream of incoming users [Fit11]. These systems are characterized as "dynamic" because they must constantly adapt to a demand that changes over time and is rarely stable or perfectly predictable [Kel15]. Regardless of the specific industry, whether it is healthcare, banking, or transportation, these systems share a common structural framework:

1. **Entities (Users):** These are the items or people arriving to request a service (e.g., patients, customers, data packets, or passengers).
2. **Resources (Servers):** These are the limited assets available to process the entities (e.g., doctors, tellers, CPUs, or buses).
3. **Queues (Buffers):** This is the waiting area that naturally forms when the demand temporarily exceeds the available capacity of the resources.

While this research focuses specifically on public transport, the underlying operational logic is identical to these general examples. By viewing the transport network as a standard dynamic service system, it becomes possible to apply established theories of flow management and capacity planning that have been proven in other sectors [Fit11], [Alt07].

2.1.2. A General Modeling Perspective: Agent-Based Systems (ABS)

To properly understand and analyze dynamic service systems, it is beneficial to adopt a general modeling perspective that captures their inherent complexity. One powerful and increasingly common perspective is **Agent-Based Modeling (ABM)**, also known as Agent-Based Simulation (ABS) [Abb16], [Zhe13]. This approach represents a shift from traditional, top-down modeling, where system-wide equations dictate behavior [Tur16]. Instead, ABM utilizes a "bottom-up" approach, where the large-scale behavior of the entire system is an emergent result of the actions and interactions of its individual components [Tur16], [Mis25].

At the heart of an ABM is the concept of an "agent", an autonomous, decision-making entity with its own set of behaviors and characteristics[Bon02]. Each agent assesses its local situation and acts according to its own rules, interacting with other agents and its environment [Tra18]. The overall system behavior is not explicitly programmed; rather, it emerges from the collection of these numerous individual interactions [Bon02], [Zhe06]. This makes ABM a particularly suitable framework for exploring complex adaptive systems in which decentralized decisions lead to system-wide patterns [Gao24].

In the general context of a service system, different entities can be effectively defined as agents:

1. **Customers or Users:** These are agents with individual goals (e.g., to receive a service, to purchase a product) and attributes (e.g., level of patience, service preference). In a public transport model, these would be passengers. In a healthcare model, they would be patients.
2. **Service Providers or Resources:** These can also be modeled as agents with their own state and behavior. A bus in a transport network, a doctor in a hospital, or a cashier at a bank can all be considered agents that provide a service according to a set of operational rules.
3. **Supervisory or Coordinating Agents:** In more complex models, there can be agents that represent higher-level control, such as a dispatch center in a logistics network or a head nurse in a hospital ward, making operational decisions.

By defining a system in terms of these interacting agents, ABM provides a natural way to model the decentralized nature of many real-world service systems, making it a valuable perspective for analysis.

2.1.3. Essential Properties and Parameters of Such Systems

When modeling dynamic service systems, it is critical to identify the specific properties that dictate their behavior. These properties explain why static mathematical calculations often fail to predict delays accurately and why simulation is required. Three core characteristics are shared across this entire class of systems:

1. **Stochastic Demand (Randomness):** The most significant source of complexity in these systems is stochasticity, or inherent randomness. Unlike deterministic systems, such as manufacturing assembly lines where events occur at fixed and known intervals, dynamic service systems are driven by events that happen randomly [Ros09]. In a general context, this unpredictability is evident in sectors like healthcare or banking; a hospital administrator cannot predict the exact moment an ambulance will arrive, nor can a bank manager foresee exactly when a customer will enter the branch.

In the specific context of this research, this stochasticity manifests in the arrival patterns of passengers at a bus hub. While general peak hours can be identified, the exact arrival time of each individual passenger varies significantly. This inherent randomness often leads to "bunches" or sudden surges in demand that rigid, pre-planned schedules fail to accommodate effectively, thereby necessitating a dynamic approach to management.

2. **Resource Contention:** The second defining characteristic is resource contention, which occurs when multiple entities compete simultaneously for the same limited resource [Sri15]. This competition is the primary constraint that limits system performance. In technical environments, such as cloud computing, this is observed when multiple software processes compete for the processing power of a single CPU core, causing performance lag.

   Within a public transport network, resource contention is observed when the number of waiting passengers exceeds the specific seating capacity of an arriving bus. Here, the bus acts as the contended "resource." When capacity is reached, the remaining "users" (passengers) are forced to wait for the next service cycle. This competition for limited capacity is a fundamental friction point in the system that directly impacts service reliability and passenger satisfaction**.**

3. **Queuing Phenomena:** The interaction between stochastic demand and resource contention inevitably leads to queuing, or the formation of waiting lines [Gro85]. Queuing theory provides the mathematical framework for analyzing these lines, which serve as the most visible indicator of system performance. While queues are a common feature in many service industries, from call centers holding callers in a virtual line to supermarkets managing physical checkout lines, they exhibit unique properties in transportation.

   For the purpose of this simulation, the queue represents passengers waiting at a specific stop or hub. Unlike a supermarket queue which typically moves at a steady pace, transport queues are dynamic and deplete in "batches" whenever a bus arrives. Understanding the formation, growth, and dissipation of these queues is the central objective of the simulation model, as the queue length directly correlates with the passenger waiting time [Gro85].

2.1.4. Classification of Model Parameters

To control and analyze the behavior of these systems within a simulation, the model is defined by a set of **parameters**. These parameters are the factors and variables that influence agent decisions and the evolution of the system [Ant24], [Ols22]. For the general class of dynamic service systems, these can be categorized as follows:

1. **System-level Parameters:** These define the overall rules and constraints of the environment. Examples include the total number of available resources (like the number of buses in a fleet), the system's maximum capacity, or global policies that all agents must follow.
2. **Agent-level Parameters:** Each agent has its own set of parameters that define its individual characteristics and attributes [Kha24]. For a passenger agent, this could be their walking speed or level of patience. For a bus agent, parameters might include seating capacity, acceleration speed, or maintenance schedule [Bro21].
3. **Behavioral Parameters:** These are the rules that govern how agents interact with each other and their environment [Bro21]. This includes the logic for decision-making, such as the criteria a passenger uses to choose a service or the rules a bus follows when its route is obstructed. Changes to these parameters can significantly affect the model's outcomes [Ols24].

Understanding and identifying these properties and parameters is a critical first step in the modeling process. It ensures that the subsequent simulation is grounded in a solid conceptual framework that accurately reflects the dynamics of the system being investigated [Thi14].

2.2. Methods for System Analysis and Modeling

2.2.1. Overview of Analytical and Simulation Methods

To analyze the behavior of dynamic service systems, researchers and practitioners primarily use two broad categories of methods: analytical methods and simulation [Asl20]. Analytical methods involve using mathematical theory, such as queuing theory or operations research algorithms, to build equations that describe the system and yield exact solutions for performance measures. For example, basic queuing theory can predict average waiting times for simple systems under specific, stable conditions [Mit25]. However, the main limitation of these analytical approaches is that they often rely on simplifying assumptions, such as purely random customer arrivals or constant service times, that may not hold true in many real-world situations. When systems become complex with many interdependencies, randomness, and non-standard behaviors, finding an exact mathematical solution becomes difficult or impossible [PTC26].

This is where computer simulation offers a powerful and practical alternative. Simulation solves real-world problems by imitating a system's operation over time, providing insights into its behavior without needing a formal mathematical solution. By replicating a real or proposed system in a virtual environment, simulation can handle the complexity and uncertainty that analytical methods often cannot [Fiv26], [Eur24]. One of the primary benefits of simulation is the ability to test different "what-if" scenarios in a risk-free environment, allowing analysts to understand the potential effects of changes before they are implemented. It is also more cost-effective and a faster alternative to conducting experiments on a real system. Because of its flexibility in handling complexity and its utility as a tool for decision-making, simulation has become an essential method for analyzing and improving dynamic service systems [Spe26].

2.2.2. Major Simulation Paradigms

Within the field of computer simulation, there are three predominant modeling paradigms, each offering a different perspective and level of abstraction for analyzing systems: System Dynamics (SD), Discrete Event Simulation (DES), and Agent-Based Modeling (ABM) [Any26c], [Noo18]. The choice of method is typically based on the characteristics of the system being modeled and the specific goals of the analysis [Ngo20].

1. **System Dynamics (SD):** This approach models a system at a highly aggregate, or macro, level. It focuses on feedback loops and uses stocks and flows, often represented by differential equations, to understand how system behavior changes over time. SD is generally applied to flowing, continuous processes and is best suited for strategic, long-term analysis rather than modeling individual entities [Sim26].
2. **Agent-Based Modeling (ABM):** As discussed previously, ABM takes a "bottom-up" view, focusing on the autonomous, decision-making behavior of individual agents within a system [Gil08]. System-wide patterns emerge from the dynamic interactions of many individual agents according to their own rules. This paradigm is particularly useful for modeling systems where individual adaptation and emergent behavior are key components[Mac09].
3. **Discrete Event Simulation (DES):** This approach models a system's operation as a chronological sequence of discrete events. The system's state only changes when a specific event occurs [Ext26]. DES is highly process-centric, focusing on how entities (like passengers or products) flow through a sequence of processes, wait in queues, and compete for limited resources. It excels at modeling predefined processes where the detailed flow of individual entities and resource contention are the main points of interest [Maj16].

2.2.3. A Detailed Examination of Discrete Event Simulation (DES)

For the purpose of this research, Discrete Event Simulation (DES) was selected as the primary modeling paradigm. While the public transport system possesses agent-like characteristics, the core problem defined in this thesis, passenger queuing at hubs and the allocation of buses to serve them align well with the process-centric strengths of DES. To fully justify this selection, it is necessary to examine the internal mechanics of DES and the methodological steps required to implement it.

2.2.3.1. The Operational Mechanics of DES

Unlike continuous simulation, which models time as a smooth, flowing variable, DES views the world as a sequence of distinct events. The system state remains unchanged between these events. This efficiency is achieved through three fundamental components [Ban10]**.**

1. **The Simulation Clock**: In a DES model, the clock does not tick second-by-second. Instead, it utilizes a "next-event" time advance mechanism. If a bus arrives at 08:00 AM and the next scheduled event (e.g., a passenger arrival) is at 08:05 AM, the

simulation clock skips the intervening five minutes instantly. This allows the simulation to model long periods of operation, such as weeks or months of bus schedules, in a matter of seconds, without wasting computational power on idle time [Law15].

2. **The Event List:** This is the "engine" of the simulation. The software maintains a priority queue of all future scheduled events, ordered chronologically.
    a. **Example:** The list might contain: [Event A: Bus 101 Arrives at 08:00], [Event B: Passenger 5 Boards at 08:02].
    b. The simulation removes the top event, updates the system state (e.g., reduces the number of people in the queue), and then advances the clock to the time of the next event.
3. **State Variables:** These are the numerical values that describe the condition of the system at any given point in time. In this research, critical state variables include the current queue length, the number of passengers on board, and the status of the bus (idle, moving, or boarding) [Ban10].

2.2.3.2. The Simulation Study Lifecycle

Implementing a DES model is not merely a coding exercise; it follows a rigorous methodological lifecycle to ensure validity. According to standard simulation literature, a successful study must proceed through specific phases [Ban10], [Sar10]:

1. **Problem Formulation:** Clearly defining the goals of the study, in this case, reducing waiting times at the hub.
2. **Data Collection:** Gathering real-world input data. DES is highly data-dependent; the accuracy of the output (waiting times) is entirely dependent on the accuracy of the input (passenger arrival rates).
3. **Model Translation:** The process of converting the conceptual model into computer code or a flowchart within the simulation software.
4. **Verification and Validation (V&V):** This is the most critical step for academic rigor.
    a. Verification ensures the model is built correctly (checking for bugs).
    b. Validation ensures the model behaves like the real system. For a transport model, this might involve comparing the simulated queue lengths against historical data to ensure they match reality.
5. **Experimentation:** Once validated, the model is used to run "what-if" scenarios, such as *"What happens to the queue if we reroute three extra buses during peak hour?"*.

The main advantage of DES for this type of research is its ability to naturally and flexibly model systems with discrete changes [Car16]. It excels at replicating complex real-world operations and can easily integrate multiple stochastic elements, making it ideal for creating predictive models of complex systems [Lyu23]. Analytical mathematical formulas can calculate the average wait time for a simple, steady queue. However, they often fail when introduced to complex, real-world logic, such as "buses only reroute if the queue exceeds 50 people and the time is after 5:00 PM." DES handles these complex logic rules effortlessly.

This allows decision-makers to identify process bottlenecks and optimize resource allocation before implementing changes in the real world [Aut25].

However, DES also has limitations. It can be suboptimal for modeling systems that change continuously, and large-scale discrete event simulations can consume a considerable amount of computer processing time [Ndi15].

2.3 Simulation Software Frameworks

Following the selection of Discrete Event Simulation as the methodological approach, the next step is to identify the most suitable software environment for implementation. The simulation software market is diverse, offering a wide range of tools that vary significantly in their modeling approach, flexibility, and visualization capabilities. To ensure the selected tool is appropriate for the general class of dynamic service systems, which require both complex decision-making logic and visual verification of queues, it is necessary to evaluate the strengths and limitations of the major software categories.

2.3.1. Overview of Simulation Tool Categories

Simulation tools are generally categorized into two distinct types: **code-centric frameworks** and **Visual Interactive Simulation (VIS**) environments:

1. **Code-Centric Frameworks (e.g., SimPy**): These frameworks represent a programming-first approach. A prominent example is SimPy, a process-based discrete-event simulation library based on the Python programming language[2]. The primary strength of such frameworks is their flexibility; because the model is constructed entirely in code, there are practically no constraints on the logic or the behavior of the agents. This makes them highly suitable for theoretical research involving complex, non-standard algorithms. However, a significant limitation for operational research is the absence of a native visual interface. To observe the flow of entities or verify the formation of queues, the researcher must essentially build a visualization engine from scratch or rely on analyzing abstract text logs. This lack of immediate visual feedback makes the verification of the model's behavior significantly more difficult [52].
2. **Visual Interactive Simulation Environments (e.g., Arena):** In contrast, tools like Arena (or Simio) represent the industry standard for manufacturing and logistics. These tools utilize a flowchart-based modeling approach, where the user constructs the system by dragging and dropping pre-defined logic blocks (such as "Create," "Process," or

[2] https://simpy.readthedocs.io/en/latest/

"Dispose") onto a graphical canvas [Kel15]. The main advantage of this approach is the speed of development for standard, linear processes and the provision of immediate, built-in animation. However, these environments often lack the flexibility required for complex logic. Implementing a dynamic rule, such as rerouting a bus based on the real-time congestion level of a distant stop, can be difficult to achieve within the rigid constraints of pre-built flowchart blocks [Alt07].

2.3.2. Rationale for Tool Selection

For the analysis of dynamic service systems conducted in this research, the ideal tool must bridge the gap between these two approaches. It requires the visual capabilities of a VIS tool to verify queue formation and passenger flow, but also the logical flexibility of a code-centric framework to implement specific agent-based behaviors.

