# Multi machine approaches for conflict resolution under moving block signalling

## M. L. Janssens

**Master thesis**
written in collaboration with TU Delft and ProRail
2022.MME.8609

TU Delft

# Multi machine approaches for conflict resolution under moving block signalling

by

## M. L. Janssens

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday February 23, 2022 at 10:00 AM.

| | | |
|---|---|---|
| Student number: | 4380738 | |
| Project duration: | April 12, 2021 – February 23, 2022 | |
| Thesis committee: | Prof. dr. R. R. Negenborn, | TU Delft, chair |
| | Dr. V. Reppa, | TU Delft, primary supervisor |
| | Dr. ir. E. Quaglietta, | TU Delft, secondary supervisor |
| | Prof. dr. R. M. P. Goverde, | TU Delft, independent member |
| | D. Middelkoop, | ProRail, company supervisor |

Cover photo [52]

# Preface

Dear reader,

Currently I am sitting in a chair in the library of Delft University of Technology. The same place, almost the same spot even, as I have been sitting in many times over the past years. However, I have completed an entire journey within that time. This journey started at the faculty of Industrial Design Engineering, where I developed an interest for innovations in technology. I changed direction when I moved to Mechanical Engineering, where I started my masters degree in Multi Machine Engineering.

During my years of studying MME, I became fascinated with intelligent systems related to logistics. The way people can increase efficiency, sustainability, and safety by using algorithms, control, or simulations is innovation at its finest to me. In the MME curriculum, these approaches are mainly directed towards the maritime sector. I became specialized in logistic solutions involving container carriers, cranes, and vessels. All of the courses and experiences provided me with good basis of knowledge in intelligent logistics. One of the last courses I took in MME was the “Railway Traffic Management” course, which I followed at the faculty of Civil Engineering. During this course, I realized that there are a lot of similarities in approaches to railway logistics and port logistics. I became highly interested in railway systems and I decided that I wanted to go in that direction for my graduation.

The completion of this thesis has been quite a journey on its own. Starting my internship at ProRail, I got many impressions of projects within the Innovation & Technological Development department. One of the highlights was definitely my visit to the simulation centre, where I was given the chance to be a train driver in one of their simulators.
The first few months of research went by rather smoothly. Some bumps in the road presented itself later on. First of all, graduating during a pandemic has not always been terrific fun. Second of all, I had never been a star programmer throughout my years at university, but nonetheless chose to develop quite some code for this thesis and write all of it myself. Despite many challenges during this process, I always kept pushing and trying. I am extremely proud of myself for everything that I have created, as well as the fact that I consider myself a much more experienced programmer now.

However, I did not do all of it alone, and there are many people I would like to thank for all their input and inspiration. First of all I want to thank my daily supervisors Egidio and Vasso. Egidio has helped me so much with his expertise on railway modelling, numerously explaining many aspects of a research field I was inexperienced but highly interested in. Vasso’s input has been very valuable in letting me see the bigger picture of the problem and not losing track of where I was going with my research as a whole. I want to thank my ProRail supervisor Dick for all his efforts to provide me with information, connect me to the right people and for showing the company’s view on my research. Also, I want to thank Maura for her very useful input on traffic management systems and spending time to meet with me multiple times. And of course thanks to my committee chair Rudy.
Additionally, there are many people who provided me with so much support during this journey. I want to thank Olger for sharing his many Python skills with me. I want to thank Roald for always being my sparring partner when I needed one and always believing in me. I want to thank my dear friends Eva, Lot, Anne-Marte, Sanne, and my parents, Marcel and Annette, for all the pep talks, motivation and necessary distraction during this time of my studies. It feels odd that my time as a student has now come to an end. I know for certain that I will look back on it as a marvelous time, where I have made many friends and have gotten lots of opportunities. But mostly, I will look back on it as a time where I have gained an enormous amount of knowledge, not just related to my studies, but also related to myself as a person. I hope you enjoy reading my thesis,

*M. L. Janssens*
*Delft, February 2022*

# Executive summary

**Trend**

The year 2021 has been declared the "year of rail" by the European Union (EU). Also, the International Energy Agency (IEA) has expressed the need to increase the role of railway transportation. Because of this focus shift towards railway transportation, development in this sector has been boosted to a higher urgency. This work contributes to innovation of the railway sector and has been written in collaboration with the Delft University of Technology and ProRail.

**Research opportunity**

An important aspect of railway innovation is the development of real time railway traffic management approaches, to perform the task of railway conflict resolution. This thesis considers track occupation conflicts, where two trains claim the same piece of infrastructure simultaneously. This type of conflict is mostly caused by an initial delay, resulting in the initial train schedule not being feasible anymore. Resolving these conflicts includes rescheduling trains, in order to obtain a new train schedule that is conflict free. These tasks are currently handled by dispatchers.

The need for improvement and innovation in real time railway traffic management has led the EU to develop the European Railway Traffic Management System (ERTMS), an overarching safety standard for all European countries, to be able to compute and support in conflict resolution actions based on real time information from the network. Where current implementations of ERTMS are in mostly in ERTMS level 1 or 2, deployed with traditional fixed block signalling systems, future railways can be equipped with ERTMS level 3, designed for railways with a moving block signalling system. The main difference between the two signalling systems is as follows. In a fixed block signalling system, railway networks are divided into sections of a fixed length. These sections are guarded by signals, and safety on the tracks is maintained by only allowing one train per section at a time. In a moving block signalling system, sections of a fixed length are no longer present and trackside signals are removed. Safety in the network is maintained by continuously measuring current train speed and position and thereby calculating a suitable speed profile that maintains a safe distance with respect to the preceding train.

However, before moving block signalling systems can be widely implemented, much more research needs to be conducted regarding this matter. In the current state of the art there are a lot of research gaps related to moving block signalling, one of which is the current absence of a suitable mathematical model for conflict resolution. This thesis focuses on conflict resolution for railway traffic management under a moving block signalling system and answers the following research question:

"How can centralized methods be designed to perform real time conflict resolution under moving block signaling?"

With the corresponding subquestions:

1. What is the current state of the art regarding centralized and non-centralized conflict resolution methods under moving block signalling?
2. What are the modelling characteristics of conflict resolution under moving block signalling?
3. How can centralized conflict resolution under moving block signalling be performed?
4. What are the impacts on conflict resolution performance for a centralized approach?
5. Which recommendations can be made to ProRail to support effective conflict resolution under moving block signalling?

