Department of Computer Science

Submitted in part fulfilment for the degree of BSc.

Optimisation of Train Scheduling to Reduce Delay Propagation

Daniel Kirkpatrick

27/04/2025

Supervisor: Detlef Plump

Contents

List of Figures

List of Tables

1 Introduction

1.1 Project Motivation

The UK rail network serves as a vital transportation backbone, fuelling economic growth while providing essential social and political benefits. However, the increasing demand for more frequent and efficient train services has exposed the network's capacity constraints, posing a significant challenge to its ability to meet growing needs. Having worked at Jacobs on the Transpennine Route Upgrade alongside Network Rail, I have experienced the troubles with resolving delays and delay propagation and their impact on passengers and the workforce. Delays within the network have become a critical issue, with only 68.3% of trains between January 1st and March 31st, 2024, arriving within one minute of the scheduled time, classified as "On Time" [1]. These delays adversely affect economic productivity, passenger satisfaction, and environmental sustainability [2] [3], as delay means that the railways can not be utilised to their fullest capacity.

While new technology such as European Train Control System (ETCS)[4] promise a long-term infrastructure improvement to reduce delay, delay remains a constant operational challenge, especially when multiple services share a limited resource such as track or platform space. In this context, optimising the order and timing of train movements becomes critical, as it can reduce knock-on delays caused by conflicts and improve punctuality. This project explores how optimisation techniques such as Ant Colony Optimisation (ACO) and dynamic rescheduling can be applied to reduce delays and improve overall railway efficiency in a simulation, and therefore in practice.

1.1.1 Prior Work

Research into delay mitigation has increasingly turned toward optimisation-based approaches, which are well-suited to modelling and improving complex, resource-constrained systems like railways. Tiong and Palmqvist [5] found that nearly 48% of delay management methods in the literature are based on optimisation techniques, including timetable design, delay

recovery algorithms, and real-time rescheduling.

In particular, a study in 2012 by Fan et al. [6] showed the effectiveness of ACO in reducing delay at a 4-way junction by up to 30%, with more recent work by Mishra et al. [7] showing it is still effective in real-world systems. Building upon this, the current project applies ACO alongside rescheduling techniques to evaluate their combined impact on the UK rail network.

1.2 Aims and Objectives of the project

This project aims to create and test a simulation-based optimisation model for train scheduling that will reduce delay and increase operational efficiency.

The project will focus on:

- Analysing existing train scheduling techniques to select an appropriate technique to be used in our algorithm.
- Identify inefficiencies related to delay in the selected techniques that could be improved in our algorithm.
- Develop an optimisation framework that reduces delay and delay propagation by adjusting train schedules dynamically based on real-time data
- Evaluate the effectiveness of the optimisation algorithm for real-time prioritisation of train movement.
- Comparing performance of the optimised schedule against a baseline model, as well as other literature.

By achieving these goals, the project contributes to the growing field of intelligent railway optimisation and offers insight into how real-time decision-making can help address capacity and delay issues in modern rail networks.

2 Literature Review

2.1 Punctuality

Punctuality measures how closely a train adheres to its timetable, specifically the deviation of its arrival at a station. Trains arriving within a certain threshold are considered punctual. This is typically expressed as a percentage: the number of punctual trains divided by the total number of trains. Delay, on the other hand, is a continuous measure of deviation, recorded in minutes. For instance, a train might be 10 minutes late. Network Rail defines punctuality as "the lateness experienced at each recorded station stop" [8].

Punctuality is widely regarded as one of the most critical metrics for evaluating the performance of transportation networks, particularly railways, and will be a key metric for assessing the improvements made by improved scheduling. This is true not only in the UK but also in other countries, such as the Netherlands and Japan [9], [10]. Its importance lies in its direct impact on passenger satisfaction. Monsuur notes that passenger satisfaction drops below 50% after a delay of 30 minutes and decreases further under conditions such as overcrowding or when passengers are commuting rather than travelling for leisure [2].

Different countries use varying indicators to assess punctuality. In the UK, the Public Performance Measure (PPM) evaluates whether a train reaches its final destination "on time" [1]. Trains are considered "on time" if they arrive at the destination within 5 or 10 minutes of the timetable, depending on whether they are short- or long-distance services. Sweden employs a similar metric called the Combined Performance Measure, while Germany, the Netherlands, and Denmark use stricter thresholds. For instance, in Germany, trains are considered "on time" if they are no more than 2 minutes late for suburban services and 5 minutes late for long-distance trains. Sweden uses a threshold of under 6 minutes for all passenger trains. Stricter thresholds, such as those in Germany, often result in more reliable train schedules and higher passenger satisfaction [11].

2.1.1 Limitation of PPM

While metrics like PPM provide a high-level view of punctuality, they focus solely on arrival times at final destinations, potentially obscuring issues at intermediate stations. Both Skagestad and Mukunzi [11] highlight this as a significant limitation. For example, in the first quarter of 2024, the UK's PPM was 86.8%, but only 68.3% of trains arrived on time at all stations along their route [1]. This suggests that focusing only on the endpoint can mask delays experienced elsewhere.

Evaluating punctuality at every station stop, rather than just the final destination, could offer a more accurate picture of network performance. Such an approach would help identify operational shortcomings and improve passenger satisfaction by addressing the root causes of delays.

2.2 Delay and Delay Propagation

2.2.1 Delay Causes

The importance of punctuality has prompted substantial research into the factors contributing to timetable deviations, more commonly referred to as delay causes. Delays caused in this context can be defined as the reason why a train has arrived behind schedule at a station, resulting in the loss of punctuality. Some common examples include broken rails, signal failures, flooding and trespassing. These causes can then be organised into different high-level categories, such as infrastructure, weather or unexpected external factors. Network Rail has grouped delay causes into different categories based on their characteristics, such as infrastructure or extreme weather, as well as by responsible parties, such as themselves or the Train Operating Companies (TOC). According to the data presented, the majority of delays (58%) , including propagating delays, stem from Network Rail infrastructure, causing operational issues such as buckled rails, as well as external, unforeseeable causes as trespasses or vehicles causing damage to infrastructure, such as a bridge [12]. Additionally, 25% of delays were caused by problems with the TOC, which include defective trains, lack of staff and late timings leaving stations. However, recent studies have shown that although there is a large variety in the delay causes, the majority of delays within rail networks are caused by propagating from these incidents, with nearly double the amount of these secondary delays than primary, causal delays [13].

2.2.2 Primary and Secondary Delays

To distinguish between delay propagation and root delay, delays can be categorised into primary and secondary. Secondary delays are also known as knock-on delays or 'delays arising as a consequence of another delay' [13]. Primary delays are where the causes of delay are external to the timetable, such as infrastructure problems or weather. Changes to the timetable may cause secondary delays due to interactions with other trains on the network that have been delayed. For example, a primary delay may be a train stuck due to a broken signal, and a secondary delay may be the delay to the trains stuck behind this train. A study by Palmqvist in 2023 [13] using simulations estimated that about 36% of observed delays in the region of Skåne, Sweden, were primary delays and that 64% were secondary delays. This figure highlights the importance of having measures to reduce delay propagation, as for every primary delay, there are roughly 1.8 secondary delays. The method used by Palmqvist to separate primary and secondary delay involved modelling the rail network in Skåne as a simulation using an application called PROTON and then introducing empirical data into the simulation as primary delay. The simulation was then used to record the total punctuality over 3 different days. Mapping this data against overall punctuality over 21 different scaling levels of delay allows the calculation of secondary delay.

The study by Palmqvist provides valuable insights into the dynamics of primary and secondary delays, and it highlights the cascading nature of disruptions and underscores the importance of measures to mitigate delay propagation. However, the study's reliance on a macroscopic simulation tool, PROTON, may limit the quality of its findings, as such models often simplify detailed train interactions. These limitations of macroscopic simulations are discussed later on. Additionally, the focus on Skåne raises the question of whether these results of a rough 1.8 times secondary delay to primary delay can be applied to other rail systems with different scopes. Despite these limitations, the method of injecting a fixed amount of primary delay data into a system to calculate secondary delay is robust and provides a simple method for similar studies.

2.3 Dealing with Delay Propagation

With how much delay propagates from primary delay, reducing secondary delay has become a major objective for train operating companies to ensure the smooth running of the network and passenger satisfaction. Extensive literature has emerged highlighting different approaches for studying and

dealing with delay propagation, which fall into different overarching methods such as optimisation, simulations and so forth, as commented on by Tiong and Palmqvist [5] in their brief overview in 2023. In the paper, it was found that 48% of these methods fell under optimisation due to their effectiveness at working with specific scenarios. Due to this, we will be focusing on looking into the optimisation methods in this section.

2.4 Optimisation Techniques

Optimisation is making the most efficient and effective use of the resources at hand to solve a problem. In terms of delay propagation and the rail network, optimisation methods and models attempt to find the best solution for reducing delay through different focuses such as creating optimal time tables for capacity utilisation while minimising delay [14], delay recovery algorithms[15] or rescheduling. These optimisation methods create mathematical models around the formulated problem and can be expressed as graphs, algebra or other formulas such as mixed integer programming. They are then solved using approximate methods such as Tabu Search [16] or exact methods such as Lagrangian relaxation [17].