Based on this comparative analysis, **AnyLogic** was selected as the implementation tool. AnyLogic is distinct in the market as a multimethod simulation environment [Bor20]. It allows for the visual flowchart modeling seen in tools like Arena (via its Process Modeling Library) while simultaneously permitting the injection of custom Java code into any block. This hybrid capability allows for the modeling of the structural flow of the network visually, while complex decision-making logic is handled via scripts, addressing the limitations found in both pure coding and pure flowchart environments.

2.4. Tools for Simulation: AnyLogic

The final component of the methodological background is the selection of a specific software tool to implement the DES model. For this research, AnyLogic was chosen.

2.4.1 Introduction to AnyLogic

AnyLogic is a modern, professional-grade simulation software used for modeling a wide range of complex business, social, and industrial systems. Its defining characteristic is its **multimethod modeling** capability, which uniquely integrates System Dynamics, Discrete Event, and Agent-Based simulation paradigms within a single development environment. This means a modeler is not limited to a single approach and can choose the most efficient method, or a combination of methods, to address a complex problem [Any26d], [Any26e], [Any26f], [Any26g]. This flexibility has established AnyLogic as a leading simulation tool in both commercial and academic research, used by over 40% of Fortune 100 companies [Ind26].

2.4.2 Key Features and Advantages for This Research

AnyLogic was selected for this thesis due to several key features that directly support the research objectives. Its multimethod nature is particularly advantageous; it allows this research to primarily use a process-oriented DES approach while retaining the flexibility to define buses and passengers with agent-based rules and behaviors [Any26h], [Any26i]. For this process-centric modeling, AnyLogic provides the Process Modeling Library, a powerful toolkit designed for the detailed-level modeling of workflows. As illustrated in Figure 2, this environment utilizes a visual, drag-and-drop interface where logical blocks, representing entity generation ('Source'), queuing, and processing delays, are connected to define the structural flow of the system. This library simplifies the creation of models involving queues, resource management, and delays, which are central to this research.

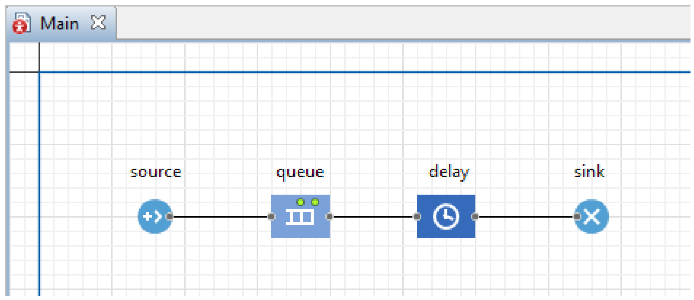

Figure 2: A basic Discrete Event Simulation workflow constructed using AnyLogic's Process Modeling Library, illustrating the connection of Source, Queue, Delay, and Sink blocks

Furthermore, AnyLogic supports **GIS maps integration**, allowing models to use real-world geographic data. While this specific thesis will use a simplified, artificial layout, this feature highlights the software's capability to expand the research toward real-world applications in the future, where bus routes and stop locations could be placed on actual maps [Any26j].

2.4.3 Data Analysis and Reporting Capabilities

Simulation tools like AnyLogic are fundamentally designed to generate data for analysis. The software does not produce final analytical reports itself but enables the model to collect a wide range of output data based on the Key Performance Indicators (KPIs) defined in the model. AnyLogic supports advanced spatial analysis, such as the **density map shown in Figure 3**. This powerful visualization tool highlights areas of high passenger congestion (red zones) versus free flow (blue zones), allowing for an immediate visual assessment of overcrowding risks at the hub before detailed statistical logs are even exported [Any26k], [Any26l].

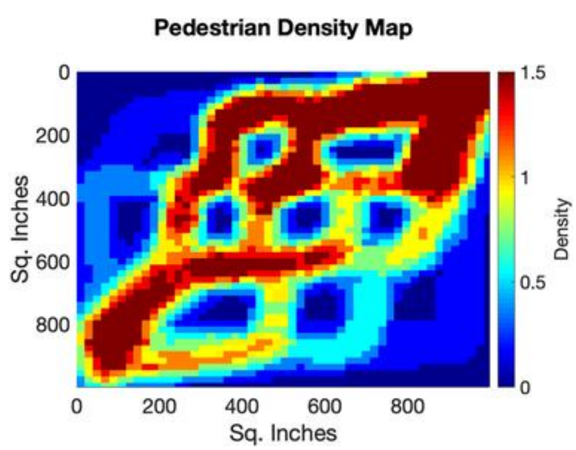

Figure 3: Pedestrian Density Map generated from simulation data. This visualization technique helps identify overcrowding hotspots and bottlenecks within a physical space, which is essential for analyzing hub congestion.

This output data, such as average waiting times or queue lengths, can be stored in collections or datasets within the model and then exported. Once exported, this data can be subjected to comprehensive statistical analysis using external software (such as R, Python, or SPSS), enabling a rigorous quantitative evaluation of the different "what-if" scenarios tested in the experiments. This capability to generate clean, structured data makes the simulation a valuable precursor to data mining and analytics [Any26m], [Any26n].

2.5. Conclusions

This chapter has established the theoretical and methodological foundations necessary for the research. By defining the problem area as a class of Dynamic Service Systems, it was identified that the system's behavior is fundamentally dictated by stochastic demand, resource contention, and the resulting queuing phenomena. These properties render static analytical methods insufficient for this study, thereby justifying the simulation-based approach outlined in the research plan.

To analyze these complex interactions, Discrete Event Simulation (DES) was selected as the most appropriate methodology. The internal mechanics of DES provide the necessary framework for creating the conceptual model and identifying the key system parameters as required by Objective 1. Furthermore, following a comparative review of simulation frameworks ranging from the code-centric SimPy to the visual-centric Arena, AnyLogic was chosen as the implementation tool. Its unique ability to combine process-centric modeling with custom coding and visualization directly supports Objective 3, the need to build a working simulation, by providing the flexibility to model both the structural flow of the network and the dynamic decisions of the agents within it.

The concepts and justifications established in this chapter will therefore serve as the direct blueprint for the practical model development and experimentation phases of this thesis, as detailed in the remaining research objectives.

3. Model Implementation and System Analysis

This chapter details the transition from the theoretical frameworks established in Research Work Part II to a functional, spatially-aware simulation environment. The implementation phase was designed to construct a conceptual simulation model of a multi-route urban transit system, facilitating the quantitative evaluation of dynamic rerouting algorithms. The primary focus was the engineering of a system that not only visualizes transit flow but also functions as a synthetic data generator, producing granular, high-fidelity event logs for informatics-based analysis. The development followed an iterative engineering process, where initial systemic failures were identified through boundary testing and resolved through logic optimization and unit calibration.

3.1. Physical Topology and Spatial Environment Design

To satisfy the requirement of a "multi-stop public transport system" established in Part I, a spatially-aware network topology was developed within the AnyLogic environment. I utilized the Space Markup Palette to construct a 12-node non-linear network, representing a primary urban corridor intersected by a secondary feeder route. This topology should be interpreted as a simplified Vilnius-inspired abstraction rather than a full geographic reconstruction of the Vilnius public transport network. The purpose of the topology is to provide a controlled environment for analysing hub congestion, route interaction, and dynamic allocation logic.

3.1.1. Modeling Spatial Constraints and Latency

The physical environment was engineered using discrete Point Nodes to represent bus stops and intersections. As illustrated in Figure 4, twelve individual nodes were strategically placed to define the geographic boundaries of the system.

1. **The Hub Junction (nodeHUB):** This central node functions as the primary intersection where Route 1 and Route 2 converge. By utilizing a single Point Node for the Hub, the model creates a "Shared Logical Resource" where buses from different origins must compete to serve the same waiting population.

2. **Directed Paths:** The nodes were connected using the Path tool to create road infrastructure. These paths incorporate curves to simulate real-world street topography. In AnyLogic, the physical length of a path dictates the duration a bus agent spends in transit. This addresses the "Spatial Latency" requirement: it ensures that a diverted bus

from Route 2 must physically travel the distance to the Hub, thereby incurring a realistic temporal penalty.

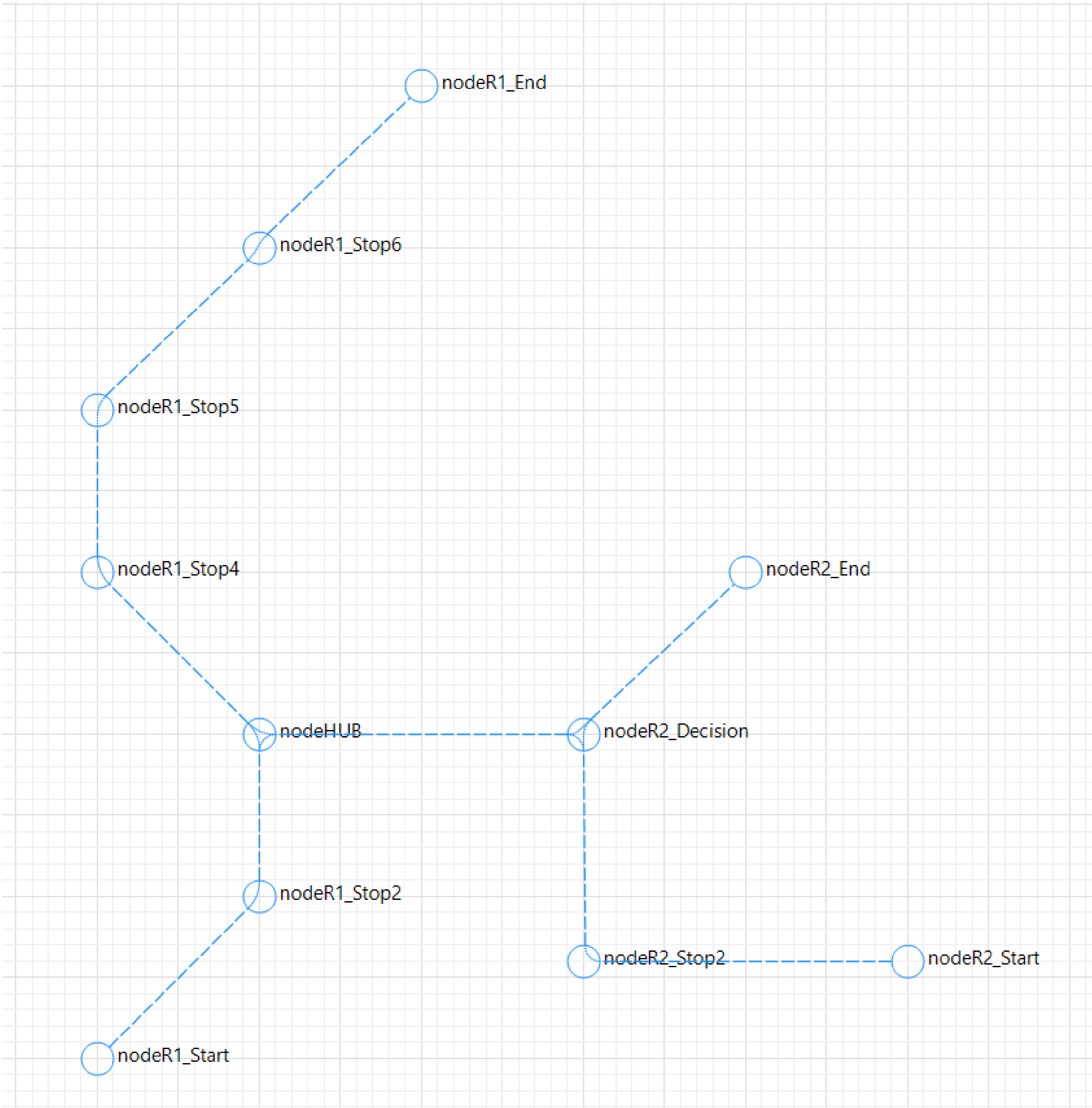

Figure 4 The 2D Spatial Environment showing the 12-node network topology, intersecting routes, and the hub waiting area configuration

3.1.2. Junction Complexity and Point Node Bottleneck

A specific technical decision made during the design of the topology was the implementation of "Agent Egress Logic." As observed in the simulation, bus agents and passengers are programmed to "vanish" upon reaching the final terminal node (nodeR1_End).

For this model, this is an intentional design for Memory Efficiency. By routing agents into a Sink block at the end of their journey, the system clears the simulation memory of completed transactions. This ensures that the data-logging pipeline remains "clean" and that the computer performance does not degrade during long-duration stress tests, such as the 10,000-second experimental window used for final data collection.

3.2. Informatics Logic Architecture and System Components

While the nodes provide the physical space, the intelligence of the system is contained within the Discrete Event Logic Layer. This layer was constructed using the Process Modeling Library to translate urban service dynamics into a programmable flowchart.

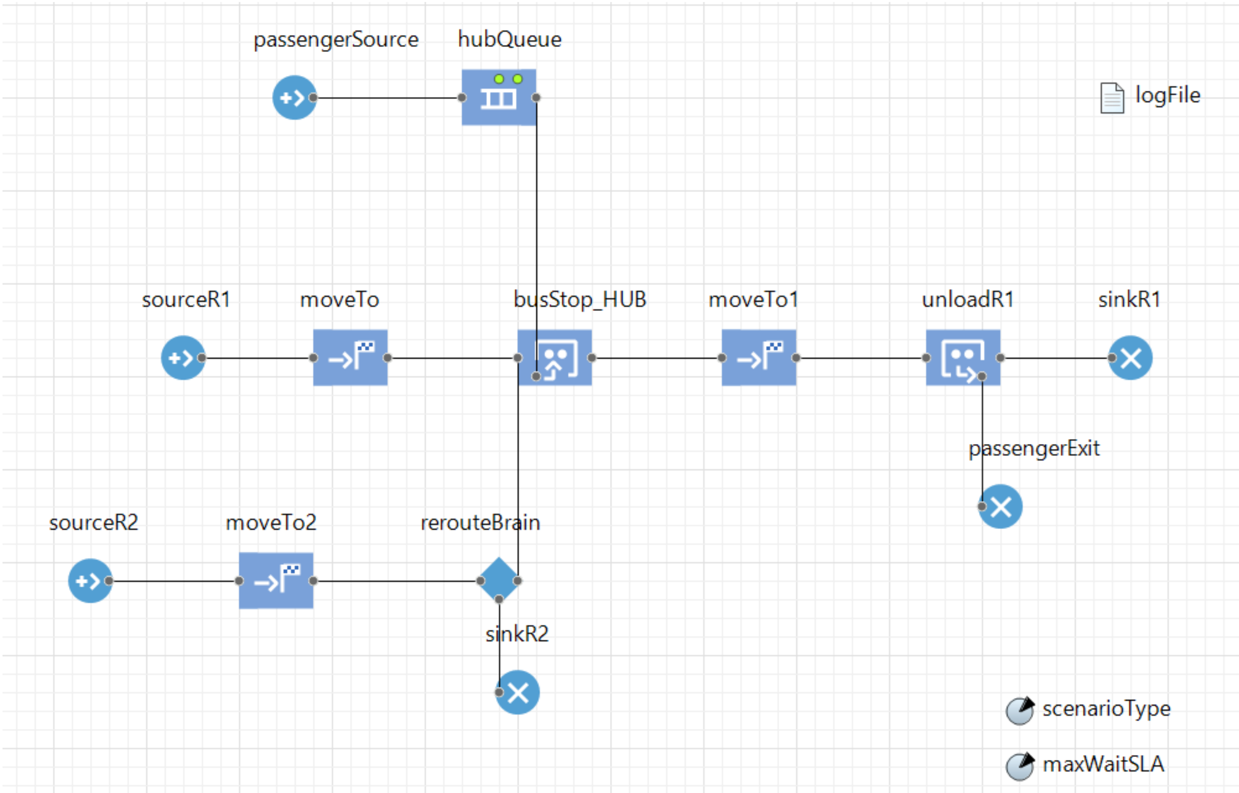

Figure 5 Detailed informatics architecture showing the logic connections between Demand, Supply, and the Dispatcher Brain

Figure 5 presents the detailed informatics architecture of the simulation model. The architecture separates the system into three main logical components: demand, supply, and the dispatcher brain. The demand side represents passenger generation and queue formation at the hub, while the supply side represents bus generation, movement, capacity, and passenger pickup. The dispatcher brain connects these two components by monitoring the state of the system and deciding whether dynamic bus allocation should be activated.