The conflict resolution measures this thesis considers are reordering and retiming of trains. These measures have been modelled mathematically for fixed block systems in a state-of-the-art method called the alternative graph model. This model is based on a job shop scheduling problem, which is a formulation to schedule jobs to be performed on machines. In a railway approach to this scheduling problem, railway tracks are depicted as multiple machines acting in a system. This thesis extends the alternative graph method from fixed block to moving block suitability.
The state-the-art method models fixed block railway networks as a graph consisting of nodes and two types of arcs, namely fixed and alternative arcs. It is important to state that alternative arcs are able to describe reordering alternatives in the railway network. This thesis creates a graph for moving block networks by adding two new components to this mathematical approach. It should be noted that a distinction is made between moving- and fixed block sections. This means that important infrastructural components, namely switches and stations, are still viewed as fixed sections, but the rest of the network is viewed as a collection of moving sections related to trains rather than infrastructure. The first new component is the introduction of an extra node type, called a virtual node. Virtual nodes are related to the trains driving in the network, unlike the traditional static nodes that are related to infrastructure. The second new component is the introduction of an extra arc type, called the conditional arc. This new arc type has two functions in the modelling approach. First, the arc obeys to the reordering established by the alternative arcs. Second, the arc maintains the safe distance between two consecutive trains, in absence of trackside signals. Once the network has been modelled using this alternative graph approach, an optimization problem is formulated with the objective of minimizing the maximum propagated delay in the network.

Following model development, a proof of concept has been realised to verify the working principles of the novel conflict resolution approach. This involves a Python implementation of the mathematical approach, dependent on input information describing the conflicted railway situation. A commercial solver, Gurobi, is used to solve the optimization problem. This implementation is evaluated by the means of three illustrative examples, carefully formulated in collaboration with ProRail, to test different railway situations. Subsequently, the conflict resolution approach has been tested during a case study in the Rotterdam-The Hague Dutch railway corridor. A variety of delay scenarios is given to the considered network in different test cases. These delay scenarios consist of initial delays, thereby simulating track occupation conflicts in the network. This case study investigates the performance and behaviour of the conflict resolution model based on three main Key Performance Indicators. First, the number of affected trains is considered, split into the number of retimed trains and the number of reordered trains. Second, both the percentage and absolute propagated delay reduction at the final station are compared to an unrescheduled scenario. Third, the computation time of the conflict resolution system is assessed as well, since this is an important KPI for real time systems.

### Conclusion

The mathematical approach shows promising results in terms of applying the conflict resolution measures of retiming and reordering whilst still remaining safety. The proof of concept has shown that the novel aspects of the model are clearly distinguished and put to use, whilst respecting all constraints accordingly. The case study shows the model can solve conflicts between Rotterdam and The Hague within 10 seconds of computation time and reduces the propagated delay in all situations where delay is indeed propagated. These propagated delays can be reduced for 10-50% in the majority of cases. The beneficial effects of the model are strongest in cases where reordering measures are applied. Further research needs to be performed in order to further fine tune and optimize both this approach and moving block conflict resolution in general.

# List of Figures

1.1 Distribution of transportation greenhouse gas emissions in the European Union in 2019, by mode [54] . . . 2
1.2 The Operational Control Centre of Rail (OCCR) in Utrecht, The Netherlands, owned by the company ProRail [47] . . . 3
1.3 A visualization of a fixed block signalling system (simplified) . . . 4
1.4 A visualization of a moving block signalling system (simplified) . . . 4
1.5 A view on the railway network in the Netherlands [55] . . . 5
1.6 A view on the complexity of railway infrastructure [20] . . . 5
1.7 Visualization of thesis outline . . . 8

2.1 CBTC, extracted from [5] . . . 10
2.2 CBTC, extracted from [5] . . . 10
2.3 Example of a train speed profile in two cells (block sections), retrieved from [39] . . . 14
2.4 The principle of alternative graph formulation [16] . . . 15
2.5 An alternative graph model for multiple trains and sections [15] . . . 16
2.6 System architecture of separate areas and an overarching feasibility check, extracted from [11] . . . 20

3.1 The full information loop in ERTMS level 3 (for a moving block case) from [41] . . . 24
3.2 Conceptual visualization of information flows for a decision support system for traffic management. Note: the output information are possibilities, not mandatory or definite data 25
3.3 Illustration showing different detail levels in railway problem approaches, extracted from [22] . . . 26
3.4 Blocking times under fixed block signalling represented in a time distance diagram, extracted from [48] . . . 28
3.5 Blocking times under moving block signalling represented in a time distance diagram, extracted from [48] . . . 28
3.6 Conflicting blocks in a fixed block signalling system . . . 29
3.7 Conflicting blocks in a moving block signalling system . . . 29

4.1 A visual representation of the job shop scheduling problem for fixed block signalling systems . . . 35
4.2 A visual representation of the job shop scheduling problem for moving block signalling systems . . . 35
4.3 A simple visualization of representing train operations as a graph . . . 36
4.4 Alternative graph approach for two trains approaching a junction, extracted from [16] . . 36
4.5 Solution to the alternative graph in Figure 4.4 with train A preceding . . . 37
4.6 Solution to the alternative graph in Figure 4.4 with train B preceding . . . 37
4.7 Alternative graph formulation for three trains approaching junctions and a station . . . 37
4.8 The visual alternative graph formulation for two trains on a single unidirectional track under moving block signalling . . . 40
4.9 The visual alternative graph formulation for two trains on a converging track under moving block signalling . . . 40
4.10 The visual alternative graph formulation for two trains on a diverging-converging track under moving block signalling . . . 40
4.11 Station modelling in the alternative graph formulation . . . 41
4.12 Illustration of the headway between two trains in a moving block signalling system. . . . 43
4.13 A simple example of virtual node generation using speed and position measurements and train properties . . . 43

5.1 Conceptual representation of the rescheduling process . . . . . . . . . . . . . . . . . . 47
5.2 A visualization of the working principle of a barrier algorithm, extracted from [57] . . . . 51
5.3 A visualization of the working principle of the simplex algorithm, extracted from [7] . . . 51
5.4 Illustrative example 1: two trains on a converging track . . . . . . . . . . . . . . . . . . 54
5.5 The graph representation for illustrative example 1 if no rescheduling is applied . . . . . 56
5.6 The graph for illustrative example 1 before rescheduling . . . . . . . . . . . . . . . . . . 56
5.7 The graph representation for illustrative example 1 if rescheduling is applied . . . . . . . 56
5.8 Illustrative example 2: two trains passing two stations . . . . . . . . . . . . . . . . . . . 57
5.9 The graph representation for illustrative example 2 if no rescheduling is applied . . . . . 58
5.10 The graph representation for illustrative example 2 before rescheduling . . . . . . . . . 58
5.11 The graph representation for illustrative example 2 if rescheduling is applied . . . . . . . 58
5.12 Illustrative example 3: two trains passing a station and a third train crossing the route . 59
5.13 The graph representation for illustrative example 3 if no rescheduling is applied . . . . . 60
5.14 The graph representation for illustrative example 3 before rescheduling . . . . . . . . . 60
5.15 The graph representation for illustrative example 3 if rescheduling is applied . . . . . . . 60