2.4.1 Margin Reallocation

Train timetables are often created with pre-emptive 'buffer' time supplements added into the programme to absorb any delay that may occur while the train is on the track. This buffer time is either added while the train is running, known as running time supplements, or between consecutive train movements at a track/station, known as headway buffer, as seen in Figure 2.1 (Headway refers to the minimum distance that must be kept between consecutive trains). The larger the additional time allocated, the more likely it is that the trains will arrive within the planned time. This has the trade-off of using more time than required in reality, meaning fewer trains can be allocated on the network.

In Meng's 2019 paper [14], they propose modelling the train timetable as a Directed Acyclic Graph (DAG) and making use of the Critical Path Method (CPM) [18] to reallocate existing margins for runtime supplements and headway buffers to significant activities of the train timetable. The network is expressed as the graph G(V, E), where vertices/nodes (V) are used to represent the state of the network and edges (E) are used to represent a task, such as moving from station to station, and the relationship between states. The time taken for the activities is added along the path to find the

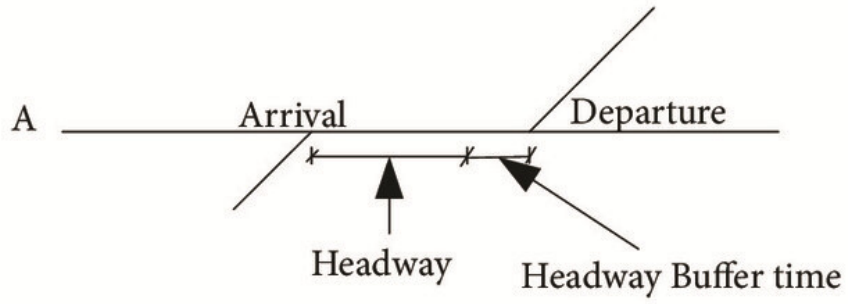

Figure 2.1: Illustration of Headway Buffer

total duration, with the longest duration being the critical path. The margins are then reallocated using the average of the three allocation equations below, which allocate based on the repetition of activity on paths and the length of paths for activity. M represents the number of original margins, N represents the number of times an activity appears on total paths (P), L is the distance of the route on which an activity appeared, and A is the total number of activities. t and s represent the path of an activity. Further detail on the steps for using the equations can be found in the paper [18].

$$m_{1_{i,(t,s)}} = M \cdot \frac{N_{i,(t,s)}}{P}. \tag{2.1}$$

$$m_{2_{i,(t,s)}} = M \cdot \frac{\text{Avg}L_{i,(t,s)}}{\sum \text{Avg}L_{i,(t,s)}}. \tag{2.2}$$

$$m_{3_{i,(t,s)}} = M \cdot \frac{\text{Avg}L_{i,(t,s)}}{L_c \cdot A \cdot \sum \left(\frac{\text{Avg}L_{i,(t,s)}}{L_c \cdot A}\right)}. \tag{2.3}$$

where:

- M = Total margin time available for allocation
- $N_{i,(t,s)}$ = Number of paths containing activity $i(t,s)$
- P = Total number of paths in the CPM network
- Avg$L_{i,(t,s)}$ = Average length of paths where activity $i(t,s)$ appears
- Sum$L_{i,(t,s)}$ = Sum of the lengths of paths where activity $i(t,s)$ appears

- L_c = Length of the critical path
- A = Total number of activities in the CPM network
- $t \in T$ = Train index from the set of trains T
- $s \in S$ = Segment index from the set of segments S

The result of this method showed a 3.25% decrease in delay on small rail networks, and a 5.18% reduction for large networks, which suggests that shifting the time supplements to more areas of higher activity is beneficial. This technique requires the original timetable to be using time supplements, which if not required will increase the time required for a train to complete its activity. However, modern timetables all tend to have buffer times to absorb delay, with larger networks having more due to the increased unpredictability of delays. On the other hand, moving the time supplements to where we can predict higher delay to occur will reduce the flexibility of the network to deal with unexpected conditions such as equipment failure or weather disruptions. This may mean that trains will not have enough buffer time to recover from unanticipated events.

2.4.2 Rescheduling

Train rescheduling is the process of adjusting planned train schedules with different strategies, such as train reordering, retiming and rerouting, to maintain safety and increase passenger satisfaction. Rescheduling aims to quickly generate a solution that provides minimal increased time cost compared to the original schedule. In a 2023 paper by Zhang et al. [19], the problem of rescheduling a large-scale network is modelled as an integer linear programming model. The study considered train reordering and retiming, as well as the option of rerouting due to sections of the network being paralysed. Reordering refers to a change in the sequence in which trains are dispatched at stations or junctions, such as FCFS. Retiming is the adjustment of arrival times and dwell times to reduce conflicts, which can make use of buffer time as mentioned before. Rerouting is the change of a planned route to avoid conflicts or delays, and may miss out on stations that were on the route. In the paper, the problem is solved using a subset of constraints from the model, employing the Lagrangian relaxation (LR) [17] method to create the subset. A heuristic algorithm is then used based on this (LR-H). This was then applied to real-world instances in China, where it was found to have a 2.27% optimality gap, meaning it was very close to the true solution. The downside of this method is that it will miss out on stations that were on the original route, and effectively abandon those passengers who wished to depart or board at that stop.

Another method [20] suggests using the max-plus algebra method [21] for creating the model and focusing on the retiming of trains. It looks at making adjustments through dwell time reduction, section acceleration, more or less overtaking, and train postponement. For each delayed train at least one of these adjustment methods will be selected. Using these, an approximate model is created, where the target function is the minimum number of delayed trains, and the best outcome of the combination of adjustments is found to fit this. This was tested on the timetable of the Beijing-Shanghai High-Speed Railway and was found to be valid and feasible. However, the method has only been tested for high-speed rail, due to the larger distance and higher speed allowing for more adjustments and a larger effect on delay.

2.4.3 Priority and Conflict Resolution

Priority in rail refers to the order in which trains are given access to infrastructure, such as junctions or railway platforms. Normally, the priority assignment rules come on a FCFS basis, however by using dynamic and fixed priority rules, trains which are delayed or a certain type can be given higher priority to reduce delay propagation. A 2012 paper [6] compared the effectiveness of 8 different priority rules on a 4-direction rail junction. It made use of the simple cost function below to decide on an optimal solution, where DT represents the delay time for each train, DP is the delay penalty, θ is the ordering of trains, and $J(\theta)$ is the total delay cost for a given ordering. Of the methods tried, it found that FCFS is the fastest and should be applied as a baseline to most scenarios, however, Ant colony optimisation (ACO) [22] provides the best solutions with practical computation times, for both complex and simple problems. ACO was found to reduce delay cost by around 30% for the problem.

$$J(\theta) = \sum_{i=1}^{n} DT_i(\theta) \cdot DP_i \tag{2.4}$$

Modern priority-based train scheduling methods still make use of ACO, such as this study done on the Indian railway [7], proving that it is still effective to this day. In this paper, ACO was used for a single-line track problem where there is a conflict with overtaking at a junction. It is proven to beat the manual methods for solving the problem currently used for the Indian Railway, with a 6.827% reduction in travel time. However, they also compare the effectiveness of the CPLEX solver [23]. CPLEX solver is a tool by IBM for solving mathematical optimisation problems. For the problem at hand, both ACO and CPLEX produced comparable results, however, when applied to a real network problem CPLEX showed a larger

percentage improvement. The drawbacks of CPLEX are the commercial licensing costs if required for a commercial tool and the high computational resource requirements.

2.5 Simulations

Simulation modelling of networks has vast amounts of benefits and has been used on railways to model and predict delay, evaluate freight train operations and evaluate what capacity of rolling stock a railway can function with [13], [24]. Simulation models are useful for the analysis of networks as they simplify the problem into a set of mathematical calculations using assumptions and variables to calculate metrics about the system. These can then be used to evaluate the impact of the user-input variables and to answer questions about the system. Due to the functionality of simulations, many different simulation programs have been designed for different problems in the railway. Some examples of such are OpenTrack [25], which is widely used for simulating train operations and delays, and RailSys [26], which is for detailed modelling of railway networks and operational planning. Simulation tools can also be split into three different high-level categories: Microscopic, Macroscopic and Mesoscopic modelling.

2.5.1 Microscopic Modelling

Microscopic simulations are the most detailed form of simulation, and can simulate individual train movements, how they interact with signalling systems, as well as the exact placement of switches and signals. Due to this, microscopic models tend to give the most accurate recreation of the network, however, this is traded for a high computational cost, meaning it is best suited for small to medium networks, or looking at a snapshot of a larger network.