This architecture is important because it shows that the model is not only a visual representation of a transport route, but also an informatics system with input variables, decision logic, and output metrics. Passenger demand creates pressure on the hub queue, bus supply determines how quickly that queue can be served, and the dispatcher brain evaluates whether an adaptive service response is needed under the selected scenario parameters.

3.2.1. Stochastic Demand Generation (passengerSource)

The generation of passenger demand was modeled as a Non-Stationary Poisson Process. Within the properties of the passengerSource block, the arrival rate was set to 120 per hour.

To ensure the "Data Engineering" objective of traceability was met, a Java-based initialization was configured in the Actions tab of this block. As shown in Figure 8, the code *agent.id = (int)self.out.count(); and agent.arrivalTime = time();* was utilized to assign a unique identifier and an arrival timestamp to every passenger. This transforms a simple visual entity into a trackable data record for the final CSV dataset.

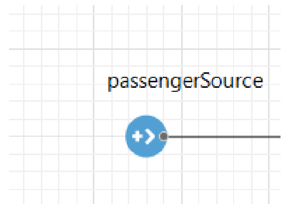

Figure 6 passengerSource block used for generating passenger agents

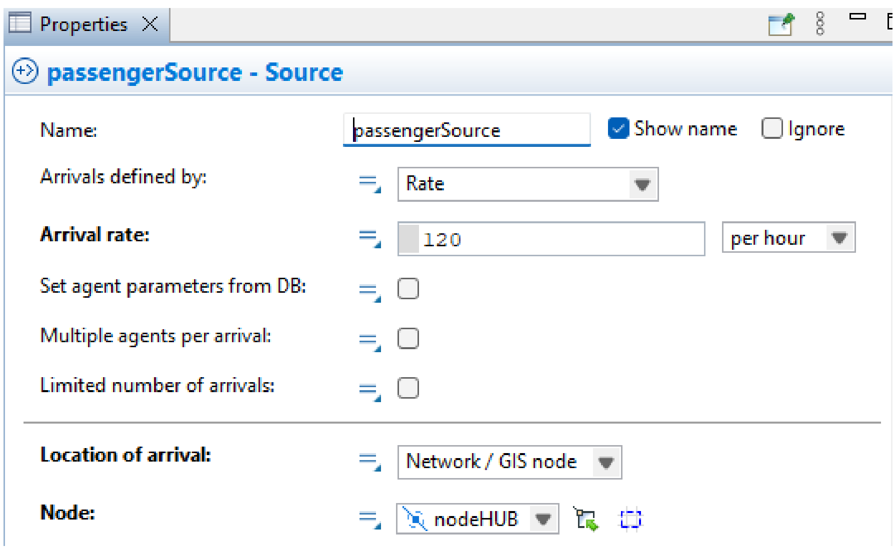

Figure 7 passengerSource properties defining the passenger arrival rate and hub location

Figures 6 and 7 show the implementation and configuration of the passenger demand component in the AnyLogic model. Figure 6 presents the `passengerSource` block as it appears in the process flow, while Figure 7 shows the main properties used to configure this source. In this model, passenger arrivals are defined by a rate of 120 passengers per hour, and generated passenger agents are placed at `nodeHUB`. This means that passenger demand is introduced into the model through a rate-based arrival process rather than by manually creating a fixed sequence of passengers.

This configuration is important because the passenger source provides the demand side of the simulation. Once passengers are generated, they enter the hub queue and wait until bus capacity becomes available. Therefore, the queue length, passenger time samples, and later probability distribution results are all influenced by the interaction between this passenger arrival process, the bus arrival process, bus capacity, disruption behaviour, and the dynamic allocation policy.

Figure 8 passengerSource Agent Type

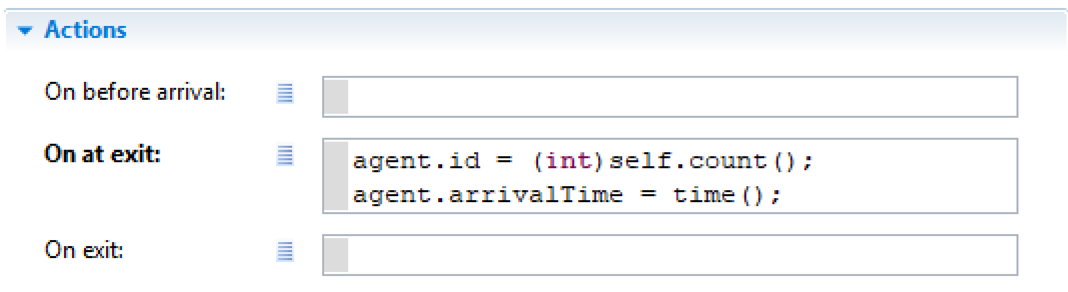

Figure 9 passengerSource technical configuration of the stochastic demand source including Java-based agent initialization

3.2.2. System State Monitoring and Buffer Dynamics (hubQueue)

The real-time state of the system is monitored via the **hubQueue** block. In Discrete Event Simulation, this block functions as a "System Sensor." As illustrated in Figure 10, the block displays a real-time numerical counter (e.g., "68") representing the In-System Inventory, passengers who have arrived but have not yet been picked up by a bus.

Initially, this buffer was set to a finite capacity, but following a "Buffer Overflow" failure during early testing, it was re-engineered to **Maximum Capacity.** This allows the simulation to record the full extent of the "Queue Divergence" in Scenario A, providing the raw data needed to prove the inefficiency of fixed schedules.

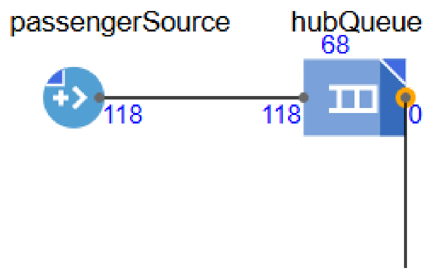

Figure 10 Real-time monitoring of the Hub bottleneck showing the accumulation of unserviced passenger agents.

3.2.3. Heuristic Decision Logic (rerouteBrain)

The "Brain" of the system is represented by the **SelectOutput** diamond. This component acts as a Cyber-Physical Controller. Every time a bus on the quiet route reaches the decision node, the system performs a state-check of the Hub. Based on **Little's Law**, the threshold was set to 30 passengers. The Java condition **hubQueue.size() > 30** was implemented to trigger the " dynamic allocation" logic. This demonstrates a shift from "Planned Transit" to "Data-Responsive Infrastructure," where the physical route of a vehicle is altered based on real-time informatics signals.

3.2.4. The Primary Operational Pipeline (Route 1)

Route 1 represents the main urban artery where the majority of service transactions occur.

1. **sourceR1:** This block generates the primary bus fleet at a fixed 30-minute interval. It initializes the "Resource Containers" that move through the city.
2. **moveTo and moveTo1**: These blocks are the spatial bridges. They take the logical bus from the flowchart and force it to physically traverse the paths drawn on the 12-node map. This ensures that the bus spends a realistic amount of time traveling between the Start, the Hub, and the Terminal.
3. **busStop_HUB** (The Interface): This is a Pickup block and is the most critical junction in the model. It has three specific connections:
    - **Input (Left):** Receives scheduled buses from Route 1.

- **Wait Port (Bottom-Left Green Dot):** This is connected to the **hubQueue.** It acts as the "Waiting Room." The bus will only stay here for a millisecond to "grab" passengers before moving to the next block.
- **Reroute Input (Bottom):** This is connected to the rerouteBrain, allowing dynamically allocated buses from Route 2 to enter the Hub logic.

3.2.5. The Adaptive Resource Pipeline (Route 2)

Route 2 functions as the "Optimization Reservoir." Its primary purpose is to provide resources to the Hub when the Service Level Agreement (SLA) is breached.

1. **sourceR2 and moveTo2:** These generate and move buses along the quiet feeder route until they reach the **nodeR2_Decision.**
2. **rerouteBrain (The Heuristic Diamond):** This is the "Brain" of the system as explained above in section 3.2.3.
   - **If the condition is True (Queue > 30):** The bus is dynamically allocated to support the**_HUB.**
   - **If the condition is False:** The bus is sent to **sinkR2,** representing the completion of its original quiet route.

3.2.6. Transaction Completion and Data Egress

To analyze and process the data for future works , the "Passenger Lifecycle" must be completed and recorded.

1. **unloadR1 (The Data Scanner):** This is a Dropoff block. It is situated at the terminal node. Its purpose is to "unpack" the passengers from the bus container. This is a technical necessity; without this block, the passengers would remain "invisible" inside the bus and their individual time-in-system samples could not be logged.
2. **passengerExit and sinkR1**: Once separated, the empty bus moves to sinkR1, and the individual passengers move to passengerExit. I placed the Java-based Event Listener inside the unloadR1 block so that the moment a human "steps off" the bus, their wait-time is calculated and written to the logFile CSV object.

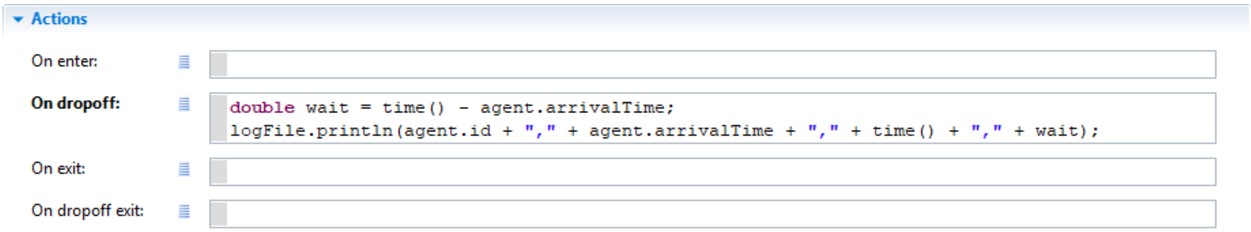

Figure 11 unloadR1 Java-based Event Listener

3.2.7. Control Parameters and Global Variables

I implemented two Control Parameters that allow for real-time model reconfiguration:

1. **scenarioType:** A String parameter that toggles the system between "***ScenarioA***" (Fixed) and "***ScenarioB***" (Dynamic). This allows for the automated partitioning of results into separate datasets.
2. **maxWaitSLA:** A Double parameter set to 15.0 minutes. This provides the mathematical basis for the rerouteBrain threshold, linking the system's physical behavior directly to a performance-driven urban policy.

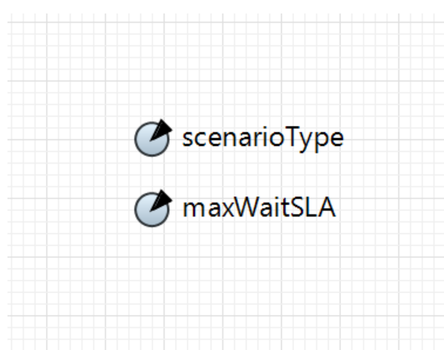

Figure 12 Global Variables

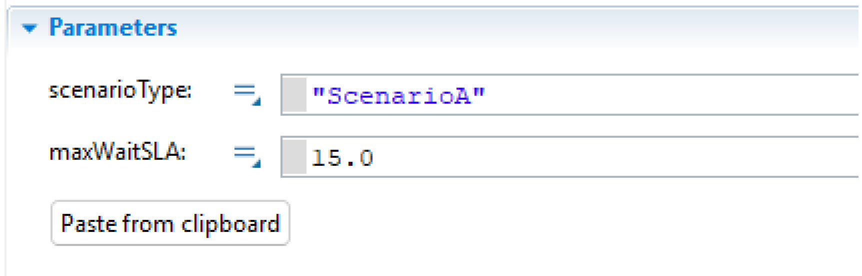

Figure 13 Global Variables Properties

3.3. Algorithmic Logic and Little's Law Optimization

A central objective of this research was to move beyond a simple count-based trigger to a mathematically justified dispatching algorithm. To achieve this, the rerouting logic in Scenario B was engineered using Little's Law **($L = \lambda W$).** In informatics and queuing theory, this law provides the fundamental relationship between the average number of items in a system (L) their arrival rate *(λ)*, and their average wait time (W).

To satisfy the Service Level Agreement (SLA) target of an average wait time of **15 minutes** (W=15), and given the programmed stochastic arrival rate of 2 passengers per minute (λ=2), the critical threshold for system intervention was mathematically calculated as L=30. As shown in the rerouteBrain properties in Figure 11, this value was used as the trigger for the dynamic "dynamic allocation" logic. By utilizing a threshold derived from Little's Law, the system transitions from a reactive state to a performance-driven state, intervening the moment the queue length suggests that the SLA is in jeopardy of being breached.

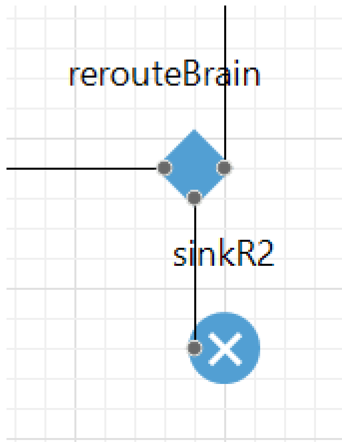

Figure 14 rerouteBrain

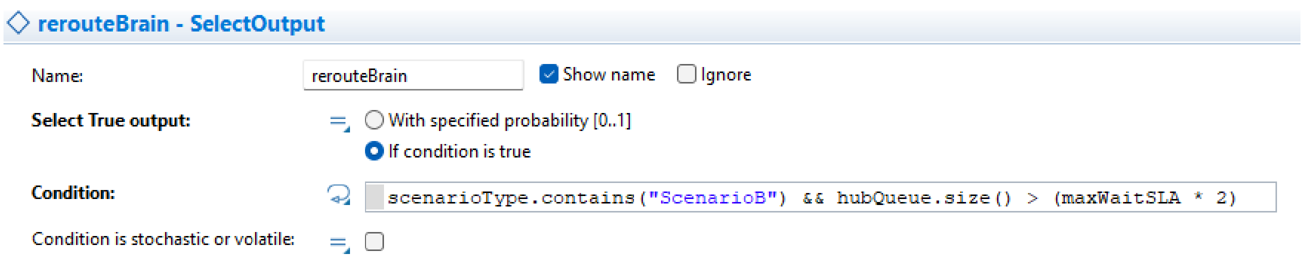

Figure 15 Implementation of the dynamic threshold logic derived from Little's Law (L=λW)

3.4. Technical Challenges and Adaptive

The implementation of the simulation model involved an iterative debugging process, which provided empirical evidence of how complex service systems fail under pressure. Two specific technical hurdles were identified and resolved to ensure the model's scientific validity.