6.1 Microscopic overview of the Rotterdam - The Hague corridor, extracted from the ProRail tool ”inframonitor” . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
6.2 Microscopic overview of platforms 8 and 9 at Rotterdam Central Station, extracted from the ProRail tool ”inframonitor”. Green line: intercity, Blue line: sprinter, Red line: coinciding routes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
6.3 Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . . . 72
6.4 Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . . . 72
6.5 Sensitivity of absolute delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . . . 73
6.6 Sensitivity of absolute delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . . . 73
6.7 Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . . . 74
6.8 Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . . . 74
6.9 Sensitivity of absolute delay reduction to initial delay, comparing test cases 1-4, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . . . 74
6.10 Sensitivity of absolute delay reduction to initial delay, comparing test cases 1-4, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . . . 74
6.11 Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . 75
6.12 Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . 75
6.13 Sensitivity of percentage delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . 75
6.14 Sensitivity of percentage delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . 75
6.15 Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . 76
6.16 Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . 76
6.17 Sensitivity of percentage delay reduction to initial delay, comparing test cases 1-4, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . . . 76
6.18 Sensitivity of percentage delay reduction to input delay, comparing test cases 1-4, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . . . 76
6.19 Sensitivity of computation time to initial delay, comparing test cases 1-4, with delay scenarios visualized in seconds . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77

6.20 Sensitivity of computation time to initial delay, comparing test cases 1-4, with delay scenarios visualized in step relative to planned headway . . . . . . . . . . . . . . . . . . . . . 77

A.1 Enlarged plot of illustrative example 1 without rescheduling . . . . . . . . . . . . . . . . . 85
A.2 Enlarged plot of illustrative example 1 before rescheduling . . . . . . . . . . . . . . . . . . 86
A.3 Enlarged plot of illustrative example 1 with rescheduling . . . . . . . . . . . . . . . . . . . 86
A.4 Enlarged plot of illustrative example 2 without rescheduling . . . . . . . . . . . . . . . . . 87
A.5 Enlarged plot of illustrative example 2 before rescheduling . . . . . . . . . . . . . . . . . . 88
A.6 Enlarged plot of illustrative example 2 with rescheduling . . . . . . . . . . . . . . . . . . . 88
A.7 Enlarged plot of illustrative example 3 without rescheduling . . . . . . . . . . . . . . . . . 89
A.8 Enlarged plot of illustrative example 3 before rescheduling . . . . . . . . . . . . . . . . . . 89
A.9 Enlarged plot of illustrative example 3 with rescheduling . . . . . . . . . . . . . . . . . . . 90

# List of Tables

2.1 Published literature in the research field of moving block signalling . . . . . . . . . . . . 10
2.2 Published literature in the research field of real time railway traffic management . . . . . 13
2.3 Published literature in the research field railway traffic management using distributed methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
2.4 State of the art and overview and research gap, to be filled by this thesis . . . . . . . . . 21

3.1 The characteristic differences between fixed and moving block signalling systems . . . 23
3.2 Possible KPI's to evaluate a conflict resolution method . . . . . . . . . . . . . . . . . . . . 29
3.3 Selected KPI's to evaluate a conflict resolution method . . . . . . . . . . . . . . . . . . . . 30

4.1 Characteristics of two possible modelling approaches . . . . . . . . . . . . . . . . . . . . 33
4.2 Overview of job shop representations for scheduling problems . . . . . . . . . . . . . . 34

5.1 Input parameters to create an implementation of the mathematical model . . . . . . . . 48
5.2 Table: *Note: these node and arc objects represent all operations and operation processing times of corresponding train . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
5.3 Characteristics of the two main algorithms provided by Gurobi optimizer . . . . . . . . . 51
5.4 Input parameters for illustrative example 1 . . . . . . . . . . . . . . . . . . . . . . . . . . 55
5.5 Illustrative example 1 - scheduled situation . . . . . . . . . . . . . . . . . . . . . . . . . . 55
5.6 Illustrative example 1 - conflicted situation . . . . . . . . . . . . . . . . . . . . . . . . . . 55
5.7 Illustrative example 1 - without rescheduling . . . . . . . . . . . . . . . . . . . . . . . . . 56
5.8 Illustrative example 1 - with rescheduling . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
5.9 The total propagated delay in illustrative example 1 in different situations . . . . . . . . 56
5.10 Input parameters for illustrative example 2 . . . . . . . . . . . . . . . . . . . . . . . . . . 57
5.11 Illustrative example 2 - scheduled situation . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.12 Illustrative example 2 - conflicted situation . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.13 Illustrative example 2 - without rescheduling . . . . . . . . . . . . . . . . . . . . . . . . . 59
5.14 Illustrative example 2 - with rescheduling . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
5.15 The total propagated delay in illustrative example 2 in different situations . . . . . . . . 59
5.16 Illustrated example 3 - scheduled situation . . . . . . . . . . . . . . . . . . . . . . . . . . 61
5.17 Illustrated example 3 - conflicted situation . . . . . . . . . . . . . . . . . . . . . . . . . . 61
5.18 Illustrated example 3 - unrescheduled . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
5.19 Illustrated example 3 - rescheduled . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
5.20 The total propagated delay in illustrative example 3 in different situations . . . . . . . . 61
5.21 Test cases for verification study and their expected outcome . . . . . . . . . . . . . . . . 62
5.22 Verification study, applied to illustrative example 1 . . . . . . . . . . . . . . . . . . . . . 62
5.23 Verification study, applied to illustrative example 2 . . . . . . . . . . . . . . . . . . . . . 63
5.24 Verification study, applied to illustrative example 3 . . . . . . . . . . . . . . . . . . . . . 63

6.1 Routes to The Hague of trains leaving within one hourly cycle from Rotterdam platforms 8 and 9 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
6.2 A few example rows of the final dataset for train A2400 . . . . . . . . . . . . . . . . . . . 68
6.3 The different inputs to be imposed on the network for the case study . . . . . . . . . . . 70
6.4 Result summary per KPI of all corresponding test cases as described in Table 6.3 *Negligible (< 30 seconds) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