Microscopic simulations tend to run in time blocks set by the user. Using this, the simulation can model the impact of trains on each other, such as if a train is occupying a block in a station that another train is trying to get into, the train will have to wait at the block entry signal until it indicates that the block is free. The block in this context would represent a platform [24]. Microscopic Models can further be split into two groups: synchronous and asynchronous. The difference between these is that synchronous models are built at the same time, in one run of the model, whereas asynchronous will build up the model with multiple runs, creating different layers. Asynchronous models are particularly good for constructing timetables as

they replicate the trains running in perfect condition. The model starts by taking a set of the highest priority trains, such as direct long-distance trains, and modelling them, before taking the next set of trains and modelling them. This means that the priority trains cannot be affected by the rest of the trains however, these trains will be affected by the priority group [24]. With the improvement in technology and the increasing density of networks, asynchronous models are being used less and less, however, they still have their benefits in specialised applications where computational efficiency and simplicity are priorities. They may also be used in conjunction with synchronous approaches. On the other hand, synchronous models simulate all operations at the same time, with the user creating rules on what the trains do if they run into conflicts, such as train priorities and overtaking [24]. This creates a more realistic model and is better for use in dispatching and reacting to real-time data, as well as looking at certain metrics such as the impact of train delays on the network. Although synchronous models have many benefits, Palmqvist notes that larger nodes, such as stations with complex track layouts, can lead to deadlocks in these types of models. This is where a certain number of trains become unable to move as they are each blocked by another train [13]. Running into a deadlock effectively makes the simulation useless, and so they should be minimised or removed. The chance of a run resulting in a deadlock depends on many factors, including the track and route, the implemented rules and the tool that was used. Additionally, the type of track lines may result in more deadlocks, such as if the majority of tracks in the area were single-track lines, as was the case in Skane. Solutions for this problem were investigated by Pachl [27] in 2011, with the suggestion of reserving multiple track sections ahead of a train in the simulation to avoid conflicts, however, the Palmqvist paper from 2023 suggests that this is still a problem.

Examples of microscopic simulation tools include OpenTrack and RailSys, both well-known tools used in a variety of different research projects on the railway [24], [28]. Some more modern projects include models for simulating the capacity of railway lines in Czech and looking at the effect of lineside automatic train protection [28], [29].

2.5.2 Macroscopic Modelling

Macroscopic simulations are more abstract models compared to microscopic simulations and tend to focus on network flows as a whole and metrics such as average delays and capacity utilisation. Due to this, macroscopic models hold an advantage when modelling large networks as it is less data-intensive with a lower computational cost. Unfortunately, it can't analyse individual train behaviour and has limited ability to assess local delay propagation [24].

A common approach for macroscopic simulations is to represent tracks and stations as nodes and edges, an example of this being PROTON. Nodes can also be used to represent smaller sections of a large station, and trains will arrive/depart from nodes. Edges will be weighted with different metrics, such as line speed and directions, as well as the number of tracks and block length. A key difference in macroscopic is that the track layout within the nodes is not modelled. This means any simultaneous arrival or departure can occur, which in reality should not be the case. Train conflicts are modelled by block and train headway which is the minimum time in block. Track lines can be modelled by blocks similar to microscopic models, but they may also be aggregated into one large edge. The study by Palmqvist in 2023 used PROTON when modelling the Swedish rail network due to its size and to remove the chance of deadlocks [13]. LUKS is another form of macroscopic simulation software for the rail, which was used for capacity analysis of rail lines in Germany [30].

2.5.3 Mesoscopic Modelling

Mesoscopic simulations are an effective in-between of microscopic and macroscopic models, allowing for the simplification of certain areas such as track layout, while still having detail in areas such as infrastructure or signalling where it is relevant. The benefits of this mean that the computational cost of the model can be reduced while still retaining important details. Of course, the drawback of this is that it won't be as detailed as a microscopic simulation or as abstracted as macroscopic simulations for larger areas, meaning that if you were focusing a study on either of these areas, you may as well choose whether to focus on detail or the larger picture. An example of mesoscopic modelling is Johansson et al. mesoscopic approach for their simulation on early freight train departure [31].

3 Design and Implementation

The goal of this project is to come up with an optimisation algorithm to reduce delay in the UK rail network. For that purpose, we will create a simulated environment to run and test this algorithm based on the UK rail's constraints and attributes. This section will explain the method and design behind this and the reasoning for the different design choices made. It will describe the scope of the network, the design of the simulation model and the optimisation methods and techniques being used in the algorithm.

3.1 Geographical Scope

This project focuses on optimising the UK rail network by selecting the Transpennine Route Upgrade (TRU) [32] area as our case study. TRU is a major railway enhancement project between York and Manchester and aims to improve connectivity and productivity in the North by planned infrastructure upgrades such as electrification, track doubling between Huddersfield and Dewsbury, and station improvements. These changes create a complex scheduling environment, making it an ideal case study for developing and testing optimisation techniques. Additionally, we will focus on the main TRU corridor, the red line in Figure 3.1, which is entirely double track—a configuration that represents much of the UK rail network. Most lines in the UK are double track, except in certain areas like the East Coast Main Line, which has four tracks in sections such as London to Grantham. A double-track system consists of two parallel tracks, one for each direction (up and down), allowing trains to travel simultaneously in both directions. We will also be including the diversionary routes, seen in grey in Figure 3.1, but only where they provide an alternative track loop back into the main corridor, such as Church Fenton to South Milford to Micklefield. Stations such as Selby will not be modelled as they do not contribute to an alternate route. These diversionary routes will allow rerouting algorithms to be tested, allowing for more optimisation methods to be used.

3.2 Simulation Model

3.2.1 Overview

A simulation will be used to model how the rail network will interact. A simulation method was chosen for this study due to its effectiveness in handling complex systems such as a rail network. Rail networks involve numerous variables, such as train schedules, station capacities, and infrastructure constraints, making it difficult to predict the outcomes of any changes directly. By using a simulation, we can model these factors in a controlled environment, allowing us to evaluate different optimisation techniques and their impact on network performance. Additionally, the ability to test multiple scenarios makes it a good choice for seeing how techniques work when faced with different problems. Its effectiveness is also seen in previous studies on delay and optimisation [13], [19].

Originally, the plan was to use prebuilt simulation software such as OpenTrack [25], a well-recognised train simulation, as this would eliminate the work of creating a simulation model from scratch. However, this was not feasible as it is only available for commercial use or those with an academic license. After researching some free alternatives, such as RailOS [33], and finding it difficult to use for the project, it was decided that the best way to meet all the requirements for the simulations was to create a model from scratch. This would provide flexibility over how it works, having the ability to easily customise the different configurations and schedules using the source code, as well as control over assumptions, which will be discussed later on. Additionally, by building the model from the ground up, we will know intuitively how the system works. The model will be mesoscopic, meaning it balances the level of detail between macroscopic (high-level network flow) and microscopic (individual train dynamics). This approach allows for the simplification of non-critical aspects while still modelling key elements, such as braking curves and station capacity constraints. This trade-off ensures a practical yet accurate representation, especially given the project's time limitations.

3.2.2 Model Design

The simulation will follow a discrete-event model, where system state updates occur at a specific time step based on scheduled events, such as train departures and arrivals. This method allows for more efficient handling of large-scale rail operations without the need for continuous real-time processing. This will reduce the computation requirements to simulate

larger models. The drawback of this is that it does not capture the real-time dynamics such as continuously changing speeds, and will have to use an approximation and assumptions. These assumptions can lead to reduced accuracy when compared to the real world. To mitigate this problem, we will use small incremental steps of 1 minute within the model.

The whole network will be represented as a directed weighted graph, where edges represent track segments between stations, nodes represent station objects, and weights represent the distance between stations in miles. This is an appropriate choice due to the route containing only double-track lines with predefined directions, meaning we can easily model the tracks to and from the station. This allows for the travel times on each edge to be calculated dynamically for each train. This will allow for different train speeds and to implement possible delay causes related to trains, such as lower speed limits. Modelling this way will not account for train signalling along the tracks, where trains normally can't move into the next signal block (this refers to a section of track where the whole length between two stations is split into different block sections) if there is a train in it. This will reduce the computational load significantly and simplify the overall logic.

Due to the use of object-oriented programming and graph-based modelling, Python [34] has been chosen to be used for implementation due to its flexibility, large range of libraries and ease of use for modelling complex systems when compared to programming languages with stricter rules, such as Java [35]. Libraries such as NetworkX can be used to model the graph network easily, and tools such as Pandas can allow for easy data evaluation through their visual outputs. Python is also fully open-source, unlike OpenTrack, meaning the experiments will be easy to recreate. Although it is not as fast as other lower-level languages such as C++ [36], faster processing is not required for discrete event modelling, where real-time processing is unnecessary.

To ensure the model reflects real-world constraints, operational limitations will be incorporated:

- Platform Availability: Trains will require a free platform before entering a station
- Minimum Dwell Time: Trains will have to wait at a station based on their dwell time requirements
- Train Conflicts: How trains deal with other trains
- Speed Limits: Different speed limits on track sections

Platform availability and dwell time can be dealt with by modelling the stations as objects with different attributes. Each station can then be given

a specific dwell time for passenger boarding, with trains having to wait at the station for at least that amount of time. Different stations have different dwell times; however, for this simulation, we will assume that trains need to wait 1 minute at minor stations and 2 minutes at major stations. Major stations can be differentiated by the large orange circles around the stations in Figure 3.1. Platforms will be assigned to the station as either up or down. When trains reach the end of an edge, they will need to check the availability of platforms. If there are none available, they will need to wait until one is free. Trains will also be given an attribute corresponding to the direction that they are travelling, which must match that of the platform.