3.4.1. Heuristic Decision Logic (rerouteBrain)

During the initial diagnostic phase, a "Unit Divergence" issue was observed where bus interarrival times were misinterpreted by the simulation engine. Due to the AnyLogic Java-expression environment defaulting to the model's base time unit (Hours), a requested 30-minute interval was initially executed as a 30-hour delay. This resulted in wait times exceeding 1500 units in early CSV logs. I resolved this by explicitly calibrating the interarrival time to **30 minutes** in the Source properties, ensuring the simulation window accurately mirrors a standard metropolitan rush-hour cycle.

3.4.2. Resolution of Buffer Overflow Exception

A secondary failure occurred during the stress-testing of Scenario A, where the system encountered a **Buffer Overflow** error as the hub queue exceeded its 100-agent limit. This crash provided a "Failure-Proof" validation of the research problem: it confirmed that under a fixed-schedule regime, the physical capacity of a transit hub is mathematically insufficient to contain the unserviced demand. To allow for the measurement of the true scale of delay divergence, I re-engineered the **hubQueue** to **Maximum Capacity**, enabling the capture of wait times exceeding 2500 seconds without system termination.

3.5. Comparative Scenario Analysis and Quantitative Results

With the system architecture finalized and technical hurdles resolved, the simulation was subjected to a rigorous comparative analysis. To ensure scientific consistency and a ceteris paribus (all other things being equal) comparison, both operational scenarios were executed for a fixed observation window of exactly **10,000 seconds** (approximately 2.7 hours). This timeframe represents a typical morning rush-hour peak in a metropolitan center like Vilnius.

3.5.1. Visual Evidence of System Divergence

1. **In Scenario A (The Baseline):** The system operated under a fixed-interval regime with no cross-route intervention. As observed in Figure 16, the Hub reached a state of "Systemic Congestion." By the end of the run, 71 passenger agents were left unserviced in the queue. The blue numbers above the blocks indicate that while the bus supply was constant, it was mathematically unable to keep pace with the stochastic arrival of 321 passengers.
2. **In Scenario B (The Optimized Model):** The SLA-based rerouting logic was activated. As shown in Figure 17, the "Reroute Brain" successfully identified the bottleneck. By dynamically allocating one bus from the quiet Route 2 (indicated by the count of '1' on the top exit of the rerouteBrain diamond), the system injected extra capacity into the Hub. Consequently, the final queue was reduced to 35 passengers, representing a significantly more stable system state.

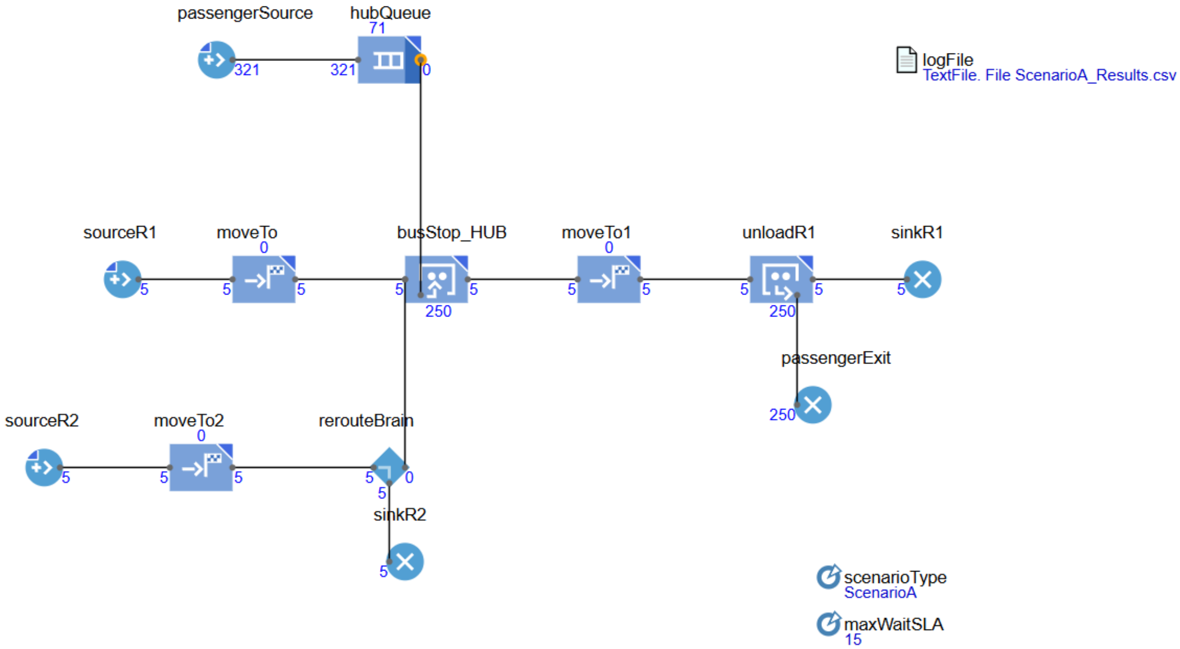

Figure 16 Scenario A simulation run at 10,000 seconds

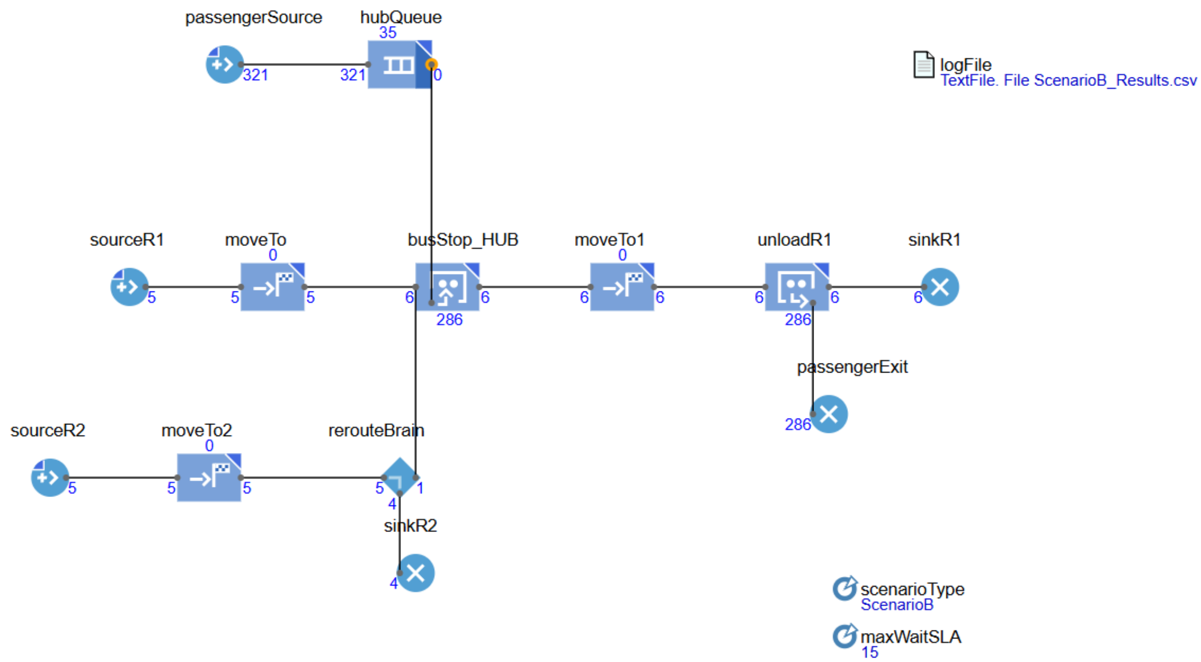

Figure 17 Scenario B simulation run at 10,000 seconds

3.5.2. Quantitative Metric Comparison

The raw data captured by the event-logging pipeline was processed to calculate the primary Key Performance Indicators (KPIs) identified in Part I of this research. The following table summarizes the divergence in performance between the two dispatching policies.

Table 1 Comparative Analysis of Simulation Results (T = 10,000s)

Performance Metric	**Scenario A (Fixed)**	**Scenario B (Dynamic)**	**Improvement (%)**
Total Passenger Arrivals	321	321	0% (Control)
Passengers Successfully Processed	250	286	**+14.4% Throughput**
Final Residual Queue	71	35	**-50.7% Reduction**
Maximum Observed Wait Time	2721s	1802s	**-33.8% Delay Reduction**

3.5.3. Analysis of Informatics-Led Optimization

The most significant finding for this implementation phase is the relationship between Information and Efficiency. The data indicates that **Scenario B** achieved a **14.4%** increase in throughput and cut the standing queue by more than half **(50.7%)** without the need for additional physical vehicles.

By utilizing the **hubQueue.size()** as a real-time informatics signal, the system reallocated an existing resource to the point of highest demand. This validates the core hypothesis: that dynamic, data-driven dispatching logic can mitigate urban transit delays more effectively than traditional, rigid schedules. The reduction in the Maximum Wait Time from 2721s to 1802s further proves that the algorithm successfully "caps" extreme delays, preventing the total service collapse seen in the baseline.

3.6. Conclusion and Future Works

The implementation phase documented in this chapter has successfully established a functional 2D simulation framework and a high-resolution data engineering pipeline. I have demonstrated that a dynamic, SLA-triggered rerouting algorithm provides a measurable improvement in urban flow.

While the current chapter focused on the implementation of the core supply-side logic, it provides the technical foundation for the final semester. In chapter 4, I plan to extending this research by:

1. **Impact Analysis of Bypassed Routes:** Measuring the exact delay penalty incurred by Route 2 passengers when a bus is diverted.
2. **GIS-Based Scalability:** Porting this logic onto a high-fidelity map of the Vilnius transit network to analyze the impact of traffic light synchronization on rerouting efficacy.

4. Extended Probabilistic Analysis of Dynamic Bus Allocation

The previous chapter presented the implementation of the initial AnyLogic simulation model and demonstrated how a fixed public transport service configuration can be compared with a dynamically adjusted configuration. The implemented model included stochastic passenger generation, a hub queue, scheduled bus arrivals, and a decision component that activates when hub congestion exceeds a defined threshold. This provided the technical foundation for evaluating how adaptive service logic can affect passenger flow and delay in a simplified public transport service system.

In this final stage of the research, the analysis is extended from a basic comparison of average and maximum delay values to a probabilistic evaluation of passenger time outcomes. This extension is necessary because dynamic service systems do not behave in a fully deterministic way. Passenger arrivals fluctuate around an expected rate, bus service can be delayed, and occasional operational disruptions can significantly change the distribution of passenger experiences. For this reason, evaluating only a single average value or a single maximum value gives an incomplete view of the system's behaviour.

The final experiment therefore focuses on the empirical distribution of passenger time samples collected from the simulation. These samples are used to construct probability distribution plots, cumulative probability distributions, and threshold exceedance probabilities. This makes it possible to evaluate not only whether the average passenger time changes, but also how likely passengers are to experience longer or unacceptable delays under different operating conditions.

In this chapter, the earlier dynamic rerouting logic is reframed more generally as dynamic bus allocation. This terminology is used because the purpose of the model is not to reproduce a full real-time rerouting system for the Vilnius public transport network, but to analyse a simplified dynamic service response in which additional bus capacity can be assigned to a congested hub when predefined operational conditions are met. The extended analysis compares normal operation, disrupted operation, and dynamically adjusted operation under fairness and cold-weather constraints.

4.1. Motivation for Extending the Initial Model

The initial implementation in Chapter 3 showed that a fixed service configuration can lead to queue accumulation at the hub, while a dynamically adjusted configuration can reduce the residual queue and improve throughput. This was an important first step because it verified that the AnyLogic model could represent passenger arrivals, queue formation, bus pickup operations, and adaptive decision logic. However, the previous analysis mainly relied on aggregate indicators such as final queue length, throughput, and maximum observed time.

Although these indicators are useful, they do not fully describe the stochastic nature of the modeled service system. In public transport, the experience of passengers is not represented only by the average. A scenario with an acceptable average passenger time may still contain a significant number of passengers who experience very long delays. Similarly, a maximum value shows the worst observed case, but it does not show how common long delays are. For this reason, a distribution-based analysis is needed to understand the probability of different passenger time outcomes.

The extended model therefore introduces a more detailed experimental layer. First, a general disruption mechanism is added to represent the combined effect of traffic congestion, weather-related slowdown, or minor operational incidents. Second, individual passenger time samples are collected during simulation runs and exported for statistical analysis. Third, multiple scenarios are compared using empirical probability distributions and cumulative probability distributions. This allows the analysis to answer questions such as how the probability of exceeding a 30-minute or 60-minute passenger time changes when the system is disrupted, and whether dynamic bus allocation reduces the likelihood of these long-delay outcomes.

This extension also makes the analysis more suitable for evaluating trade-offs. A dynamic allocation policy may improve the congested hub, but it should not be interpreted as automatically beneficial in every situation. If assigning additional capacity to the hub creates unacceptable waiting impact elsewhere, the policy may need to be restricted. For this reason, the extended model includes fairness and cold-weather constraints through the parameters **affectedRouteHeadway**, **maxWaitSLA**, and **temperatureC**. These parameters allow the model to distinguish between cases where dynamic allocation is acceptable and cases where it should be rejected to protect passengers affected by the service adjustment.

4.2. Revised Experimental Objective

The objective of the extended experiment is to evaluate how probabilistic disruptions and dynamic bus allocation policies affect passenger time outcomes in the simulated public transport service system. While the previous implementation showed how adaptive logic can reduce queue accumulation at the hub, the extended analysis focuses on a broader and more informative question: how the full distribution of passenger time samples changes under different operating conditions.

The extended experiment is therefore guided by the following research question:

1. How do probabilistic disruptions affect passenger time distributions in a simplified public transport hub model, and to what extent can dynamic bus allocation reduce the probability of excessive passenger times under fairness and cold-weather constraints?

This question is more suitable for the final analysis because it does not only compare average values. Instead, it considers the probability that passengers experience longer delays. In this context, the important output is not only whether the mean passenger time decreases, but also whether the probability of long passenger times, such as values above 30, 45, or 60 minutes, is reduced.