# Contents

**1 Introduction 1**
- 1.1 Background . . . 1
- 1.2 Problem statement . . . 5
  - 1.2.1 Objective and contribution . . . 5
  - 1.2.2 Research questions . . . 6
  - 1.2.3 Thesis structure and approach. . . . 6

**2 State of the art 9**
- 2.1 Moving block signalling. . . . 9
  - 2.1.1 Application fields . . . 10
  - 2.1.2 Motivations for using moving block signalling . . . 11
  - 2.1.3 Problem types and objectives . . . 11
- 2.2 Real time railway traffic management . . . 11
  - 2.2.1 Problem assumptions . . . 12
  - 2.2.2 Problem types and objectives . . . 14
  - 2.2.3 Problem formulations. . . . 15
  - 2.2.4 Problem solution methods . . . 16
- 2.3 Non centralized strategies in railway traffic management . . . 17
  - 2.3.1 Non-centralized problem types . . . 18
  - 2.3.2 Non-centralized approaches . . . 18
  - 2.3.3 Non-centralized problem solution methods . . . 20
- 2.4 Conclusion . . . 20

**3 Conflict resolution under moving block signalling 23**
- 3.1 The potential of moving block signalling systems. . . . 23
- 3.2 Information flows in a moving block signalling system . . . 24
- 3.3 Considerations for conflict resolution in a moving block signalling system . . . 26
- 3.4 Key performance indicators for conflict resolution under moving block signalling. . . . 29
  - 3.4.1 Relevant key performance indicators (KPI's) . . . 29
  - 3.4.2 KPI selection . . . 30
  - 3.4.3 Performance of state of the art methods . . . 31
- 3.5 Conclusion . . . 31

**4 Definition of a centralized conflict resolution model for moving block rail operations 33**
- 4.1 Conflict resolution methods . . . 33
- 4.2 The job shop scheduling problem . . . 34
- 4.3 The alternative graph model . . . 35
  - 4.3.1 The principle of alternative graph theory . . . 35
  - 4.3.2 Conflict resolution and delay minimization with alternative graph theory . . . 37
- 4.4 Alternative graph theory for a moving block signalling system . . . 38
  - 4.4.1 Model components . . . 39
  - 4.4.2 Station modelling . . . 41
  - 4.4.3 Arc weights . . . 41
  - 4.4.4 Generation of virtual nodes . . . 43
  - 4.4.5 Modelling assumptions. . . . 43
- 4.5 Problem formulation . . . 44
- 4.6 Conclusion . . . 46

**5 Verification of a centralized conflict resolution model under moving block signalling 47**
5.1 Generating an alternative graph . . . 47
5.1.1 Nodes . . . 48
5.1.2 Arcs . . . 48
5.1.3 Trains . . . 49
5.1.4 Graph . . . 49
5.2 Initializing the model . . . 50
5.3 Solution approaches . . . 50
5.4 Solving the optimization problem with a commercial solver: Gurobi optimizer . . . 50
5.4.1 Gurobi solution algorithms . . . 51
5.4.2 Model tailoring . . . 51
5.5 Illustrative examples . . . 54
5.5.1 Two trains on a converging track . . . 54
5.5.2 Two trains passing two stations . . . 57
5.5.3 Two trains passing a station, a third train crossing the route . . . 59
5.6 Model verification . . . 61
5.7 Conclusion . . . 63

**6 Model validation and assessment on the Rotterdam-The Hague Dutch corridor 65**
6.1 Rotterdam - The Hague corridor . . . 65
6.2 Obtaining simulated data . . . 66
6.2.1 Prorail simulation tools . . . 67
6.2.2 Data set construction . . . 67
6.3 Creating test cases for conflict resolution . . . 69
6.4 Conflict resolution results . . . 70
6.4.1 Result overview . . . 70
6.4.2 Sensitivity analysis of absolute reduction of propagated delay to different test cases . . . 72
6.4.3 Sensitivity analysis of percentage reduction of propagated delay to different test cases . . . 75
6.4.4 Sensitivity analysis of computation time to different test cases . . . 77
6.5 Conclusion . . . 78

**7 Conclusions and recommendations 79**
7.1 Addressing research subquestions . . . 79
7.1.1 Subquestion 1: What is the current state of the art regarding centralized and non-centralized conflict resolution methods under moving block signalling? . . . 79
7.1.2 Subquestion 2: What are the modelling characteristics of conflict resolution under moving block signalling? . . . 79
7.1.3 Subquestion 3: How can centralized conflict resolution under moving block signaling be performed? . . . 80
7.1.4 Subquestion 4: What are the impacts on conflict resolution performance for a centralized approach? . . . 80
7.2 Addressing main research question: How can centralized methods be designed to perform real time conflict resolution under moving block signaling?” . . . 80
7.3 Recommendations for further research . . . 81
7.3.1 Recommendations following from scope boundaries and assumptions . . . 81
7.3.2 Recommendations regarding the mathematical approach . . . 81
7.3.3 Recommendations for non-centralized approaches . . . 82
7.3.4 Other recommendations . . . 82

**Appendices 85**

**A Enlarged plots 85**
A.1 Illustrative example 1 . . . 85
A.2 Illustrative example 2 . . . 87
A.3 Illustrative example 3 . . . 89

**B Additional information on trains and data in Rtd-Gvc corridor 91**
B.1 Additional train information. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
B.2 Additional dataset information . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91

**C Detailed case study results 93**
C.1 Situation 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
C.1.1 Input step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
C.1.2 Input step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
C.1.3 Input step 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
C.2 Situation 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
C.2.1 Input step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
C.2.2 Input step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
C.2.3 Input step 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
C.3 Situation 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97
C.3.1 Input step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97
C.3.2 Input step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
C.3.3 Input step 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
C.4 Situation 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
C.4.1 Input step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
C.4.2 Input step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .100
C.4.3 Input step 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .100

**D Python functions 103**
D.1 Function to initialize node values . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .103
D.2 Function to determine minimum headways . . . . . . . . . . . . . . . . . . . . . . . . . .103
D.3 Function to create alternative arcs. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .105
D.4 Function to establish conditional arc obedience . . . . . . . . . . . . . . . . . . . . . . . .105

# 1

# Introduction

## 1.1. Background

Railway transport is a popular means of public transportation and has been under development for almost 200 years. From the first steam passenger train by Stephenson in 1829 to international railway lines across Europe, for example the Amsterdam-Vienna connection released in June 2021. Using rails as a transportation mode for both passengers and freight is recognized as one of the most sustainable ways of travel to this day. The sustainability of railway transport expresses itself in multiple aspects, including the fact trains can transport large groups of people with a low rate of energy use per person [26]. In the European Union, railway transport is responsible for less than 0.5% of all green house gas emissions [9]. In comparison, road transport is accountable for more than 70% of transport related emissions [54], see also Figure 1.1. Additionally, trains are more environmentally friendly in terms of noise levels, use of space, and the ability to apply regenerative braking to save energy [26, 51].