Only one train can leave or enter a station in 1 time step. Therefore, the base model will work on an FCFS basis, with the next trains added to a queue. Additionally, trains are unable to overtake while on an edge, so they must wait for the train before they reach the end of the edge. Trains will also be given different speeds based on the type of train being used. Trains that do not stop at a station will enter and leave the station within the same minute.

3.3 Timetable

For the operation of the simulation, a timetable must be made with routes and schedules for different trains. To achieve this, we have observed existing train routes using the application TrainLine [37] to create the train timetable seen in Figure A.2. Specifically, we looked at train routes on weekdays that travel through at least two major TRU stations on the route and cover all the TRU stations in the route consecutively. For example, the York to Manchester Victoria stops at Stalybridge, and the major stations start at Redcar Central; however, we have cut out the stations before York. The train travelling from York to Leeds that travels through Poppleton has not been included, even though it contains York, Leeds and Church Fenton, because it stops at other stops between York and Leeds that are not on the TRU route. The train names are based on the train operating company, the starting station, the end station and the number of stops on the route. TPX_Leeds_MV_4 stands for Transpennine Express from Leeds to Manchester Victoria, with 4 stops on the route. The timetable focuses on 6 long-distance trains travelling between York and Manchester Victoria, with one train leaving from York or Manchester every 20 minutes, which is based on the frequency of long-distance trains on this route found on TrainLine. Additionally, two short-distance trains have been included from York to Leeds/Leeds to York and from Leeds to Manchester Victoria/Manchester Victoria to Leeds to increase the density of the network, based on real routes from TrainLine. In reality, the route would be much more densely

populated with trains from these routes and other routes, such as the stopping train between Leeds and York, but by modelling only these routes, we keep the simulation simple, making it easier to analyse the delay.

Each long-distance train is to complete a round trip and finish at its original station. Each short-distance train will complete 3 round trips. The arrival timing was calculated using the distance between the two stations plus one minute of buffer time. An assumption is made that the train crew will not need any extra time to switch to the other side of the train on the return journey.

3.4 Optimisation Techniques

This section will cover the optimisation techniques that will be implemented to reduce the amount of delay. We will focus on priority and conflict management, rescheduling and margin reallocation.

3.4.1 Priority and Conflict Resolution

Due to the limited track infrastructure on this route, with only double-track lines, the system is prone to many conflicts at station junctions. Ant Colony Optimisation (ACO) will be implemented to improve efficiency and minimise delays using dynamic conflict resolution and train priority rules. ACO has been selected due to its adaptive and self-optimising approach, unlike traditional methods such as FCFS or mixed integer linear programming. This makes it more suitable for solving dynamic, complex problems such as rail optimisation, as seen in previous studies by Mishra et al. [7] on the Indian railway and Fan's study on junction conflict [6].

ACO is built upon the behaviour of foraging ants, where pheromone trails guide good decision-making. This is suitable for train scheduling as it provides dynamic prioritisation by considering train delay, train type and past pheromone levels for making scheduling decisions, as well as adaptive learning where the algorithm remembers past successful decisions through pheromone levels. This will be added into the model whenever more than one train needs access to a shared resource - in the case of the model, it will be entering or leaving a station platform. When this occurs, trains requesting entry/exit to the same station will be placed into a queue. It is assumed that trains cannot overtake on the double-track sections where FCFS is used; however, they can do so when entering the stations.

Trains in the queue are then assigned a selection probability based on this formula [22]:

$$P(i) = \frac{\tau(i)^{\alpha} \cdot \eta(i)^{\beta} \cdot T(i)^{\gamma}}{\sum_{j \in T} \tau(j)^{\alpha} \cdot \eta(j)^{\beta} \cdot T(j)^{\gamma}} \tag{3.1}$$

where:

- $\tau(i)$ = Pheromone level (reinforces past good decisions)
- $\eta(i)$ = Heuristic desirability (based on delay, prioritising delayed trains)
- $T(i)$ = Train type weighting (express trains get higher priority)
- α, β, γ = Tunable parameters controlling the influence of pheromones, delays, and train type.

This formula expresses the probability of selecting a specific train from the queue based on the results of picking this train before (pheromone levels), the amount of delay being experienced by the train, and the priority of the train.. For pheromone levels, the higher the value of $\tau(i)$, the more successful instances of picking train i in the past have occurred. α simply represents the weight we have given to the historical success. We multiply this by the train's current delay, $\eta(i)$, which also has a weight assigned to it β, as well as the train priority expressed by $T(i)$. This allows us to express the probability of the train being chosen as a scalar value, where it is then normalised against all other trains j in the queue using the denominator so that they all add up to 1. This allows us to assign the probabilities of each train being chosen based on these conditions.

Using this, the algorithm will then select a train randomly based on the weighted probability and then update the pheromone levels based on the selected train's performance. This will be done in Python using the random module. The pheromone levels are updated using the formula below:

$$\tau(i) = (1 - \rho) \cdot \tau(i) + \Delta\tau(i) \tag{3.2}$$

where:

- ρ = Evaporation rate (prevents pheromones from accumulating too much).
- $\Delta\tau(i)$ = Pheromone deposit, rewarding trains that helped reduce delay.

This expresses the impact the train had on the delay in the network, whether it was positive or negative. The pheromone deposit is built up for trains that have a positive impact, leading to them being selected more frequently in the future.

This encourages future train conflicts to follow similar successful scheduling decisions, leading to self-optimising schedules over multiple runs. An example of this would be if two trains arrive at a station at the same time, and one is delayed by 1 minute and one by 3 minutes. It is more likely that the train with 3 minutes delay will be selected unless the 1-minute delayed train has a higher pheromone trail, meaning ACO has selected it and successfully reduced delay more times. Within our timetable, the long-distance trains will be given a higher priority in their train type, while short-distance trains will have lower priority. This is because long-distance trains stop at fewer stations along the way, whereas short-distance trains stop at more stations between two set points. This means if the short-distance trains are given priority, the long-distance trains would constantly have to stop behind them, instead of passing straight through stations.

3.4.2 Rescheduling

While ACO effectively schedules trains in a better order, it has the potential to create cascading delays by always focusing on the most delayed train. This can lead to an increase in overall system-wide delay. To reduce this, we will use a time window rescheduling, which dynamically adjusts train priorities while adjusting their scheduled arrival times when a train passes a set amount of delay. This will make sure that severely delayed trains do not disrupt the overall schedule. For the threshold of delay, we will use either 10 minutes for a long-distance train or 5 minutes for any other train. The reason for this is how the UK declares a train as arriving on time, with long-distance trains having to arrive within 10 minutes of their arrival time and 5 minutes for short-distance trains [1].

$$D_{\text{threshold}} = 5 \text{ or } 10 \text{ minutes}$$

For any train T_i, rescheduling is triggered if:

$$D_i > D_{\text{threshold}}$$

Once a train is rescheduled, its arrival time is shifted to the next available slot that avoids delaying the following trains. The original delay is preserved to measure the disruption's impact.