The revised objective can be divided into four experimental goals:

1. to introduce a disruption mechanism that represents adverse operating conditions such as traffic congestion, weather-related slowdown, or minor incidents;
2. to collect individual passenger time samples from simulation runs;
3. to compare the empirical probability distributions and cumulative probability distributions of these samples across different scenarios;
4. to evaluate whether dynamic bus allocation reduces the probability of excessive passenger time compared with the disrupted baseline system.

The analysis is focused on a simplified but parameterized model. The aim is not to reproduce the complete Vilnius public transport network, but to evaluate the behaviour of a dynamic service system under controlled conditions. This is consistent with the purpose of discrete-event simulation, where simplified models are used to study how system parameters, stochastic events, and decision rules influence performance measures.

4.3. Extended Model Parameters and Dynamic Modes

To support the extended analysis, several additional parameters and variables were introduced into the AnyLogic model. These additions make it possible to distinguish between fixed scenario settings, dynamic operating modes, and output metrics collected during the simulation.

The initial model already included passenger generation, bus generation, hub queueing, bus pickup logic, and a congestion-based decision component. In the extended model, the same structure is preserved, but additional parameters are introduced to describe disruption probability, bus delay behaviour, dynamic allocation policy, and passenger time-sample collection. Table 2 presents the main parameters used in the extended experiment.

Table 2 Main parameters used in the extended probabilistic experiment

Parameter	Value used in experiments	Meaning
passengerArrivalRatePerHour	120	Average passenger arrival intensity at the hub.
baseBusHeadwayMinutes	30	Scheduled base interval between bus arrivals.
normalDelayMinutes	2	Additional delay applied in normal operating mode.
disruptionDelayMinutes	20	Additional delay applied in disruption mode.
disruptionProbability	0 or 1	Probability that a generated bus is affected by disruption in the tested scenario.
Q_THRESHOLD	30	Hub queue threshold that activates dynamic allocation logic.
maxWaitSLA	15	Maximum acceptable affected-route waiting threshold in normal conditions.
temperatureC	5 or -15	Temperature value used for weather-adaptive scenarios.
affectedRouteHeadway	10 or 15	Estimated waiting impact on the affected route if bus capacity is dynamically allocated.

policyType	0, 1, 2, 3	Selects the operating policy used in the simulation run.

As shown in Table 2, the model separates passenger demand, bus service frequency, disruption behaviour, and allocation policy into explicit parameters. This is important because it allows alternative operating conditions to be tested without changing the structure of the model itself.

The **policyType** parameter defines the dynamic service policy used in a simulation run. Its values are shown in Table 3.

Table 3 Operating policies defined by the policyType parameter

policyType	Policy name	Description
0	Baseline operation	No dynamic bus allocation is applied.
1	Greedy dynamic allocation	Additional service capacity is allocated when the hub queue exceeds the congestion threshold.
2	Fairness-protected allocation	Allocation is allowed only if the estimated affected-route wait does not exceed maxWaitSLA.
3	Weather-adaptive allocation	Allocation follows the fairness rule, but the acceptable affected-route wait becomes stricter under cold-weather conditions.

In addition to fixed parameters, the extended model includes a dynamic disruption mode. This mode represents the combined effect of adverse conditions such as traffic congestion, weather slowdown, and minor incidents. Instead of modelling each cause separately, the current version uses one aggregate disruption mechanism. This keeps the model manageable while still demonstrating how probabilistic disruption can be represented in a discrete-event simulation.

At each bus generation event, the model checks whether disruption occurs according to the **disruptionProbability** parameter. If no disruption occurs, the bus receives the normal additional delay. If disruption occurs, the bus receives the larger disruption delay. The effective bus interarrival time is therefore calculated as:

- effective bus interarrival time = **baseBusHeadwayMinutes** + sampled delay

where the sampled delay is either normalDelayMinutes or disruptionDelayMinutes, depending on the sampled operating mode.

This mechanism creates two possible operating modes during simulation:

- **normal mode**, where buses experience only a small additional delay;
- **disruption mode**, where buses experience larger delays due to adverse operating conditions.

This distinction is important because public transport systems do not always operate under the same conditions. A model that assumes only fixed bus arrivals may underestimate the probability of long passenger waits. By adding a disruption mode, the simulation can be used to observe how passenger time distributions change when bus service becomes less reliable.

The cold-weather logic is represented separately from the disruption mode. In the weather-adaptive policy, low temperature reduces the acceptable waiting threshold for affected passengers. In the implemented experiment, when **temperatureC** is equal to **-15**, the allowed affected-route wait is reduced to 10 minutes. Therefore, a dynamic allocation that would be acceptable in mild weather may be rejected in severe cold conditions.

Together, these parameters and dynamic modes allow the extended model to compare not only whether dynamic allocation reduces the average passenger time, but also how it changes the probability distribution of passenger time outcomes under normal, disrupted, and weather-constrained conditions.

4.4. Scenario Design

The extended experiment was organized into a set of controlled scenarios. Each scenario uses the same underlying AnyLogic model structure, but changes selected input parameters such as disruption probability, policy type, temperature, and affected route headway.

This allows the effect of each operating condition to be compared without changing the general model architecture.

The scenarios were designed to answer three main analytical questions:

1. **How does the system behave under normal operating conditions?**
2. **How much does performance degrade when disruption affects bus arrivals?**
3. **Can dynamic bus allocation reduce the probability of long passenger times under disruption, and how do fairness or cold-weather constraints affect this response?**

The scenario set is shown in Table 4.

Table 4 Scenario configuration used in the extended experiment

Scenario	policyType	Operating condition	disruption Probability	temperatureC	affectedRoute Headway	Purpose
S0 Normal baseline	0	Normal baseline operation	0	5	15	Reference case without disruption or dynamic allocation.
S1 Disrupted baseline	0	Disrupted operation without response	1	5	15	Measures the effect of disruption when no dynamic response is applied.
S2 Disrupted greedy allocation	1	Disrupted operation with gr eedy allocation	1	5	15	Evaluates the effect of dynamic allocation when hub congestion is prioritized.
S3 Disrupted fairness allocation	2	Disrupted operation with fairness-protected allocation	1	5	15	Evaluates dynamic allocation when affect ed-route waiting impact is checked.
S4 Cold-weather adaptive rejected	3	Disrupted cold-weather operation	1	-15	15	Tests the case where cold-weather constraints rej ect allocation.
S5 Cold-weather adaptive allowed	3	Disrupted cold-weather operation	1	-15	10	Tests the case where allocation is allowed under the cold-weather threshold.

Scenario S0 is used as the normal reference case. It represents standard operation without disruption and without dynamic allocation. Scenario S1 introduces sustained disruption but keeps **policyType = 0**, meaning that the system does not dynamically respond. This scenario is used to measure how strongly disruptions affect passenger time outcomes when the system remains static.

Scenarios S2 and S3 introduce dynamic bus allocation under the same disruption setting. In S2, the greedy allocation policy is used. This policy prioritizes the congested hub and accepts allocation when the hub queue exceeds the threshold. In S3, the fairness-protected policy is used. Since **affectedRouteHeadway = 15** and **maxWaitSLA = 15**, the fairness condition is satisfied, so allocation is allowed.

Scenarios S4 and S5 test the weather-adaptive policy under cold-weather conditions. In both cases, **temperatureC = -15**, which reduces the acceptable affected-route wait to 10 minutes. In S4, **affectedRouteHeadway = 15**, so allocation is rejected. In S5, **affectedRouteHeadway = 10**, so allocation is allowed. These two scenarios are important because they show that the weather-adaptive policy is conditional: it does not automatically reject all allocation in cold weather, but only rejects it when the estimated affected-route waiting impact exceeds the cold-weather threshold.

The disruption probability is set to 1 in the disrupted scenarios to create a controlled stress-test condition. This does not represent ordinary everyday service operation. Instead, it is used to clearly observe how the system behaves when adverse operating conditions persist and to compare how different dynamic allocation policies respond to this situation.

4.5. Data Collection and Output Metrics

The extended experiment required collecting more detailed output data than the initial model. Instead of relying only on aggregate indicators, the model records individual passenger time samples during each simulation run. These samples provide the basis for the probability distribution and cumulative probability analysis presented later in this chapter.

Each passenger agent contains an **arrivalTime** variable, which is assigned when the passenger is generated by the passenger source. When the passenger exits the modeled process, the elapsed time is calculated as:

passenger time sample = current simulation time - passenger arrival time

The calculated value is converted into minutes and stored in the **waitingTimes** collection. This collection is then exported to CSV format and analysed using Python. The exported samples make it possible to create empirical distribution plots, cumulative probability plots, and probability threshold tables.

In the current implementation, the recorded value represents a passenger time sample or time-in-system measure within the simplified model. It includes the time between passenger generation and passenger exit from the modeled process. Therefore, it should be interpreted as an observed passenger time sample rather than a fully separated pure queue waiting time. A more detailed model could record several separate components, such as queue waiting time, boarding time, and in-vehicle travel time. This is identified as a possible future extension.

The main output metrics used in the analysis are shown in Table 5.

Table 5 Output metrics used in the extended analysis

Metric	Meaning
Passenger sample count	Number of passenger time samples collected in the scenario.
Mean passenger time	Average value of collected passenger time samples.
Median passenger time	Central value of passenger time samples.
Maximum passenger time	Largest observed passenger time sample.
75th percentile	Value below which 75% of passenger samples fall.
90th percentile	Value below which 90% of passenger samples fall.

Metric	Meaning
95th percentile	Value below which 95% of passenger samples fall.
P(time > 15)	Probability that a passenger time sample exceeds 15 minutes.
P(time > 30)	Probability that a passenger time sample exceeds 30 minutes.
P(time > 45)	Probability that a passenger time sample exceeds 45 minutes.
P(time > 60)	Probability that a passenger time sample exceeds 60 minutes.
Accepted allocations	Number of dynamic allocation actions accepted by the policy.
Rejected allocations	Number of candidate allocation actions rejected by fairness or weather constraints.
Fairness violations	Number of cases where affected-route waiting impact exceeded the acceptable threshold.
Disruption count	Number of disruption events sampled during the simulation run.

The probability threshold metrics are especially important because they provide a direct way to interpret the empirical distributions. For example, **P(time > 60)** shows the probability that a passenger time sample exceeds 60 minutes in a given scenario. This is more informative than the maximum value alone, because it shows whether long passenger times are rare or frequent.

The combination of summary statistics, percentiles, threshold probabilities, and distribution plots allows the extended analysis to evaluate both average system performance and tail-risk behaviour. This is important because a public transport service policy should not only reduce average delays, but should also reduce the likelihood of extreme passenger delays under disrupted operating conditions.

4.6. Summary Statistics of Passenger Time Samples

After running the scenarios defined in Section 4.4, the collected passenger time samples were exported from AnyLogic and analysed using Python. The purpose of the summary statistics is to provide an initial numerical comparison before examining the full probability distributions. While the distribution plots in the following sections provide a more complete view, the summary statistics help identify the main changes in average behaviour and upper-tail passenger time values.

Table 6 presents the main descriptive statistics for the six scenarios.

Table 6 Summary statistics of passenger time samples by scenario

Scenario	Sample count	Mean	Median	75th percentile	90th percentile	95th percentile	Minimum	Maximum
S0 Normal baseline	250	31.39	31.50	40.81	47.38	51.05	2.61	58.87
S1 Disrupted baseline	150	60.70	61.81	80.76	90.95	94.69	20.61	97.41
S2 Disrupted greedy allocation	291	25.43	24.17	38.17	45.67	48.14	0.13	52.99
S3 Disrupted fairness allocation	291	25.43	24.17	38.17	45.67	48.14	0.13	52.99
S4 Cold-weather adaptive rejected	150	60.70	61.81	80.76	90.95	94.69	20.61	97.41
S5 Cold-weather adaptive allowed	291	25.43	24.17	38.17	45.67	48.14	0.13	52.99

As shown in Table 6, the disrupted baseline scenario S1 produces substantially larger passenger time values than the normal baseline scenario S0. The mean passenger time increases from 31.39 minutes in S0 to 60.70 minutes in S1. The 95th percentile also increases from 51.05 minutes to 94.69 minutes. This indicates that the disruption mechanism affects not only the average passenger time, but also the upper part of the distribution.

The dynamic allocation scenarios S2 and S3 produce considerably lower passenger time values than the disrupted baseline. In both scenarios, the mean passenger time is reduced to 25.43 minutes, while the 95th percentile decreases to 48.14 minutes. This suggests that dynamic allocation can reduce the negative effect of sustained disruption in the tested configuration.

The results for S2 and S3 are identical because the fairness constraint is satisfied in S3. In this scenario, **affectedRouteHeadway = 15** and **maxWaitSLA = 15**, so the fairness-protected policy allows dynamic allocation. Therefore, the operational behaviour becomes the same as in the greedy allocation scenario S2. This is not a contradiction, but an expected result of the policy logic.

Scenario S4 produces the same statistical result as S1. This occurs because S4 uses the weather-adaptive policy under **temperatureC = -15**, where the acceptable affected-route waiting threshold is reduced to 10 minutes. Since **affectedRouteHeadway = 15**, the policy rejects allocation. As a result, the system behaves like the disrupted baseline. Scenario S5 confirms the conditional nature of the weather-adaptive policy. When **affectedRouteHeadway** is reduced to 10 minutes, the cold-weather threshold is satisfied and allocation is allowed, producing the same improved outcome as S2 and S3.

Overall, the summary statistics show three main system behaviours: normal operation, disrupted operation without an effective response, and disrupted operation with accepted dynamic allocation. However, summary values alone do not fully explain the shape of the passenger time distribution. For this reason, the following sections analyse the same results using probability distribution plots and cumulative probability distributions.

4.7. Probability Distribution Analysis

The first graphical analysis uses empirical probability distribution plots of the passenger time samples. These plots show how frequently different passenger time values occur in each scenario. This is useful because two scenarios may have similar averages but different distribution shapes. A scenario with a narrow distribution concentrated around lower values indicates more stable passenger experience, while a scenario with a wider distribution shifted toward higher values indicates a greater probability of long passenger times.

Figure 18 presents the histogram comparison of passenger time samples across the tested scenarios.