The International Energy Agency (IEA) has expressed the need to increase the role of railway transportation for both freight and passengers. First, because of the aforementioned environmentally friendly nature. Second, because of the general increase in transport demand that followed from population growth and increasing globalization and urbanisation [1]. In 2021, the European Union declared that year to be the ”year of rail”, throughout which the benefits of sustainable, safe, and smart transportation would be continuously highlighted [9]. The European year of rail is part of the EU’s smart and sustainable mobility strategy, which includes several rail related milestones. The EU wants to double high speed railway traffic by 2030, and additionally deploy automated traffic at large scale. By the year 2050, freight transport will be doubled as well. The European Green Deal states that in this same year, transport emissions should by reduced by 90% [9].

Because of this shift towards increased use of railway transport, railway development has been boosted to a higher urgency. In relation to this, a lot of new infrastructure projects are being executed and the number of rails, tracks, and stations are being significantly increased [1]. However, building new infrastructure is very time intensive and therefore other solutions to satisfy the growing need for railway transport are being investigated [8]. This calls for innovations directed towards the increase of railway capacity and efficiency with the current infrastructural resources. Part of these innovations are directed towards smart solutions in railway traffic management.

Railway traffic management is handled by railway dispatchers, that ensure safe and efficient operations throughout the railway network. Figure 1.2 shows dispatchers at work in the Operational Control Centre of Rail (OCCR), the national train dispatching post in the Netherlands owned by the company ProRail. Dispatchers closely monitor all activities in the railway network and see to the avoidance of conflicts or the mitigation of conflict effects. In order to decide which actions need to be taken, dispatchers currently rely on predefined action plans. These action plans describe a set of basic rules for frequently occurring situations on the railway tracks, for example delayed sprinters during morning rush hour. In addition to these plans, or in situations where these plans are invalid, dispatchers use

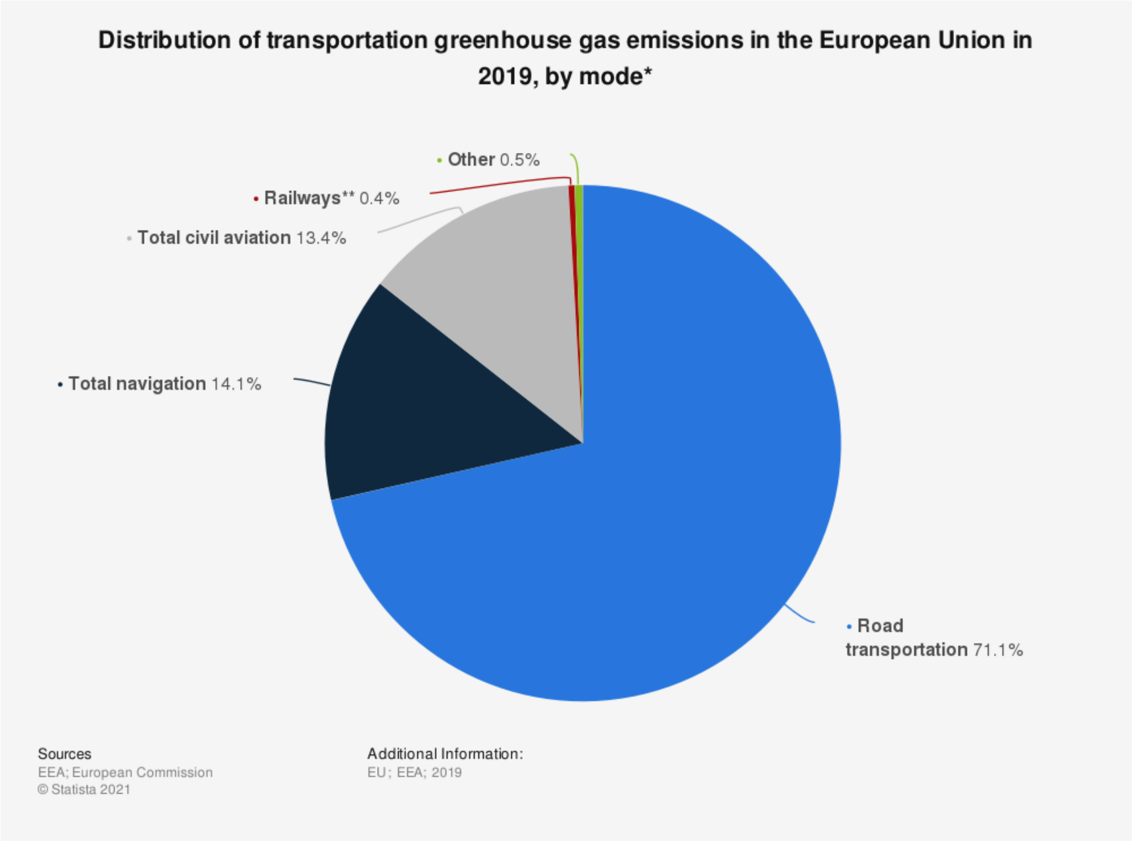


Figure 1.1: Distribution of transportation greenhouse gas emissions in the European Union in 2019, by mode [54]

their own experience and expertise to decide on a suitable solution for a conflict.