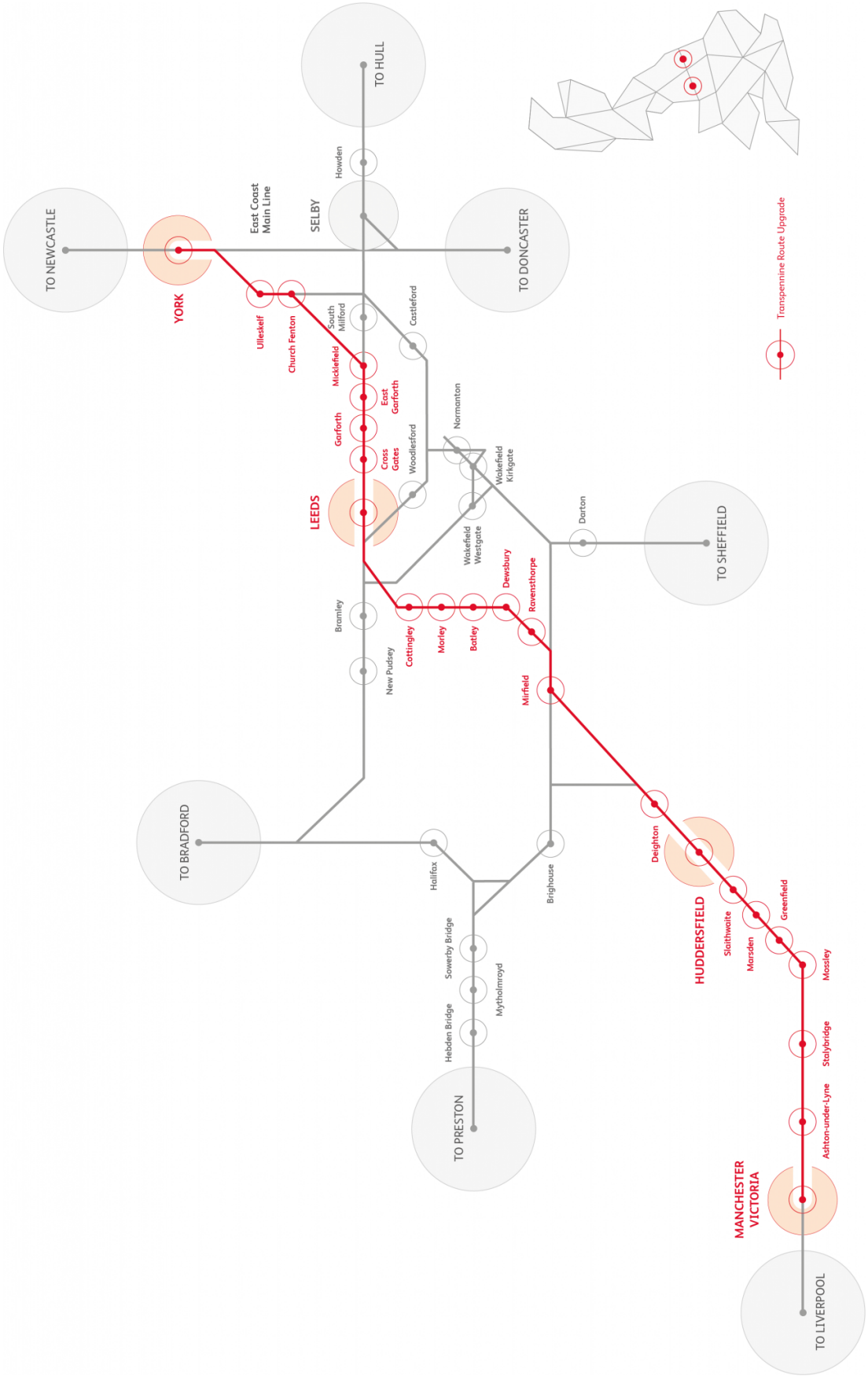

Figure 3.1: Diagram showing the Transpennine Route Upgrade Area

4 Experiments and Results

4.1 Experimental Setup

This section will cover experiments to validate the effectiveness of ACO with rescheduling. We will run controlled experiments to measure the key performance metrics of total delay and train punctuality with and without the above-mentioned optimisation method. The key research questions are as follows, as defined in our introduction:

- Does the optimisation algorithm decrease overall delay?
- Does the optimisation algorithm increase train punctuality?
- Does the optimisation algorithm reduce overall system-wide delay?

We will test the train simulations and tests under 3 different scheduling modes:

Table 4.1: Comparison of Baseline (FCFS), ACO-Only, and ACO + Rescheduling

Scenario	**Baseline (FCFS)**	**ACO-Only**	**ACO + Rescheduling**
Train Selection	FCFS	ACO selects best train based on pheromones	ACO selects best train with rescheduling adjustments
Delay Handling	No adjustments for delayed trains	ACO may overprioritize delayed trains	Rescheduling prevents cascading delays
Decision Adaptability	Static train order	Pheromone-based optimization	Dynamically adjusts dispatching to prevent disruptions

FCFS has been selected as a baseline test to compare against. Whilst it is not the most effective available method, we do not have access to the IBM CPLEX Solver due to not having a license. The best practice would be to compare against a manual method, as this is what is used currently in the UK industry alongside aids, as well as being the baseline used in other

literature that we will be comparing against. However, we do not have a way to do this for our simulation, and we assume that FCFS will give the closest comparison. Additionally, FCFS is a well-known and used method still in the railway industry that will allow us to easily see improvements. We will be running two tests to evaluate the effect of our optimisation algorithm. The first will be designed to evaluate the priority train impact by implementing a lower-priority train travelling through the network, which will intentionally be scheduled to create delays. We will then evaluate the different scheduling modes for dealing with this. Then we will evaluate the effect of random delay on the network, and how the different scheduling modes deal with this.

The hypothesis for these experiments is that FCFS will perform the worst, with no dynamic decision-making. ACO should improve scheduling, but it may cause an increase in cascading delay. Finally, ACO + Rescheduling should give the best solution, with dynamically adjustable train priorities.

4.1.1 Train Experiment

The purpose of the algorithm we are implementing is to optimise the ordering of trains receiving a scarce resource, in this case, platform allocation. Therefore, this experiment will aim to cause clashes within the different train timetables to observe the effect of each scheduling mode. This will be done by introducing two trains between York and Leeds, and two between Huddersfield and Leeds. We have chosen these trains as they stop at nearly all the stations they come across, making it a problem that can clash with other trains' timetables. They will have a train priority of 'local' and their timetable will intentionally clash with other trains to create delays. The timetable can be seen in Figure A.3.

These train routes are based on real train routes found again with TrainLine. We will evaluate the key metrics in this scenario between the different scheduling methods to identify which method is the best. Due to a degree of randomness in the selection of the next train, 50 tests are required to test for statistical significance for both ACO and ACO + Rescheduling.

4.2 Results

Results for each section will be displayed in tables, showing the total delay for each scheduling mode, and total train punctuality, and we will also look at the variance between the total delay results for ACO and

ACO+Rescheduling. In the experiments, due to ACO having a degree of randomness, we have taken the average of 50 test results for the results to be statistically significant. Train punctuality is calculated using the definition given to us by the Office of Rail and Road [1], where a train is declared on time if it arrives within 5 minutes of its scheduled time, or 10 minutes if it is a long-distance service. In this experiment, the trains were classified as long-distance travel between Manchester and York. As discussed in the literature review, we will not be using PPM to measure delay, as this focuses solely on the arrival time at the last destination, and may ignore delay issues in the middle of the route, which was recovered by the arrival buffer time added to each station along the route. Instead, we will take any new delay experienced at each station and add it to the total amount of delay experienced in the system, as seen in Table 4.2.

Next, we looked to improve the consistency of our best method at making good decisions by increasing the learning time of the algorithm. In the previous test, the algorithm would only learn over one day, before the pheromone trails indicating a good decision would be cleared. However, this time we allowed the pheromones to accumulate over 50 days, which we hoped to lead to more consistent positive scheduling decisions and lower variance. The results of this can be seen in Table A.4 in the appendix. To test whether the difference between the two methods was statistically significant, a two-sample t-test was performed on the 50 runs using Excel's built-in T.TEST function. The test compared the mean delays of the groups to see if it was likely due to chance, using a 0.05 significance level. The null hypothesis was that there is no difference in the mean delays between the two methods. The resulting p-value of 0.0286 indicated the difference was statistically significant, providing sufficient evidence to reject the null hypothesis.

4.2.1 Data

Table 4.2: ACO vs ACO+Rescheduling Delay and Punctuality

Test #	**ACO Total Delay**	**Punctuality**	**ACO+Resched Total Delay**	**ACO+Resched Punctuality**
1	76	71.43%	133	50%
2	139	50%	109	57.14%
3	86	71 3/7%	88	71.43%
4	74	71.43%	98	57.14%
5	126	57.14%	93	57.14%
6	123	64.29%	74	64.29%
7	125	50%	103	57.14%
8	119	50%	86	64.29%

9	91	57.14%	80	71.43%
10	84	64.29%	117	71.43%
11	73	57.14%	127	57.14%
12	96	71.43%	133	50%
13	113	64.29%	102	57.14%
14	106	57.14%	61	85.71%
15	105	57.14%	83	71.43%
16	94	71.43%	78	71.43%
17	129	57.14%	140	42.86%
18	131	50%	103	64.29%
19	117	64.29%	127	57.14%
20	139	42.86%	83	71.43%
21	138	57.14%	106	64.29%
22	137	42.86%	111	57.14%
23	115	50%	156	35.71%
24	101	50%	110	57.14%
25	94	64.29%	121	64.29%
26	147	57.14%	128	50%
27	87	71.43%	91	71.43%
28	93	64.29%	143	57.14%
29	136	71.43%	91	64.29%
30	120	64.29%	137	50%
31	111	50%	122	64.29%
32	118	50%	89	71.43%
33	114	57.14%	140	64.29%
34	82	64.29%	131	50%
35	76	64.29%	81	57.14%
36	143	71.43%	96	64.29%
37	153	42.86%	109	57.14%
38	87	57.14%	136	50%
39	146	71.43%	110	57.14%
40	155	42.86%	107	64.29%
41	152	35.71%	111	71.43%
42	104	57.14%	97	57.14%
43	112	57.14%	106	50%
44	165	64.29%	92	57.14%
45	139	42.86%	92	57.14%
46	103	57.14%	84	64.29%
47	104	64.29%	80	71.43%
48	98	50%	101	50%
49	95	57.14%	112	57.14%
50	68	64.29%	108	64.29%

FCFS did not have any randomness in this test and therefore only had to be tested once, with the results shown in Table 4.3.