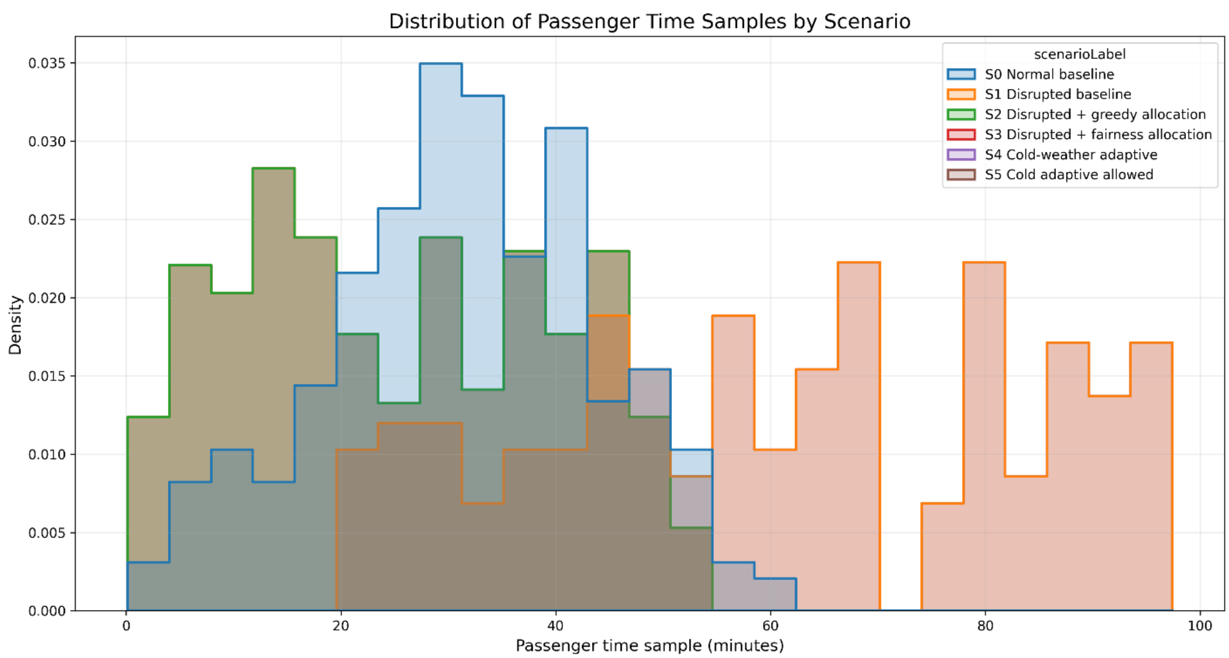

Figure 18 Distribution of passenger time samples by scenario

Figure 18 shows that the disrupted baseline scenarios produce a clear shift toward higher passenger time values. In particular, S1 and S4 are concentrated in higher time ranges than the normal baseline S0. This means that sustained disruption increases the frequency of long passenger time outcomes. The dynamic allocation scenarios S2, S3, and S5 are shifted toward lower time values, showing that accepted allocation reduces the concentration of long passenger times.

The overlap between some scenarios is also meaningful. Scenarios S2, S3, and S5 have the same distribution because the allocation condition is satisfied in each case and the system applies the same effective response. Similarly, S1 and S4 overlap because allocation is rejected in S4, causing it to behave like the disrupted baseline. These overlapping curves confirm that the policy logic is functioning consistently: when allocation is allowed, the improved distribution appears; when allocation is rejected, the disrupted baseline distribution remains

To further examine the shape of the distributions, an estimated probability density plot was also generated. Figure 19 shows a smoothed estimate of the probability density of passenger time samples.

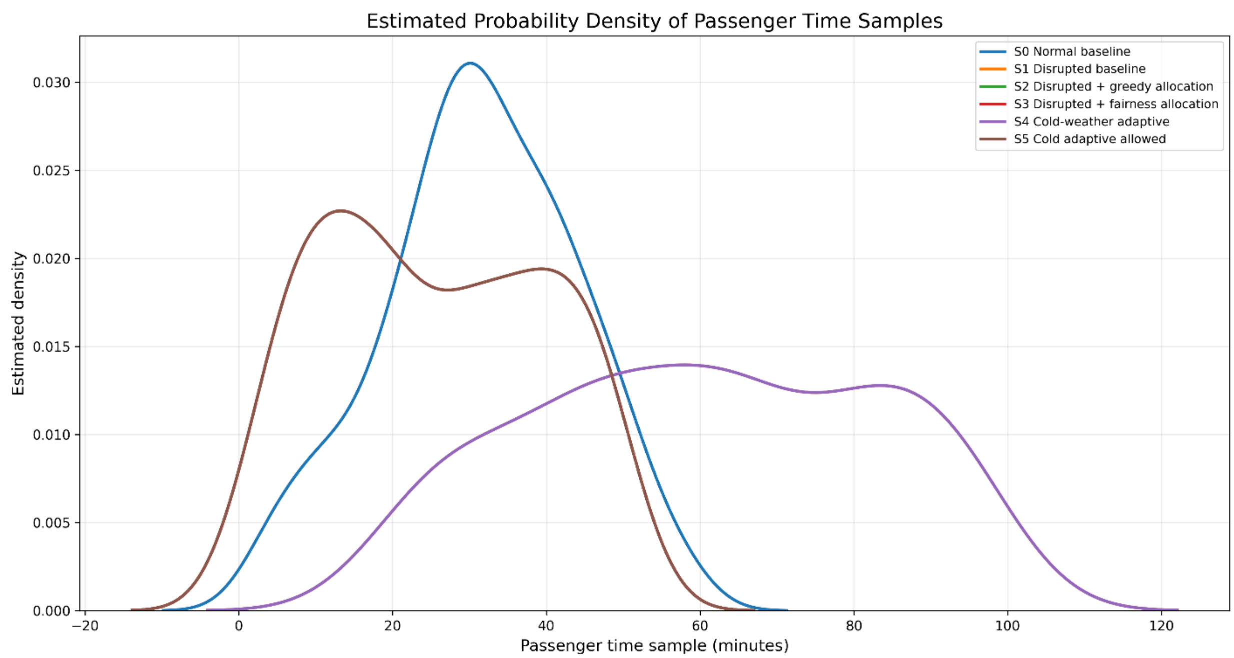

Figure 19 Estimated probability density of passenger time samples

The estimated density plot in Figure 19 provides a smoother view of the same distributional behaviour. The disrupted baseline cases show a rightward shift, indicating that passenger time samples are more likely to occur at higher values. In contrast, the dynamic allocation cases show higher density at lower time values. This supports the conclusion that accepted dynamic allocation improves the distribution of passenger time outcomes under disruption.

The shape of the probability density is useful for interpreting the stability of the system. A distribution concentrated around lower values suggests a more predictable and stable passenger experience. A distribution spread across higher values indicates that the system produces more long-delay outcomes. In this experiment, the dynamic allocation scenarios produce a more favourable distribution than the disrupted baseline.

The probability distribution plots therefore provide more information than a single average value. They show not only that disruption increases passenger time, but also how the entire distribution shifts under different operating policies. This is important for analysing public transport service systems, where the frequency of long passenger delays is often as important as the average delay.

4.8. Cumulative Probability Distribution Analysis

While the histogram and probability density plots show the general shape of the passenger time distributions, the cumulative probability distribution provides a more direct way to evaluate service reliability. A cumulative distribution function shows, for each time value, the probability that a passenger time sample is less than or equal to that value. In other words, it answers the question: what proportion of passengers experience a time not exceeding a selected threshold?

This type of plot is especially useful for public transport analysis because service quality is often evaluated using acceptable waiting or travel-time limits. For example, if a cumulative distribution reaches 0.90 at 45 minutes, then approximately 90% of passenger samples are no greater than 45 minutes. The same information can also be used in reverse: if only 48% of samples are below 60 minutes, then 52% of samples exceed 60 minutes. Figure 20 shows the cumulative distribution of passenger time samples for the main scenarios.

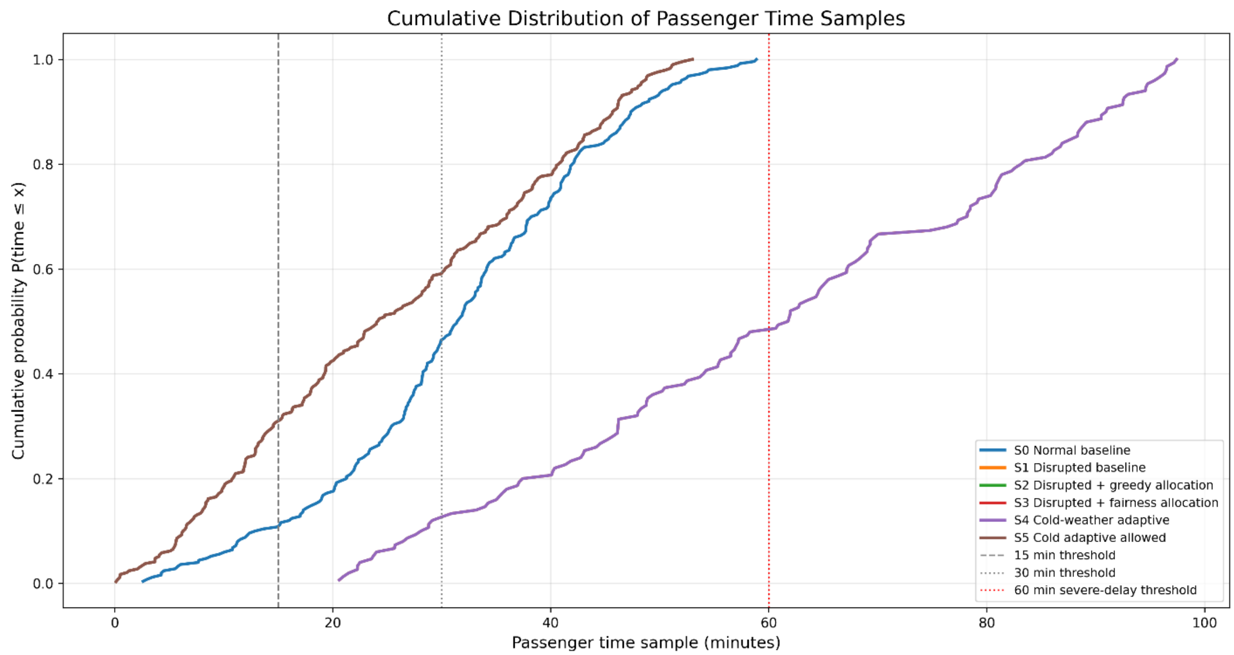

Figure 20 Cumulative distribution of passenger time samples by scenario

As shown in Figure 20, the disrupted baseline scenario is shifted to the right compared with the normal baseline. This indicates that a larger share of passengers experience longer time values when disruption affects bus arrivals and no dynamic allocation response is available. The curve for the disrupted baseline rises more slowly, meaning that passengers accumulate below shorter time thresholds at a lower rate.

The dynamic allocation scenarios rise more quickly and are shifted to the left. This means that a higher proportion of passengers complete the modeled process within shorter time values. In particular, the dynamic allocation scenarios reach high cumulative probability before the severe-delay range, while the disrupted baseline remains spread across much higher values.

The cold-weather adaptive scenario S4 behaves similarly to the disrupted baseline because allocation is rejected under the stricter cold-weather threshold. In this scenario, **temperatureC = -15**, which reduces the acceptable affected-route wait to 10 minutes. Since **affectedRouteHeadway = 15**, the system does not apply dynamic allocation. The cumulative distribution therefore remains close to the disrupted baseline. This result is important because it shows that the weather-adaptive policy does not simply optimize the hub at any cost; it may reject allocation when the affected-route impact is considered unacceptable.

Scenario S5 provides the contrasting case. Here, **temperatureC = -15** is still used, but **affectedRouteHeadway = 10**, which satisfies the cold-weather threshold. As a result, allocation is accepted, and the cumulative distribution moves back toward the improved dynamic allocation scenarios. This demonstrates that the weather-adaptive policy is conditional: it allows intervention in cold weather only when the affected-route waiting impact remains within the stricter acceptable limit.

Overall, the cumulative distribution analysis provides the strongest visual evidence for the effect of dynamic allocation. It shows not only that average passenger time changes, but also how the probability of remaining below selected time thresholds changes across scenarios. This directly supports a more probabilistic interpretation of the simulation results.

4.9. Threshold Exceedance Probability Analysis

To make the cumulative probability results easier to interpret numerically, threshold exceedance probabilities were also calculated. These probabilities show how likely it is that a passenger time sample exceeds selected values. In this experiment, the thresholds of 15, 30, 45, and 60 minutes were used. These values were selected to represent increasing levels of passenger delay severity, from moderate delay to severe delay.

Table 7 presents the threshold exceedance probabilities for each scenario.

Table 7 Probability of passenger time samples exceeding selected thresholds

Scenario	P(time > 15)	P(time > 30)	P(time > 45)	P(time > 60)
S0 Normal baseline	0.892	0.536	0.160	0.000
S1 Disrupted baseline	1.000	0.873	0.733	0.520
S2 Disrupted greedy allocation	0.691	0.409	0.117	0.000

Scenario	P(time > 15)	P(time > 30)	P(time > 45)	P(time > 60)
S3 Disrupted fairness allocation	0.691	0.409	0.117	0.000
S4 Cold-weather adaptive rejected	1.000	0.873	0.733	0.520
S5 Cold-weather adaptive allowed	0.691	0.409	0.117	0.000

Figure 21 visualizes the same probabilities using a grouped bar chart.

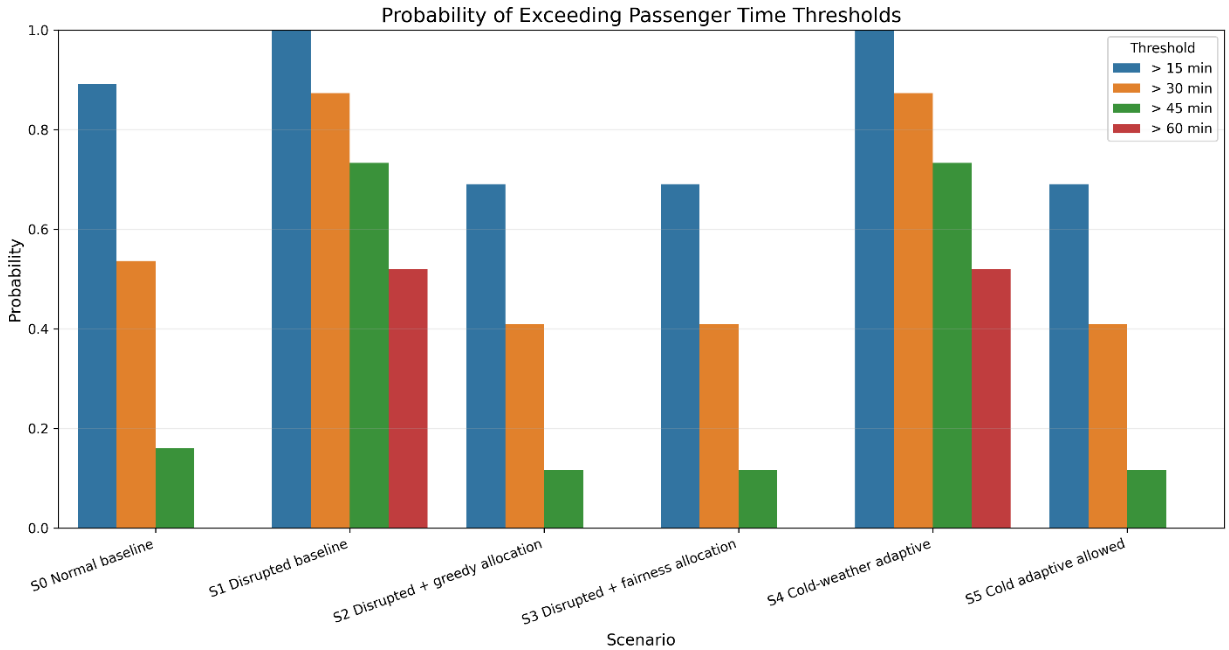

Figure 21 Probability of exceeding selected passenger time thresholds

Table 7 and Figure 21 show that the disruption scenario substantially increases the probability of excessive passenger time. In the normal baseline scenario S0, no passenger sample exceeds 60 minutes. However, in the disrupted baseline scenario S1, 52.0% of passenger samples exceed 60 minutes. This indicates that the disruption mechanism does not only increase the average passenger time; it also creates a substantial upper-tail risk.