Conflicts are unavoidable in railway transport, since railway networks are dynamic systems that are subjected to uncertainties at all times. Railway conflicts can be related to crew scheduling, passenger connections, rolling stock and track occupation [49]. A railway track occupation conflict occurs when two trains claim the same infrastructural resource simultaneously [11]. In this work, this is the only type of railway conflict considered, thus all references to conflicts are of this type. The motivation for accurate conflict management is the fact that trains drive according to a strict timetable, and conflicts may result in the original timetable not being feasible anymore. The application and implementation of conflict resolution actions can regain feasibility in the railway network, reestablishing the efficiency of railway operations. Conflict resolution actions are mainly divided into five possibilities [49]:

- **Retiming** is the action of altering the points in time where a train should pass a certain point in infrastructure (mostly stations)
- **Reordering** is the action of changing the order in which two or more trains pass infrastructure points, for example a switch
- **Rerouting** is the action of changing the route of a train
- **Cancelling** is the action of taking a train operation out of the timetable completely
- **Short turning** is the action of leaving the station in the direction the train had arrived from, instead of pursuing its original route

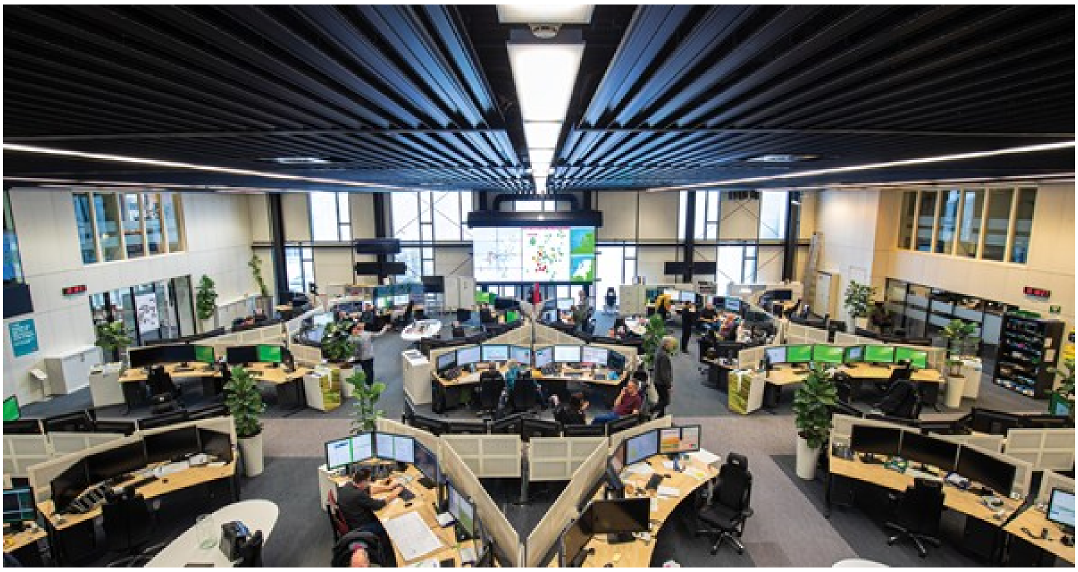


Figure 1.2: The Operational Control Centre of Rail (OCCR) in Utrecht, The Netherlands, owned by the company ProRail [47]

Conflicts can cause undesired consequences, mostly in the form of delays. One of the ways to, possibly intelligently, increase railway efficiency is by creating ways to predict conflicts, resolve conflicts, and minimize the consequences of conflicts. Basing these decisions on predefined plans and human expertise, like the dispatchers in Figure 1.2, may not always result in the best solution. This is because not all contributing factors are taken into account when using a predefined plan and because humans can not always oversee the entire network and all the corresponding consequences of decisions. New strategies are being developed, where dispatchers make use of real time data describing the current situation. This results in decisions not having to be solely based on expertise or experience, since the dispatcher can be aided by a decision support system. Decision support systems are, as defined in [21]: ”computer-based interactive systems that support decision makers rather than replace them, utilising data and models with varying degrees of structure”. Decision support systems can add great value to real time railway traffic management. In the past years, scientific research has been performed concerning the design of decision support systems for real time railway traffic management. This can vary from real time implementations to increase network efficiency [13], computational studies using optimization tools [18], or approaches for real time train control [14]. Within current research, the actions of conflict detection, resolution, solution implementation, and train control are often interrelated. However, this work will focus on conflict resolution, as will be discussed more elaborately in the problem statement.

An important property to be taken into account when designing conflict resolution systems is the safety and signalling system of the railway network. The European Union is working on an overarching safety standard for all European countries, the European Railway Traffic Management System (ERTMS), of which three different levels have been designed. In ERTMS level 1 and 2, rails will be equipped with a fixed block signalling system. This is the signalling system that current railway tracks are traditionally equipped with, where the railway network is divided into track sections (blocks). All track sections are guarded by a signal, that can either give a red, yellow or green light [13]. Every block can be occupied by one train at a time, causing the occupied section to be guarded by a red signal whilst a train is running through that section. The preceding track section is then equipped with a yellow signal, to warn the approaching train that the next section is occupied. Sections that are free to enter are indicated with a green signal. In this way, trains are always safely separated and warned in time for potential conflict. A visualization of this signalling system is given in Figure 1.3. In a fixed block signalling system, the most desired operational situation occurs when a train only encounters green signals, and thus does not have to slow down or stop unexpectedly. This is called the green wave principle [13].

In the newly developed ERTMS level 3, railway systems will be equipped with a moving block signalling system. In this system, track side signals are removed and replaced by on board monitoring systems on trains. The on board monitoring system consists of sensors, software, and the ability to exchange information, which makes it possible to continuously measure the train's current speed and position [25]. This information is then sent to a traffic management system (TMS) and is processed. The traffic management system will calculate the desired speed profile for the train to follow, ensuring trains drive at a safe distance from each other. In other words, providing the train with it's current Movement Authority (MA). Traffic management control centres (like the OCCR) are on continuous dialogue with trains and can thus calculate and dispatch to each train (in safe mode) the current movement authority info [25]. A visualization of this signalling system is given in Figure 1.4. The moving block signalling system comes with multiple benefits. The most prominent benefit is that headways can be significantly reduced because of the absence of fixed blocks. This means railway tracks are no longer designed for trains with the worst braking performance, and trains can have a tailored MA to their braking properties. Trains no longer have to wait for a full block length to be cleared before they can enter the corresponding infrastructure, allowing trains to drive much closer together [5, 27, 37, 60]. Consequently, this can lead to increased railway capacity and higher management efficiency [25, 46, 56]. In addition, moving block signalling systems can provide possibilities for higher operating speeds [60], reduced transit times and even more reduced energy consumption due to the expected decrease in unexpected stops or slowdowns [25]. A lot of research on moving block signalling systems is yet to be performed, which is what this work will be part of.