Table 4.3: Comparison of Total Delay and Punctuality across different modes

Mode	**FCFS**	**ACO**	**ACO+Rescheduling**	**ACO+Reschduling with Learning**
Total Delay (mins)	119.00	112.78	106.32	94.26
Punctuality (%)	57.14	58.72	61.00	61.29

5 Evaluation

5.1 Evaluation Criteria

This chapter will evaluate the effectiveness of our chosen optimisation technique, ACO and rescheduling, against the following objectives: Reducing overall delay, improving train punctuality, and reducing delay propagation. The evaluation metrics we will be looking at will be total delay in minutes, punctuality, as well as variance of these to see the stability of the algorithm. Additionally, we will be looking at the best and worst performances for each of the three scheduling modes.

5.2 Baseline vs Optimisation Performance

Using the results from Table 4.3, when compared to the baseline FCFS method, ACO reduced total delay by 5.23% and slightly improved punctuality by 1.58%. This shows some improvement and falls in line with the improvement seen in Mishra's 2020 study [7], which saw a 6.826% improvement in delay using ACO on a single-line rail network. Adding rescheduling improved results further. It reduced total delay by 10.62% and increased punctuality by 3.86% over FCFS. This is because ACO by itself prioritises delayed trains, causing others to wait, reducing overall punctuality. Rescheduling, however, deprioritises trains once they are deemed late, focusing on keeping the other trains on time.

After implementing long-term learning into the algorithm through the use of pheromones, the results improved even further, as seen in Table A.4 with an average delay of 94.26 minutes. This is a 20.79% improvement over the baseline, 10.17% better than the algorithm without learning. Furthermore, we can observe a lower overall variance in the boxplot in Figure 5.1, and although the interquartile range appears to be similar, the mean is skewed lower. This means that it gives a lower delay more consistently than the other methods, due to its pheromone learning, allowing it to select trains that will reduce delay more frequently. A statistical t-test showed the difference between ACO + Rescheduling with and without learning was significant (p

= 0.0286). This means the improvements are not due to chance.

Overall, the results support that ACO is effective at reducing delay, and that rescheduling can make it more robust, particularly in congested situations or when fast and slow trains share tracks. When comparing these results to the literature, they align with Mishra's 2020 study [7], which found a 6.827% improvement in performance using ACO compared to FCFS in a single-line rail network. Our 5.23% improvement with ACO supports their findings. It's possible that had their study also included rescheduling, performance gains would have been even higher. Mishra also tested over a much shorter time period, over 24 minutes. When we tested learning over 50 days using our timetable, we saw a 20.79% increase in performance, suggesting that if they ran the test for longer, they would have seen greater improvements as well. However, their paper also showed that IBM's CPLEX solver produced much better solutions than ACO overall. As we did not have access to a CPLEX license, a direct comparison was not possible, so it is not clear whether CPLEX would perform better or not.

That said, while the average results are positive, ACO and rescheduling still have the potential to perform poorly. In the worst-case run, the delay increased by 31.09% compared to the baseline, and across 50 simulations, 12% of ACO + Rescheduling with long-term learning outcomes were worse than the baseline. In a real-world scenario, this could lead to widespread disruptions, customer dissatisfaction, and financial cost for train operators. These risks could be reduced by running the algorithm over a longer period to further improve decision-making. Additionally, the weights used in the ACO code could be tuned to yield better results, potentially through an automated tuning method such as Bayesian Optimisation, as mentioned in the literature review. Nevertheless, there may still be a need for human verification to oversee the outputs, especially in critical cases, to prevent severe disruptions.

5.3 Limitations

Several limitations must be considered when interpreting these results

- Real-time disruptions such as signal disruptions or track maintenance were not modelled, only timetable clashes over stations
- ACO pheromone learning only occurred over a single day; no long-term learning effects were evaluated.
- Track logic has been simplified, such as the use of signalling blocks, and trains requiring certain amounts of headway between them before

proceeding. This means the simulation will only give an idea of how the techniques will perform in the real world

- Assumed trains will have the same dwell time based on the station, which in reality will differ based on the class of train.
- Assumed trains will immediately reach the correct speed upon moving, not taking into consideration acceleration curves.
- The simulated model is a closed system, meaning trains from other routes that have to pass through the TRU corridor will not be taken into account.

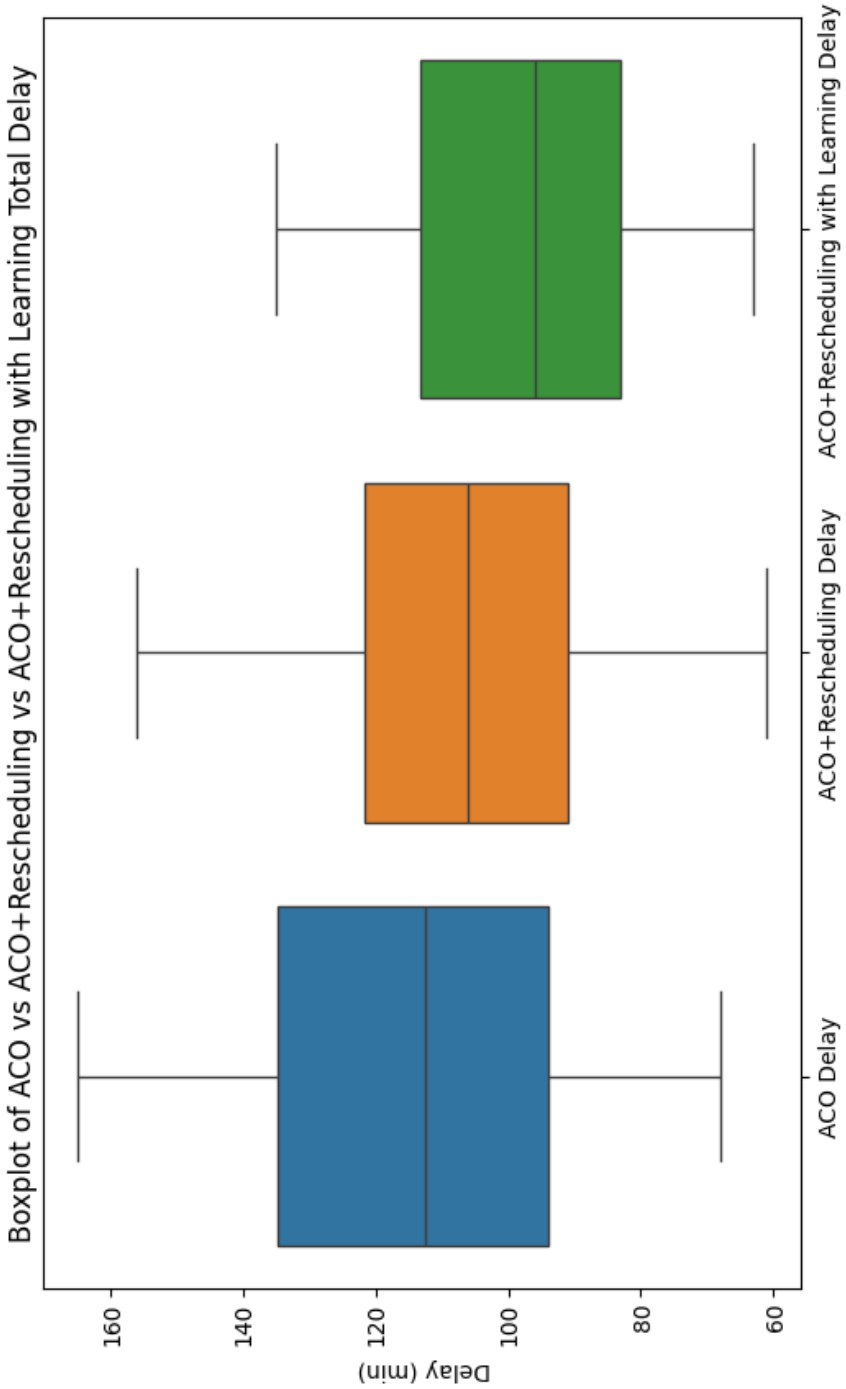

Figure 5.1: Boxplot of ACO vs ACO+Rescheduling vs ACO+Rescheduling with Learning Delay

6 Conclusion

In this project, we explored the use of optimisation techniques to reduce delay and delay propagation within the UK rail network. We developed a discrete-event simulation model in Python, mapping the Transpennine Route Upgrade corridor from Manchester to York. This simulation contained a simplified double-track rail network to simulate train movements in time steps, and incorporated track occupation, dwell times, platform constraints and directionality.

Within the simulation runs, we decided to use Ant Colony Optimisation (ACO) alongside threshold rescheduling to attempt to reduce delay. Through a series of controlled experiments, we evaluated the performance of First-Come-First-Served (FCFS), ACO and ACO with rescheduling. These experiments purposely used varied train types and schedules to create station occupation clashes to test these methods. Our results showed that ACO by itself reduced delay by 5.23%, and reduced delay by 10.62% when rescheduling is also implemented. It also fulfilled the goal of reducing delay propagation that could be caused by prioritising delayed trains, with a 2.28% increase over ACO by itself. Additionally, rescheduling appears to have reduced the variance in the results from the algorithm, leading to consistently better performance. We then further optimised with long-term learning over 50 days, which led to a further 10.17% increase in performance.