The dynamic allocation scenarios S2 and S3 reduce this upper-tail risk. In both cases, the probability of exceeding 60 minutes falls to 0.0%. The probability of exceeding 45 minutes also decreases from 73.3% in the disrupted baseline to 11.7% in the dynamic allocation scenarios. This result shows that dynamic allocation can reduce not only the mean passenger time, but also the probability of severe passenger time outcomes.

Scenario S4 again matches the disrupted baseline because the weather-adaptive policy rejects allocation under cold-weather constraints. In this case, the system protects the affected route from an unacceptable additional wait, but the hub-side distribution remains poor. This illustrates the trade-off between reducing congestion at the hub and protecting passengers who may be negatively affected by dynamic service adjustment.

Scenario S5 shows that when the affected-route headway is within the cold-weather threshold, the weather-adaptive allocation can still provide the same improvement as the other accepted allocation scenarios. The probability of exceeding 60 minutes falls to 0.0%, and the probability of exceeding 45 minutes falls to 11.7%. This confirms that the weather-adaptive policy is not simply conservative in all cold-weather cases, but conditional on the estimated affected-route impact.

The threshold exceedance analysis is useful because it translates the distribution into direct probability statements. Instead of saying only that one scenario has a higher average than another, it becomes possible to state how likely passengers are to experience long time values. This is more informative for evaluating service reliability under disruption.

4.10. Discussion of Dynamic Allocation and Cold-Weather Constraints

The results of the extended experiment show that disruption has a strong negative effect on passenger time outcomes. When disruption is introduced without any dynamic response, both the mean and the upper percentiles of passenger time increase substantially. This is visible in the summary statistics, the distribution plots, and the cumulative probability analysis.

Dynamic bus allocation changes this behaviour when the allocation condition is satisfied. In the tested scenarios, accepted dynamic allocation reduces the mean passenger time from 60.70 minutes in the disrupted baseline to 25.43 minutes. It also reduces the 95th percentile from 94.69 minutes to 48.14 minutes. Most importantly, it reduces the probability of passenger time exceeding 60 minutes from 52.0% to 0.0%. This indicates that the benefit of dynamic allocation is not limited to improving average performance. It also reduces the risk of extreme passenger time outcomes.

However, the fairness and weather-adaptive scenarios show that dynamic allocation should not be interpreted as automatically beneficial in every situation. If additional bus capacity is assigned to the congested hub by taking or reallocating capacity from another route, passengers on the affected route may experience longer waits. The model represents this impact using the affectedRouteHeadway parameter. Allocation is accepted only when this estimated affected-route impact remains within the allowed threshold.

The cold-weather scenarios illustrate this trade-off clearly. In S4, the temperature is set to -15°C. Under this condition, the allowed affected-route wait is reduced to 10 minutes. Since the affected route headway is 15 minutes, allocation is rejected. This protects passengers on the affected route from excessive waiting under severe cold conditions, but it also means that the hub does not receive the additional capacity and the passenger time distribution remains similar to the disrupted baseline.

In S5, the temperature remains -15°C, but the affected route headway is reduced to 10 minutes. This satisfies the cold-weather threshold, so allocation is allowed. As a result, the passenger time distribution improves and matches the other accepted dynamic allocation scenarios. This comparison between S4 and S5 is important because it demonstrates that the weather-adaptive policy is not simply disabling dynamic allocation in cold weather. Instead, it applies a stricter condition for accepting allocation.

From a service-system perspective, these results show that the value of dynamic allocation depends on both system disruption and fairness constraints. A purely greedy policy can improve the congested hub, but it may ignore the cost imposed on passengers elsewhere. A fairness-protected or weather-adaptive policy provides a more cautious response. It can still improve performance when the affected-route impact is acceptable, but it may reject intervention when the cost to affected passengers is too high.

This trade-off is central to the interpretation of the extended model. The purpose of the simulation is not to claim that dynamic allocation should always be used. Instead, the model demonstrates how a discrete-event simulation can be used to evaluate when such a policy improves the distribution of passenger outcomes and when it should be restricted due to fairness or environmental constraints.

4.11. Limitations of the Extended Model

The extended model improves the analytical depth of the initial simulation by introducing probabilistic disruption, dynamic bus allocation policies, passenger time-sample collection, and distribution-based evaluation. However, the model remains a simplified representation of a public transport service system. These limitations are important to state clearly, because the purpose of the model is not to reproduce the full Vilnius public transport network, but to analyse system behaviour under controlled simulation conditions. Therefore, the numerical results should be interpreted as scenario-based simulation results rather than direct operational predictions for Vilnius.

First, the model focuses on a simplified hub-based structure rather than a complete city-wide transport network. The model includes passenger generation, a hub queue, bus arrivals, pickup logic, disruption delay, and dynamic allocation decisions. This structure is sufficient for analysing congestion and passenger time outcomes at a selected hub, but it does not represent all stops, routes, transfer points, and schedule dependencies of a real public transport network. In a full-scale implementation, additional routes and stops could be represented by adding more passenger arrival streams, bus sources, route-specific headways, and destination probabilities.

Second, the affected route is represented through the **affectedRouteHeadway** parameter rather than by explicitly modelling every downstream stop on the affected route. This parameter estimates the waiting impact imposed on passengers when bus capacity is dynamically allocated to the congested hub. The simplified approach is useful because it allows the model to include fairness constraints without significantly increasing model complexity. However, it does not capture the full spatial distribution of passengers along the affected route. A more detailed model could represent each downstream stop separately and calculate passenger waiting times at each location.

Third, the disruption mechanism is represented as a general aggregated factor. The parameter **disruptionProbability** is used to model the probability that a bus arrival is affected by adverse conditions, while **disruptionDelayMinutes** represents the additional delay in disrupted operation. This general disruption mode can represent traffic congestion, weather-related slowdown, or minor incidents. In reality, these causes may have different probabilities, durations, and effects on different locations. They were combined into one factor in this thesis to keep the model manageable within the available research scope. Future extensions could separate traffic, weather, accidents, vehicle breakdowns, and other incident types into individual probabilistic factors.

Fourth, the model uses passenger time samples recorded at the exit point of the simplified process. These samples are suitable for comparing scenarios because they are collected consistently across all simulation runs. However, they represent observed passenger time in the model rather than a complete decomposition into separate waiting, boarding, and in-vehicle travel components. A more detailed future model could record separate time components for each passenger, including queue waiting time, boarding time, in-vehicle time, and total journey time.

Fifth, the current model uses fixed bus capacity but does not fully randomize the number of passengers already inside a bus before it reaches the hub. In real public transport systems, a bus may arrive almost empty, partially occupied, or almost full. This affects how many passengers can board at the hub. The current model includes bus capacity as a constraint, but future work could extend this by sampling the available remaining capacity from a probability distribution depending on time of day or route load.

Finally, the experiment uses synthetic but transparent parameters rather than real operational data from the Vilnius public transport system. This means the results should be interpreted as a controlled simulation study rather than a direct prediction of actual Vilnius service performance. The value of the model lies in demonstrating how discrete-event simulation can be used to compare dynamic service policies, disruption effects, and passenger time distributions under clearly defined assumptions.

4.12. Chapter Summary

This chapter extended the initial AnyLogic model into a more analytical simulation experiment based on probabilistic disruption and empirical passenger time distributions. The extension introduced a general disruption mechanism, dynamic bus allocation policies, weather-adaptive constraints, and passenger time-sample collection. The collected samples were exported and analysed using Python to generate probability distributions, cumulative probability distributions, threshold exceedance probabilities, and percentile-based metrics.

The results show that disruption substantially worsens passenger time outcomes. In the disrupted baseline scenario, the mean passenger time increased to 60.70 minutes, and 52.0% of passenger samples exceeded 60 minutes. This demonstrates that disruption affects not only average performance but also the upper tail of the passenger time distribution.

When dynamic bus allocation was accepted, the distribution improved significantly. In the greedy, fairness-protected, and cold-weather allowed scenarios, the mean passenger time decreased to 25.43 minutes, and the probability of exceeding 60 minutes decreased to 0.0%. This shows that dynamic allocation can reduce the probability of extreme passenger time outcomes under sustained disruption.

The weather-adaptive results also demonstrate an important trade-off. In the cold-weather rejected scenario, allocation was not allowed because the affected route headway exceeded the stricter cold-weather threshold. As a result, the system behaved similarly to the disrupted baseline. This shows that a policy designed to protect passengers under harsh weather conditions may reduce the ability of the hub to recover from disruption. However, when the affected route headway satisfied the cold-weather threshold, dynamic allocation was accepted and the improved distribution returned.

Overall, the extended experiment shows that the usefulness of dynamic bus allocation cannot be judged only by average waiting time. A more complete evaluation requires analysing the probability distribution of passenger time outcomes, especially the probability of long delays. This distribution-based analysis provides a more informative basis for comparing public transport service policies under uncertain and disrupted operating conditions.

5 Results and Conclusions

The aim of this thesis was to analyse flow and delays in a dynamic public transport service system using discrete-event simulation. The work focused on a simplified hub-based public transport model, where passenger arrivals, bus service, queue formation, and dynamic service adjustment policies were represented in AnyLogic. The model was used to investigate how different operating conditions affect passenger time outcomes and how dynamic bus allocation can be used as a response to congestion and disruption.

The first result of the work is the construction of a working discrete-event simulation model of a simplified public transport hub. The model includes passenger generation, passenger queueing, bus generation, bus capacity, passenger pickup logic, and policy-based dynamic service adjustment. The model also includes parameters for passenger arrival intensity, bus headway, hub congestion threshold, acceptable waiting-time threshold, affected route headway, temperature, disruption probability, and disruption delay. This makes the model configurable and allows different operational scenarios to be tested without changing the overall model structure.

The second result is the introduction of dynamic bus allocation policies. The model distinguishes between baseline operation, greedy dynamic allocation, fairness-protected allocation, and weather-adaptive allocation. The baseline policy represents standard operation without dynamic adjustment. The greedy policy allocates additional service capacity when the hub queue exceeds the congestion threshold. The fairness-protected policy adds a constraint that prevents allocation when the affected route headway would exceed the acceptable waiting threshold. The weather-adaptive policy further strengthens this constraint under cold conditions. This policy structure makes it possible to analyse not only whether dynamic allocation improves hub performance, but also whether it may create unacceptable effects for passengers elsewhere.

The third result is the addition of a probabilistic disruption mechanism. The model separates normal operation from disrupted operation. In normal operation, buses receive only a small additional delay. In disrupted operation, a larger delay is applied, representing the combined effect of traffic congestion, weather-related slowdown, or minor operational incidents. This makes the model more realistic than a purely fixed-headway simulation and allows the effect of adverse operating conditions to be analysed.

The fourth result is the collection and analysis of passenger time samples. Instead of relying only on average and maximum values, the extended analysis used empirical passenger time samples collected from simulation runs. These samples were exported and analysed using Python. The analysis included summary statistics, percentile values, probability distribution plots, cumulative probability distributions, and threshold exceedance probabilities. This reflects the stochastic nature of the service system more clearly than a single average value.

The experimental results show that disruption has a substantial negative effect on passenger time outcomes. In the normal baseline scenario, the mean passenger time was 31.39 minutes and the 95th percentile was 51.05 minutes. In the disrupted baseline scenario, the mean increased to 60.70 minutes and the 95th percentile increased to 94.69 minutes. The probability of a passenger time sample exceeding 60 minutes also increased from 0.0% in the normal baseline scenario to 52.0% in the disrupted baseline scenario. This confirms that disruption affects not only the average passenger time, but also the upper tail of the passenger time distribution.

The results also show that accepted dynamic bus allocation can significantly improve the disrupted system. In the greedy and fairness-protected allocation scenarios, the mean passenger time decreased to 25.43 minutes and the 95th percentile decreased to 48.14 minutes. The probability of exceeding 60 minutes decreased to 0.0%. This demonstrates that dynamic allocation can reduce the probability of extreme passenger time outcomes under sustained disruption, not only improve the average result.

The fairness and weather-adaptive scenarios show that dynamic allocation should not be interpreted as automatically beneficial in every situation. When the affected route headway is within the acceptable threshold, dynamic allocation is allowed and improves the passenger time distribution. However, when cold-weather constraints reduce the acceptable affected-route waiting time, allocation may be rejected. In the cold-weather rejected scenario, the system behaves similarly to the disrupted baseline because the policy prevents intervention. This demonstrates the trade-off between improving the congested hub and protecting passengers who may be affected by the service adjustment.

Overall, the thesis shows that discrete-event simulation is a useful tool for analysing dynamic public transport service systems under uncertainty. The model allows input parameters and policy rules to be changed, while the resulting passenger time distributions can be compared across scenarios. The use of cumulative probability distributions and threshold exceedance probabilities provides a more informative view of system behaviour than average and maximum values alone.

The work has several limitations. The model uses a simplified hub-based structure and does not reproduce the full Vilnius public transport network. The affected route is represented through the **affectedRouteHeadway** parameter rather than by explicitly modelling all downstream stops. The disruption mechanism is represented as one aggregated factor instead of separate traffic, weather, accident, or vehicle breakdown processes. Passenger time samples are recorded at the exit point of the simplified process and therefore represent observed time-in-system rather than a fully decomposed waiting, boarding, and travel-time measure. The model also uses synthetic but transparent parameters rather than real operational data.

Future work could extend the model by adding more routes, more stops, real timetable data, route-specific passenger demand, and live or historical bus arrival data. The disruption mechanism could be separated into multiple probabilistic factors, such as traffic congestion, winter weather, vehicle breakdown, and road incidents. Bus occupancy could also be modelled probabilistically, so that buses may arrive partly full and not always be able to board all waiting passengers. Finally, passenger time could be decomposed into separate waiting time, boarding time, in-vehicle time, and total journey time, allowing more detailed analysis of service quality.

In conclusion, the thesis achieved its main objective by developing and extending a discrete-event simulation model for analysing flow and delays in a dynamic public transport service system. The final analysis demonstrates that probabilistic disruption can significantly increase the likelihood of long passenger times, while dynamic bus allocation can reduce this risk when allocation is allowed by the policy constraints. The results also show that fairness and weather-adaptive constraints are important because improving one part of the system may create negative effects elsewhere. Therefore, dynamic service adjustment policies should be evaluated not only by average performance, but also by their effect on the full distribution of passenger time outcomes.

References

[Abb16] ABBAS, M.; MACHIANI, S. G. Agent-based modeling and simulation of connected corridors: merits evaluation and future steps. International Journal of Transportation, 2016, vol. 4, no. 1, pp. 71–84. DOI: 10.14257/ijt.2016.4.1.05.