Figure 1.3: A visualization of a fixed block signalling system (simplified)

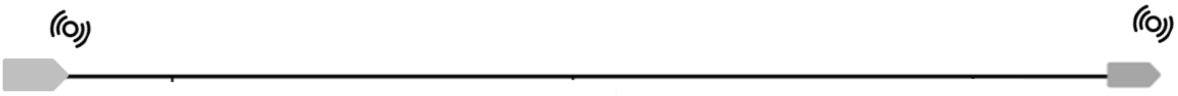


Figure 1.4: A visualization of a moving block signalling system (simplified)

This thesis will focus on conflict resolution in railway traffic management under a moving block signalling system. For the case of track occupation conflicts, this means obtaining a new train driving schedule, in which trains can drive conflict free. This focus means that conflict detection, train control and design for fixed block tracks are out of the thesis scope. However, fixed block signalling can be used to explain concepts or methods also applicable for moving block signalling.

Conflict resolution in railway networks is a challenging task, that is partially due to some important railway network properties. Railway networks are large scale, complex, dynamic systems that are subjected to uncertainties at all times. Figure 1.5 shows a part of the Dutch railway network, indicating the large scale complexity of this transport mode. Figure 1.6 shows trains approaching multiple switches, showing that railways are complex not only on macroscopic (coarse), but also on microscopic (detailed) scale. A moving block signalling system contributes to the overall complexity, since real time data are inevitably involved, which can lead to difficulties in large scale systems in terms of computational burden, high problem complexity [38, 61], or solution quality [12]. In conflict resolution, both data communication and quick solutions are important, especially with regard to the real time aspect. For these reasons, it can be valuable to not only investigate centralized, but also non-centralized approaches for conflict resolution. In centralized approaches, the whole problem and its corresponding data is regarded as a whole, whilst a non-centralized approach requires system decomposition into several subsystems. Non-centralized traffic management can be beneficial in complex real time operations for multiple reasons. First, because it can significantly reduce computation times of decision support systems [38, 10, 32]. Second, because it can prevent fault propagation or error enhancement through the entire network [53]. Additionally, it can be suitable for networks owned by different organizations and can also have practical reasons in terms of sensor and software placement. Despite these prospective benefits, this thesis will only shortly discuss non-centralized traffic management in current literature, but this will not be taken further throughout the rest of this work. This is due to the fact that no mathematical model for conflict resolution under moving block signalling yet exists. This must first be available before moving towards extended approaches, which is why the focus will lie on the centralized approach first.

Figure 1.5: A view on the railway network in the Netherlands [55]

Figure 1.6: A view on the complexity of railway infrastructure [20]

This thesis has been performed in collaboration with ProRail and Delft University of Technology (TU Delft). ProRail is the owner and manager of the railway network in the Netherlands. ProRail is always aiming at renewing and improving Dutch railway transport and recognizes need to innovate in order to prepare Dutch railways for the future. Collaborations between ProRail and TU Delft have been successful multiple times, and this thesis will contribute to railway research leading towards a more sustainable, innovative and efficient future for rail transportation. Research on conflict resolution under moving block signalling can give ProRail and TU Delft new insights in railway traffic management. This topic and its contribution will be further elaborated in the next section.

## 1.2. Problem statement

As extensively described in the introduction, railways are expected to play an increasingly large role in European transportation. In order to deal with a growing demand for railway transport, railways can either expand infrastructure or make more efficient and intelligent use of its existing assets. In the latter category, one of the ways to increase railway efficiency is to apply intelligent decision support for conflict resolution. The type of railway signalling system can play an important role in the development of conflict resolution approaches. Since railway networks are large scale and complex systems, designing efficient real time decision support systems can be a challenging task.

### 1.2.1. Objective and contribution

The objective of this thesis is to investigate how conflict resolution can be performed under a moving block signaling system. This includes developing a mathematical model for conflict resolution and investigating whether and how this new model can be applied in practice. Note that this thesis does not cover speed profile generation or train control.

The contribution of this work is as follows. This thesis will provide an extensive and theoretically supported mathematical model for conflict resolution in a fully moving block railway system, which has not been published before. It should be noted that this is done whilst keeping in mind that a centralized method is first being designed, and that non-centralized options could be beneficial in future research. Therefore, it is mentioned explicitly that this work considers a centralized approach. The main research question of this thesis will be:

*”How can centralized methods be designed to perform real time conflict resolution under moving block signalling?”*

### 1.2.2. Research questions

The research question stated above can be divided into several subquestions, in order to provide a thorough answer to the main research objective. These subquestions are as follows:

1. What is the current state of the art regarding centralized and non-centralized conflict resolution methods under moving block signalling?
    (a) To what extent has moving block signalling been researched in current literature?
    (b) What are current strategies for conflict resolution in railway traffic management?
    (c) How have non-centralized methods been applied in the railway sector?
2. What are the modelling characteristics of conflict resolution under moving block signalling?
    (a) What are the characteristics of moving block signalling systems?
    (b) What are relevant assumptions and considerations?
    (c) How can the performance of a mathematical model for conflict resolution be indicated?
    (d) What is the performance of state-of-the-art solution methods?
    (e) How can a mathematical model represent railway conflict resolution under moving block signalling?
3. How can centralized conflict resolution under moving block signaling be performed?
    (a) How can the mathematical model be implemented?
    (b) What is a suitable centralized solution method?
    (c) How can this solution method be verified?
4. What are the impacts on conflict resolution performance for a centralized approach?
    (a) How can the performance of centralized solution methods be investigated?
    (b) What is the performance of the centralized solution approach?
    (c) How does the performance of the designed centralized solution approach compare to the performance of state-of-the-art approaches?
5. Which recommendations can be made to ProRail to support effective conflict resolution under moving block signalling?

### 1.2.3. Thesis structure and approach

The outline of this thesis is as follows. First, in Chapter 2, the current state of the art is analyzed and described. Throughout this literature review, focus will lie on three main aspects regarding subquestions 1a-c respectively. Literature tables will be used to provide clear overviews of the current state of the art, thereby answering subquestion 1.

In Chapter 3, the principles and characteristics of moving block signalling will be investigated, with the goal of finding relevant considerations and assumptions. In addition, relevant Key Performance Indicators (KPI’s) will be analysed and selected. This will answer subquestions 2a-c. Some short attention will be devoted to the performance of state-of-the-art methods, answering subquestion 2d. All these aspects will be taken into account when moving towards the next chapter.

In Chapter 4, a mathematical model will be formulated to represent railway networks under moving block signalling. The mathematical formulation will be inspired by both state of the art methods and

ProRail input. This is followed by an optimization problem to perform conflict resolution, thus answering research question 2e.