When compared to other literature, it was found that they align with the improvements seen in ACO over FCFS, suggesting that including rescheduling would further improve this, along with a longer learning time. However, although we were unable to test against the CPLEX solver, which is the current recognised best way to solve train rescheduling problems. The current implementation would need to be tested against this to prove this.

This project has demonstrated that optimisation techniques such as ACO and threshold scheduling can meaningfully improve train scheduling performance through the use of simulation modelling. It has laid the groundwork for further investigation into how to further optimise the algorithm for improved consistency, and for its use in more complex real-world applications in railway networks.

A Appendix

A.1 Acronym Table

Table A.1: Acronym Table

Acronym	Definition
ACO	Ant Colony Optimisation
CPM	Critical Path Method
DAG	Directed Acyclic Graph
ETCS	European Train Control System
FCFS	First Come First Served
LR	Lagrangian Relaxation
PPM	Public Performance Measure
TOC	Train Operating Company
TRU	Transpennine Route Upgrade

A.2 Original Train Schedule

Table A.2: Train Routes and Schedules

Main Line	**TPX_York_MV_4Y**	**TPX_MV_York_4**	**TPX_York_MV_5**	**TPX_MV_York_5M**	**TPX_York_MV_6**	**TPX_MV_York_6**
York	08:00		08:20		08:40	
Garforth						09:05
Leeds	08:35		08:55		09:16	
Dewsbury				09:10		09:31
Huddersfield	09:01		09:22		09:43	
Manchester Victoria	09:38	08:00	09:59	08:20	10:20	08:40
Huddersfield	10:15	08:36	10:36	08:55	10:57	09:16
Dewsbury			10:49	09:08	11:10	09:29
Leeds	10:41	09:02	11:03	09:22	11:24	09:43
Garforth					11:36	09:55
York	11:20	09:38	11:39	10:02	12:01	10:20
Garforth						10:46
Leeds		10:14		10:41		10:57
Dewsbury				10:56		11:12
Huddersfield		10:40		11:08		11:23
Manchester Victoria		11:17		11:45		12:01

A.3 Addtional Train Schedules

Table A.3: Additional Train Routes and Schedules

Train ID	Route	Schedule
TRX_NOR_York_Leeds_7	York - Church Fenton - Micklefield - East Garforth - Garforth - Cross Gates - Leeds - Cross Gates - Garforth - East Garforth - Micklefield - Church Fenton - York - Church Fenton - Micklefield - East Garforth - Garforth - Cross Gates - Leeds - Cross Gates - Garforth - East Garforth - Micklefield - Church Fenton - York - Church Fenton - Micklefield - East Garforth - Garforth - Cross Gates - Leeds - Cross Gates - Garforth - East Garforth - Micklefield - Church Fenton - York - Church Fenton - Micklefield - East Garforth - Garforth - Cross Gates - Leeds - Cross Gates - Garforth - East Garforth - Micklefield - Church Fenton - York	08:10, 08:28, 08:37, 08:41, 08:43, 08:49, 08:57, 09:06, 09:12, 09:14, 09:18, 09:27, 09:45, 10:04, 10:13, 10:17, 10:19, 10:25, 10:33, 10:42, 10:48, 10:50, 10:54, 11:03, 11:21, 11:40, 11:49, 11:53, 11:55, 12:01, 12:09, 12:18, 12:24, 12:26, 12:30, 12:39, 12:57, 13:16, 13:25, 13:29, 13:31, 13:37, 13:45, 13:54, 14:00, 14:02, 14:06, 14:15, 14:33, 14:52, 15:01, 15:05, 15:07, 15:13, 15:21, 15:30, 15:36, 15:38, 15:42, 15:51, 16:09

TRX_NOR_Leeds_York_7	Leeds - Cross Gates - Garforth - East Garforth - Micklefield - Church Fenton - York - Church Fenton - Micklefield - East Garforth - Garforth - Cross Gates - Leeds - Cross Gates - Garforth - East Garforth - Micklefield - Church Fenton - York - Church Fenton - Micklefield - East Garforth - Garforth - Cross Gates - Leeds - Cross Gates - Garforth - East Garforth - Micklefield - Church Fenton - York - Church Fenton - Micklefield - East Garforth - Garforth - Cross Gates - Leeds	08:10, 08:18, 08:24, 08:26, 08:30, 08:39, 08:57, 09:04, 09:06, 09:10, 09:19, 09:37, 09:56, 10:05, 10:09, 10:11, 10:17, 10:25, 10:34, 10:40, 10:42, 10:46, 10:55, 11:13, 11:32, 11:41, 11:45, 11:47, 11:53, 12:01, 12:10, 12:16, 12:18, 12:22, 12:31, 12:49, 13:08, 13:17, 13:21, 13:23, 13:29, 13:37
TPX_Leeds_Huddersfield_9	Leeds - Cottingley - Morley - Batley - Dewsbury - Ravensthorpe - Mirfield - Deighton - Huddersfield - Deighton - Mirfield - Ravensthorpe - Dewsbury - Batley - Morley - Cottingley - Leeds - Cottingley - Morley - Batley - Dewsbury - Ravensthorpe - Mirfield - Deighton - Huddersfield - Deighton - Mirfield - Ravensthorpe - Dewsbury - Batley - Morley - Cottingley - Leeds - Cottingley - Morley - Batley - Dewsbury - Ravensthorpe - Mirfield - Deighton - Huddersfield	08:10, 08:15, 08:20, 08:26, 08:30, 08:34, 08:38, 08:44, 08:48, 08:53, 08:59, 09:03, 09:07, 09:11, 09:17, 09:22, 09:27, 09:33, 09:38, 09:44, 09:48, 09:52, 09:56, 10:02, 10:06, 10:11, 10:17, 10:21, 10:25, 10:29, 10:35, 10:40, 10:45, 10:51, 10:56, 11:02, 11:06, 11:10, 11:14, 11:20, 11:24, 11:30, 11:36, 11:40, 11:44, 11:48, 11:54, 11:58, 12:03, 12:09, 12:13, 12:17, 12:21, 12:27, 12:32, 12:37, 12:43, 12:48, 12:54, 12:58, 13:02, 13:06, 13:12, 13:16, 13:21, 13:27, 13:31, 13:35, 13:39, 13:45, 13:50, 13:55, 14:01, 14:06, 14:12, 14:16, 14:20, 14:24, 14:30, 14:34

TPX_Huddersfield_Leeds_9	Huddersfield - Deighton - Mirfield - Ravensthorpe - Dewsbury - Batley - Morley - Cottingley - Leeds - Cottingley - Morley - Batley - Dewsbury - Ravensthorpe - Mirfield - Deighton - Huddersfield - Deighton - Mirfield - Ravensthorpe - Dewsbury - Batley - Morley - Cottingley - Leeds - Cottingley - Morley - Batley - Dewsbury - Ravensthorpe - Mirfield - Deighton - Huddersfield - Deighton - Mirfield - Ravensthorpe - Dewsbury - Batley - Morley - Cottingley - Leeds	08:10, 08:14, 08:20, 08:24, 08:28, 08:32, 08:38, 08:43, 08:48, 08:54, 08:59, 09:05, 09:09, 09:13, 09:17, 09:23, 09:27, 09:32, 09:38, 09:42, 09:46, 09:50, 09:56, 10:01, 10:06, 10:12, 10:17, 10:23, 10:27, 10:31, 10:35, 10:41, 10:45, 10:50, 10:56, 11:00, 11:04, 11:08, 11:14, 11:19, 11:24, 11:31, 11:35, 11:39, 11:43, 11:49, 11:54, 11:59, 12:05, 12:10, 12:16, 12:20, 12:24, 12:28, 12:34, 12:38, 12:43, 12:49, 12:53, 12:57, 13:01, 13:07, 13:12, 13:17, 13:23, 13:28, 13:34, 13:38, 13:42, 13:46, 13:52, 13:56, 14:01, 14:07, 14:11, 14:15, 14:19, 14:25, 14:30, 14:35

A.4 ACO+Rescheduling with Learning Data

Table A.4: ACO+Rescheduling with Learning

Test #	Total Delay	Punctuality (%)
1	87	57.14
2	135	71.43
3	88	50.00
4	98	50.00
5	95	42.86
6	108	64.29
7	135	71.43
8	88	57.14
9	130	64.29
10	75	57.14
11	95	57.14
12	114	57.14
13	87	85.71
14	100	57.14
15	118	71.43
16	122	85.71

Test #	Total Delay	Punctuality (%)
17	125	71.43
18	83	71.43
19	99	64.29
20	94	78.57
21	117	57.14
22	75	64.29
23	103	50.00
24	83	35.71
25	72	42.86
26	69	71.43
27	72	42.86
28	115	64.29
29	86	64.29
30	72	71.43
31	85	71.43
32	76	50.00
33	92	50.00
34	98	57.14
35	102	57.14
36	73	64.29
37	97	57.14
38	130	42.86
39	111	71.43
40	70	50.00
41	93	64.29
42	132	71.43
43	97	92.86
44	75	64.29
45	63	71.43
46	67	57.14
47	129	71.43
48	124	64.29
49	99	71.43
50	102	57.14
Mean	**94.26**	**61.29**

Bibliography

[1] Office of Rail and Road, *Passenger rail performance 1 January to 31 March 2024*, Accessed: 2024-11-05, 2024. [Online]. Available: https://dataportal.orr.gov.uk/media/jwfpdpty/performance-stats-release-jan-mar-2024.pdf.