[Alt07] ALTIOK, T.; MELAMED, B. Simulation Modeling and Analysis with Arena. Amsterdam: Academic Press, 2007. ISBN 978-0-12-370523-5.

[Ant24] ANTELMI, A.; CORDASCO, M.; SPAGNUOLO, C.; SENSI, D. D. Reliable and efficient agent-based modeling and simulation. Journal of Artificial Societies and Social Simulation, 2024, vol. 27, no. 2. DOI: 10.18564/JASSS.5300.

[Any26a] AnyLogic. AnyLogic: simulation modeling software tools and solutions [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/>.

[Any26b] AnyLogic. Selected models [online]. [accessed 2026-02]. Available at: <https://cloud.anylogic.com/models>.

[Any26c] AnyLogic. From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/resources/articles/from-system-dynamics-and-discrete-event-to-practical-agent-based-modeling-reasons-techniques-tools/>.

[Any26d] AnyLogic. Why use multimethod modeling? [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/blog/why-use-multimethod-modeling/>.

[Any26e] AnyLogic. Multimethod modeling. In: AnyLogic Help [online]. [accessed 2026-02]. Available at: <https://anylogic.help/anylogic/ui/multimethod.html>.

[Any26f] AnyLogic. Multimethod modeling and simulation with AnyLogic [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/resources/educational-videos/multimethod-modeling-and-simulation-with-anylogic/>.

[Any26g] AnyLogic. Multimethod modeling [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/use-of-simulation/multimethod-modeling/>.

[Any26h] AnyLogic. Agent-based simulation modeling [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/use-of-simulation/agent-based-modeling/>.

[Any26i] AnyLogic. Process modeling library. In: AnyLogic Features [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/features/libraries/process-modeling-library/>.

[Any26j] AnyLogic. Transport planning simulation tool for suburban public transportation scheduling. In: AnyLogic Case Studies [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/resources/case-studies/transport-planning-simulation-tool-for-suburban-public-transportation-scheduling/>.

[Any26k] AnyLogic. Collecting statistics. In: AnyLogic Help [online]. [accessed 2026-02]. Available at: <https://anylogic.help/tutorials/bank-office/4-collecting-utilization-statistics.html>.

[Any26l] AnyLogic. Measuring time in system [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/resources/educational-videos/measuring-time-in-system/>.

[Any26m] AnyLogic. Collection. In: AnyLogic Help [online]. [accessed 2026-02]. Available at: <https://anylogic.help/anylogic/data/collection-variables.html>.

[Any26n] AnyLogic. Synthetic data generation. In: AnyLogic Artificial Intelligence Features [online]. [accessed 2026-02]. Available at: <https://www.anylogic.com/features/artificial-intelligence/synthetic-data/>.

[Asl20] ASLANSEFAT, K.; KABIR, S.; GHERAIBIA, Y.; PAPADOPOULOS, Y. Dynamic fault tree analysis. In: Reliability Management and Engineering. Boca Raton: CRC Press, 2020, pp. 73–112. DOI: 10.1201/9780429268922-4.

[Aut25] AUTODESK. Discrete event simulation (DES): the engine behind data-driven manufacturing decisions [online]. 2025 [accessed 2026-02]. Available at: <https://www.autodesk.com/blogs/design-and-manufacturing/how-discrete-event-simulation-helps-manufacturers-make-better-decisions/>.

[Ban10] BANKS, J.; CARSON, J. S.; NELSON, B. L.; NICOL, D. M. Discrete-Event System Simulation. 5th ed. Upper Saddle River: Pearson Education, 2010.

[Ban97] BANISTER, D.; WATSON, S.; WOOD, C. Sustainable cities: transport, energy, and urban form. Environment and Planning B: Planning and Design, 1997, vol. 24, no. 1, pp. 1–25.

[Bon02] BONABEAU, E. Agent-based modeling: methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences, 2002, vol. 99, suppl. 3, pp. 7280–7287. DOI: 10.1073/PNAS.082080899.

[Bor20] BORSHCHEV, A.; GRIGORYEV, I. The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 8 [online]. AnyLogic, 2020 [accessed 2026-02]. Available at: <https://www.anylogic.com/resources/books/big-book-of-simulation-modeling/>.

[Bro21] BRONIEC, W.; AN, S.; RUGABER, S.; GOEL, A. K. Guiding parameter estimation of agent-based modeling through knowledge-based function approximation. In: Proceedings of the Knowledge-Based Systems Workshop. CEUR Workshop Proceedings, 2021. Available at: <https://ceur-ws.org/Vol-2846/paper20.pdf>.

[Car16] CARO, J. J.; MÖLLER, J. Advantages and disadvantages of discrete-event simulation for health economic analyses. Expert Review of Pharmacoeconomics & Outcomes Research, 2016, vol. 16, no. 3, pp. 327–329. DOI: 10.1586/14737167.2016.1165608.

[Cos25] COSTESCU, D.; et al. Analyzing public transport wait times and identifying the most affected users in the metropolitan area of Valparaíso, Chile. Applied Sciences, 2025, vol. 15, no. 11, article 5969. DOI: 10.3390/APP15115969.

[Eur24] EURYSTIC SOLUTIONS. Analytical models vs simulation models in supply chain management: what each is used for [online]. 2024 [accessed 2026-02]. Available at: <https://eurysticsolutions.com/2024/10/15/analytical-models-vs-simulation-models-in-supply-chain-management-what-each-is-used-for/>.

[Ext26] EXTENDSIM. Understanding different modeling methodologies [online]. [accessed 2026-02]. Available at: <https://extendsim.com/solutions/simulation/methodologies>.

[Fit11] FITZSIMMONS, J. A.; FITZSIMMONS, M. J. Service Management: Operations, Strategy, Information Technology. 7th ed. New York: McGraw-Hill Education, 2011. Available at: <http://www.mhhe.com/fi>.

[Fiv26] FIVEABLE. Simulation modeling and analysis: introduction to industrial engineering class notes [online]. [accessed 2026-02]. Available at: <https://fiveable.me/introduction-industrial-engineering/unit-10>.

[Gao24] GAO, C.; GAO, M.; PECAN, N.; DING, Z.; REN, J.; QIN, B. Large language models empowered agent-based modeling and simulation: a survey and perspectives. Humanities and Social Sciences Communications, 2024, vol. 11, no. 1, article 509. DOI: 10.1057/s41599-024-03611-3.

[Gil08] GILBERT, N. Agent-Based Models. Los Angeles: SAGE Publications, 2008.

[Gro85] GROSS, D.; HARRIS, C. M. Fundamentals of Queueing Theory. 2nd ed. New York: John Wiley & Sons, 1985. ISBN 0-471-04521-1.

[Ind26] Engineering Industries Excellence (INDX). AnyLogic multimethod simulation modeling software [online]. [accessed 2026-02]. Available at: <https://www.indx.com/en/product/anylogic-simulation-software>.

[Kel15] KELTON, W. D.; SADOWSKI, R. P.; ZUPICK, N. B. Simulation with Arena. 6th ed. New York: McGraw-Hill Education, 2015. Available at: <http://www.mhhe.com>.

[Kha24] KHAN, S.; DENG, Z. Agent-based crowd simulation: an in-depth survey of determining factors for heterogeneous behavior. The Visual Computer, 2024, vol. 40, no. 7. DOI: 10.1007/s00371-024-03503-2.

[Law15] LAW, A. M. Simulation Modeling and Analysis. 5th ed. New York: McGraw-Hill Education, 2015.

[Lyu23] LYU, Z.; PONS, D.; PALLIPARAMPIL, G.; ZHANG, Y. Optimising urban freight logistics using discrete-event simulation and cluster analysis: a stochastic two-tier hub-and-spoke architecture approach. Smart Cities, 2023, vol. 6, no. 5, pp. 2347–2366. DOI: 10.3390/SMARTCITIES6050107.

[Mac09] MACAL, C. M.; NORTH, M. J. Agent-based modeling and simulation. In: Proceedings of the 2009 Winter Simulation Conference (WSC). IEEE, 2009, pp. 1–30. DOI: 10.1109/WSC.2009.5429318.

[Maj16] MAJID, M. A.; FAKHRELDIN, M.; ZUHAIRI, K. Z. Comparing discrete event and agent based simulation in modelling human behaviour at airport check-in counter. In: Lecture Notes in Computer Science. Vol. 9731. Cham: Springer, 2016, pp. 510–522. DOI: 10.1007/978-3-319-39510-4_47.

[Mar24] MARTINS, A. L.; FERREIRA, J. C.; KOCIAN, A.; TOKKOZHINA, U.; HELGHEIM, B. I.; BRÅTHEN, S. (eds.). Intelligent Transport Systems. Vol. 540. Springer, 2024. DOI: 10.1007/978-3-031-49379-9.

[May95] MAY, A. D.; ROBERTS, M. The design of integrated transport strategies. Transport Policy, 1995, vol. 2, no. 2, pp. 97–105. DOI: 10.1016/0967-070X(95)91989-W.

[Mis25] MISHRA, S.; SINHA, S.; SINGH, N. Agent-based modeling: insights into consumer behavior, urban dynamics, grid management, and market interactions. Energy Strategy Reviews, 2025, vol. 57, article 101613. DOI: 10.1016/j.esr.2024.101613.

[Mit25] MITZENMACHER, M.; SHAHOUT, R. Queueing, predictions, and LLMs: challenges and open problems. arXiv:2503.07545 [online]. 2025 [accessed 2026-02]. Available at: <https://arxiv.org/abs/2503.07545>.

[Mur98] MURRAY, A. T.; DAVIS, R.; STIMSON, R. J.; FERREIRA, L. Public transportation access. Transportation Research Part D: Transport and Environment, 1998, vol. 3, no. 5, pp. 319–328. DOI: 10.1016/S1361-9209(98)00010-8.

[Ndi15] NDIH, E. D. N.; CHERKAOUI, S. Simulation methods, techniques and tools of computer systems and networks. In: Modeling and Simulation of Computer Networks and Systems: Methodologies and Applications. Amsterdam: Elsevier, 2015, pp. 485–504. DOI: 10.1016/B978-0-12-800887-4.00017-1.

[Ngo20] NGO-HOANG, D.-L. The three methods in simulation modeling [online]. 2020. DOI: 10.13140/RG.2.2.29143.09125.

[Noo18] NOORJAX CONSULTING. Agent-based, discrete-events and system dynamics [online]. 2018 [accessed 2026-02]. Available at: <https://noorjax.com/2018/09/20/simulationparadigms/>.

[Ols22] OLSEN, M.; RAUNAK, M. S. Efficient parameter exploration of simulation studies. In: Proceedings of the 2022 IEEE 29th Annual Software Technology Conference (STC). IEEE, 2022, pp. 190–191. DOI: 10.1109/STC55697.2022.00034.

[Ols24] OLSEN, M.; KUHN, D. R.; RAUNAK, M. S. Explaining the impact of parameter combinations in agent-based models. Journal of Computational Science, 2024, vol. 81, article 102342. DOI: 10.1016/j.jocs.2024.102342.

[PTC26] PTC. Analytical methods and simulation overview [online]. [accessed 2026-02]. Available at: <https://support.ptc.com/help/wrr/r12.0.2.0/en/index.html#page/wrr/ReferenceGuide/rbd/analytical_methods_simulation_overview.html>.

[Ros09] ROSS, S. M. Introduction to Probability Models. 10th ed. Amsterdam: Academic Press, 2009. ISBN 978-0-12-375686-2. DOI: 10.1016/C2009-0-30640-6.

[Sar10] SARGENT, R. G. Verification and validation of simulation models. In: Proceedings of the 2010 Winter Simulation Conference (WSC). IEEE, 2010, pp. 166–183. DOI: 10.1109/WSC.2010.5679166.

[Sim26] SIMIO LLC. Differences between discrete, continuous, and agent-based simulation software [online]. [accessed 2026-02]. Available at: <https://www.simio.com/what-are-the-differences-between-simulation-software-discrete-continuous-and-agent-based/>.

[Spe26] SPEC INNOVATIONS. The role of simulation in informed decision-making [online]. [accessed 2026-02]. Available at: <https://specinnovations.com/blog/the-role-of-simulation-in-informed-decision-making>.

[Sri15] SRIKANTHAN, S.; DWARKADAS, S.; SHEN, K. Data sharing or resource contention: performance transparency on multicore systems. In: Proceedings of the 2015 USENIX Annual Technical Conference (USENIX ATC 15). Santa Clara, CA: USENIX Association, 2015, pp. 529–540. Available at: <https://www.researchgate.net/publication/279940863_Data_Sharing_or_Resource_Contention_Toward_Performance_Transparency_Multicore_Systems>.

[Thi14] THIELE, J. C.; KURTH, W.; GRIMM, V. Facilitating parameter estimation and sensitivity analysis of agent-based models: a cookbook using NetLogo and R. Journal of Artificial Societies and Social Simulation, 2014, vol. 17, no. 3, article 11. DOI: 10.18564/JASSS.2503.

[Tra18] TRACY, M.; CERDÁ, M.; KEYES, K. M. Agent-based modeling in public health: current applications and future directions. Annual Review of Public Health, 2018, vol. 39, pp. 77–94. DOI: 10.1146/ANNUREV-PUBLHEALTH-040617-014317.

[Tsa12] TSAI, C.-H.; MULLEY, C.; CLIFTON, G. The spatial interactions between public transport demand and land use characteristics in the Sydney greater metropolitan area. 2012.

[Tur16] TURRELL, A. Agent-based models: understanding the economy from the bottom up. Bank of England Advanced Analytics Division [online]. 2016 [accessed 2026-02]. Available at: <https://ssrn.com/abstract=2898740>.

[Uni24] United Nations, Department of Economic and Social Affairs, Population Division. World Population Prospects 2024 [online]. 2024 [accessed 2026-02]. Available at: <https://population.un.org/wpp/graphs?loc=900&type=Probabilistic%20Projections&category=Population&subcategory=1_Total%20Population>.

[Ver82] VERMA, A.; RAMANAYYA, T. V. Economics of public transport. In: Public Transport Planning and Management in Developing Countries. Routledge, 1982, pp. 249–268. DOI: 10.1201/B17891-13.

[Xia05] XIANG, X.; KENNEDY, R.; MADEY, G.; CABANISS, S. Verification and validation of agent-based scientific simulation models. In: Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Data. Chicago, IL, 2005.

[Zhe06] ZHENGPING, L.; CHENG, H. S.; MALCOLM, Y. H. L. A survey of emergent behavior and its impacts in agent-based systems. In: 2006 IEEE International Conference on Industrial Informatics. IEEE, 2006, pp. 1295–1300. DOI: 10.1109/INDIN.2006.275846.

[Zhe13] ZHENG, H.; SON, Y.-J.; CHIU, Y.-C.; HEAD, L.; FENG, Y.; XI, H.; KIM, S.; HICKMAN, M. Chapter 1: Agent-based modeling and simulation – a primer for agent-based simulation and modeling in transportation applications. Federal Highway Administration Report FHWA-HRT-13-054 [online]. 2013 [accessed 2026-02]. Available at: <https://www.fhwa.dot.gov/publications/research/ear/13054/002.cfm>.