[2] Fredrik Monsuur, Marcus Enoch, Mohammed Quddus and Stuart Meek, 'Modelling the impact of rail delays on passenger satisfaction,' *Transportation Research Part A: Policy and Practice*, vol. 152, pp. 19–35, Oct. 2021. DOI: 10.1016/j.tra.2021.08.002.

[3] Manny Purewal Emma Stratton, *The effects of cancelled or delayed freight services*, Accessed: 2024-11-05, 2022. [Online]. Available: https://fishbonesolutions.co.uk/wp-content/uploads/2022/09/Effects-of-Cancelled-Trains-on-Freight-service-paper.pdf.

[4] *Digital railway*, https://www.networkrail.co.uk/running-the-railway/railway-upgrade-plan/digital-railway/, Accessed: 2024-12-3, Jan. 2019.

[5] Kah Yong Tiong and Carl-William Palmqvist, 'Quantitative methods for train delay propagation research,' *Transportation Research Procedia*, vol. 72, pp. 80–86, 2023. DOI: 10.1016/j.trpro.2023.11.326.

[6] Bo Fan, Clive Roberts and Paul Weston, 'A comparison of algorithms for minimising delay costs in disturbed railway traffic scenarios,' *Journal of Rail Transport Planning and Managment*, vol. 2, no. 1-2, pp. 23–33, Nov. 2012. DOI: 10.1016/j.jrtpm.2012.09.002.

[7] Abhishek Mishra, Neeraj Kumar and Seema Kharb, 'Priority based train scheduling method using ACO in indian railway perspective,' *IOP Conference Series:Materials Science and Engineering*, vol. 998, no. 1, p. 012 016, Dec. 2020. DOI: 10.1088/1757-899X/998/1/012016.

[8] Stephen Draper, *Definitions of railway performance metrics*, Oct. 2017. [Online]. Available: https://www.raildeliverygroup.com/media-centre-docman/acop/287-rdgntfdorpm-final/file.html.

[9] Richard Batley, Joyce Dargay and Mark Wardman, 'The impact of lateness and reliability on passenger rail demand,' *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 1, pp. 61–72, Jan. 2011. DOI: 10.1016/j.tre.2010.07.004.

[10] Didier M van de Velde, 'Learning from the japanese railways: Experience in the netherlands,' *Policy and Society*, vol. 32, no. 2, pp. 143–161, May 2013. DOI: 10.1016/j.polsoc.2013.05.003.

[11] Grace Mukunzi and Carl-William Palmqvist, 'The impact of railway incidents on train delays: A case of the swedish railway network,' *Journal of Rail Transport Planning and Managament*, vol. 30, no. 100445, p. 100 445, Jun. 2024. DOI: 10.1016/j.jrtpm.2024.100445.

[12] Network Rail, *Public performance measure and delay responsibility*, Accessed: 2024-11-05, 2024. [Online]. Available: https://www.networkrail.co.uk/who-we-are/how-we-work/performance/railway-performance/public-performance-measure-and-delay-responsibility/.

[13] Carl-William Palmqvist, Ingrid Johansson and Hans Sipilä, 'A method to separate primary and secondary train delays in past and future timetables using macroscopic simulation,' *Transportation Research Interdisciplinary Perspectives*, vol. 17, no. 100747, p. 100 747, Jan. 2023. DOI: 10.1016/j.trip.2022.100747.

[14] Lingyun Meng, Malik Muneeb Abid, Xinguo Jiang, Afaq Khattak and Muhammad Babar Khan, 'Increasing robustness by reallocating the margins in the timetable,' *Journal of Advanced Transport*, vol. 2019, no. 1, pp. 1–15, Jul. 2019. DOI: 10.1155/2019/1382394.

[15] Yafei Hou, Chao Wen, Ping Huang, Liping Fu and Chaozhe Jiang, 'Delay recovery model for high-speed trains with compressed train dwell time and running time,' *Railway Engineering Science*, vol. 28, no. 4, pp. 424–434, Dec. 2020. DOI: 10.1007/s40534-020-00225-8.

[16] Holger H Hoos and Thomas Stützle, 'SLS METHODS,' in *Stochastic Local Search*, Elsevier, 2005, pp. 61–112. DOI: 10.1016/b978-155860872-6/50019-6.

[17] Şebnem Yılmaz Balaman, 'Modeling and optimization approaches in design and management of biomass-based production chains,' in *Decision-Making for Biomass-Based Production Chains*, Elsevier, 2019, pp. 185–236. DOI: 10.1016/b978-0-12-814278-3.00007-8.

[18] D R Kiran, 'Critical path method,' in *Production Planning and Control*, Elsevier, 2019, pp. 457–471. DOI: 10.1016/b978-0-12-818364-9.00032-9.

[19] Chuntian Zhang, Yuan Gao, Valentina Cacchiani, Lixing Yang and Ziyou Gao, 'Train rescheduling for large-scale disruptions in a large-scale railway network,' *Transportation Research Part B: Methodological*, vol. 174, no. 102786, p. 102 786, Aug. 2023. DOI: 10.1016/j.trb.2023.102786.

[20] Xiaojuan Li, Yueying Huo, Zhenying Yan and Baoming Han, 'An optimization method for operation adjustment of high-speed delayed trains,' *Discrete Dynamics in Nature and Society*, vol. 2019, no. 1, pp. 1–16, May 2019. DOI: 10.1155/2019/9846970.

[21] Hiroyuki Goto, 'Introduction to max-plus algebra,' in *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ACM, Jul. 2014. DOI: 10.1145/2608628.2627496.

[22] Hasnain Roopawalla, *Ant colony optimization - hasnain roopawalla*, https://medium.com/@hasnain.roopawalla/ant-colony-optimization-1bbc346c2da5, Accessed: 2025-4-2, Apr. 2024.

[23] *Mathematical program solvers - IBM CPLEX*, https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer, Accessed: 2025-4-2, Nov. 2024.

[24] A Nash and D Huerlimann, 'Railroad simulation using opentrack,' in *WIT Transactions on The Built Environment*, vol. 74, WIT Press, May 2004. DOI: 10.2495/CR040051.

[25] *OpenTrack railway technology - railway simulation*, https://www.opentrack.ch/opentrack/opentrack_e/opentrack_e.html#Events, Accessed: 2025-2-16.

[26] *RailSys® suite*, https://rmcon-int.de/en/railsys-suite-en/, Accessed: 2025-2-24.

[27] Joern Pachl, 'Deadlock avoidance in railroad operations simulations,' Tech. Rep. 11-0175, 2011. [Online]. Available: https://trid.trb.org/View/1091281.

[28] J-P Bendfeldt, U Mohr and L Muller, 'Railsys, a system to plan future railway needs,' in *WIT Transactions on The Built Environment*, vol. 50, WIT Press, Aug. 2000. DOI: 10.2495/CR000241.

[29] Tomas Rosberg and Birgitta Thorslund, 'Simulated and real train driving in a lineside automatic train protection system environment,' *Journal of Rail Transport Planning and Managament*, vol. 16, no. 100205, p. 100 205, Dec. 2020. DOI: 10.1016/j.jrtpm.2020.100205.

[30] Norman Weik, Nora Niebel and Nils Nießen, 'Capacity analysis of railway lines in germany – a rigorous discussion of the queueing based approach,' *Journal of Rail Transport Planning and Management*, vol. 6, no. 2, pp. 99–115, Sep. 2016. DOI: 10.1016/j.jrtpm.2016.06.001.

[31] Ingrid Johansson, Carl-William Palmqvist, Hans Sipilä, Jennifer Warg and Markus Bohlin, 'Microscopic and macroscopic simulation of early freight train departures,' *Journal of Rail Transport Planning and Managment*, vol. 21, no. 100295, p. 100 295, Mar. 2022. DOI: 10.1016/j.jrtpm.2022.100295.

[32] *Transpennine route upgrade*, https://www.networkrail.co.uk/running-the-railway/railway-upgrade-plan/key-projects/transpennine-route-upgrade/, Accessed: 2025-2-16, Jul. 2021.

[33] *Railos*, https://wiki.railos.org/index.php/About, Accessed: 2025-2-16.

[34] *Python documentation*, https://www.python.org/doc/, Accessed: 2025-2-23.

[35] *Java documentation*, https://docs.oracle.com/en/java/, Accessed: 2025-2-23, Jul. 2018.

[36] *C++ documentation — DevDocs*, https://devdocs.io/cpp/, Accessed: 2025-2-23.

[37] *Trainline*, https://www.thetrainline.com/about-us, Accessed: 2025-3-9.