In Chapter 5, the mathematical model for solving the conflict resolution problem in a centralized manner will be verified as a proof of concept. This will be done by implementing the mathematical approach into a Python model and, after finding an appropriate solution method, verification will be performed by constructing illustrative examples. Research subquestions 3a-c will thus be answered in this chapter.

In Chapter 6, the developed method will be further evalutated and tested on Key Performance Indicators by executing a case study in the Rotterdam-The Hague Dutch corridor. In order to properly investigate the performance of the model, a selection of different test cases will be used. This will answer research questions 4a-c.

This thesis will finish with a conclusion and several recommendations for further research, which will answer subquestions 5 and the main research question of this work. A visual overview of all the chapters and the corresponding research questions can be found in Figure 1.7.

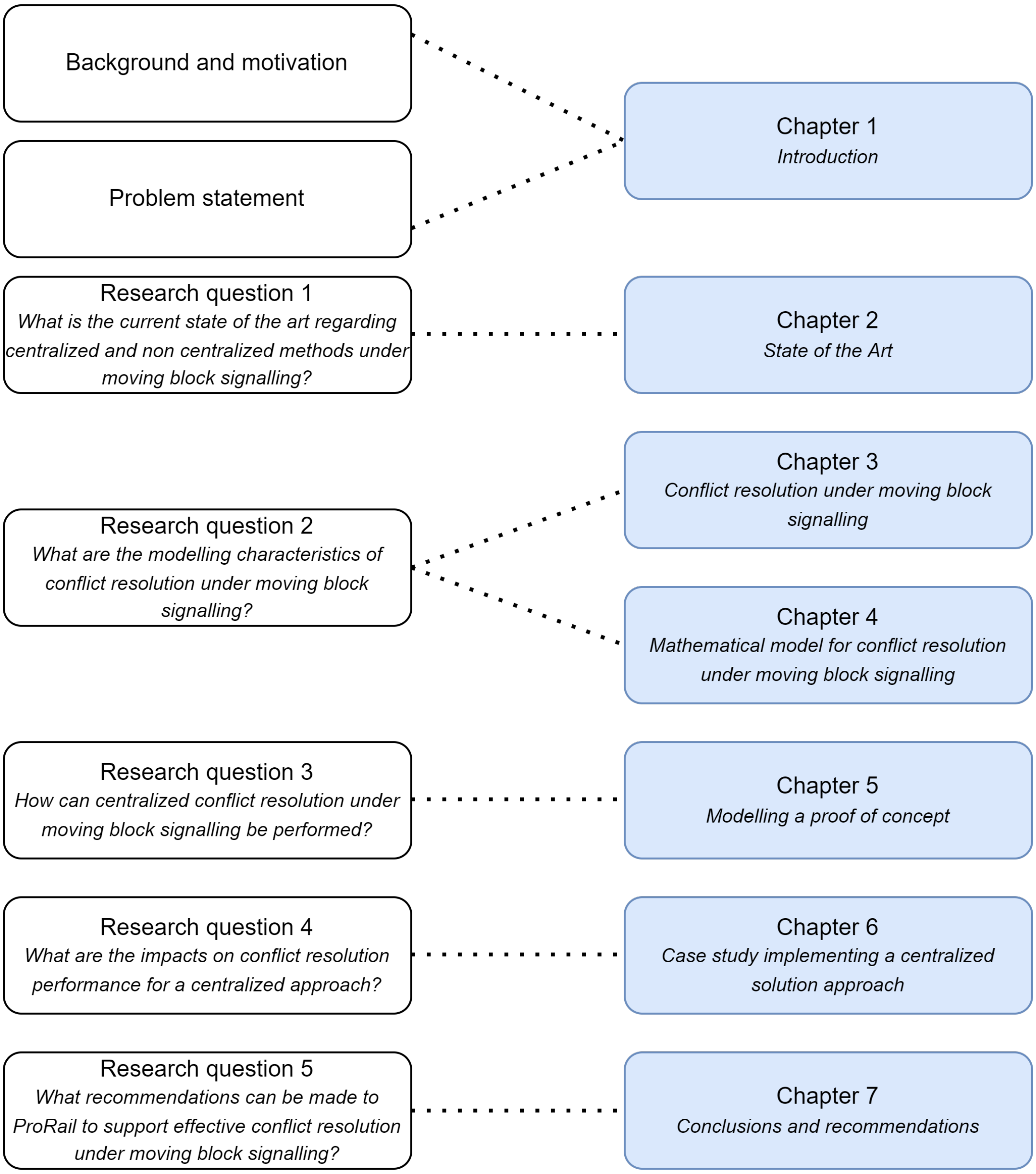


Figure 1.7: Visualization of thesis outline

# 2

# State of the art

This chapter contains a review of current literature, related to the topic of this thesis. Several papers will be discussed and analysed with regard to different aspects, thereby exposing research gaps and providing proof for the scientific contribution of this thesis. This chapter is structured as follows. First, some papers addressing moving block signalling systems will be presented and analysed. Second, the focus will shift to papers concerning the topic of real time railway traffic management, elaborating on current practices, strategies and methods. Third, the focus will lie upon papers discussing non-centralized applications in the railway sector, exposing different approaches for non-centralized optimization. Altogether, this will reveal the state of the art regarding multi machine approaches for real time railway traffic management under moving block signalling, meaning this chapter provides the answer to research question 1. This chapter will close with a short conclusion and identify the gaps found in current literature.

## 2.1. Moving block signalling

This section will discuss literature that concerns moving block signalling systems. It will provide an overview of different applications and objectives through an analysis of scientific papers. The concept of moving block signalling has been proposed a few decades ago [31], with the main benefit of reducing headway times between trains and thereby increasing railway capacity.

Because implementation of moving block signalling in actual large scale railway networks is still far in the future, research that explicitly focuses on moving block signalling systems is rather scarce, although not completely absent. This section will present research that has been done within this field.

Table 2.1 presents current literature that investigates moving block signalling systems. Some important and useful aspects of these papers are expressed according to four categories, as described below.

- **Application field** Describes in what application field the research has been conducted. Although this thesis concerns railway networks, other application fields for moving block signalling could be found in literature.
- **Moving Block motivation** Describes what motivations to apply a moving block signalling systems are mentioned.
- **Problem type** Indicates what type of problem the paper is focusing on.
- **Objective** Indicates the objective of the paper.

Table 2.1: Published literature in the research field of moving block signalling

| Reference | Application | MB Motivation | Problem type | Objective |
|---|---|---|---|---|
| Carvajal 2016 [5] | Metro lines | Headway reduction | Distance control | Increase energy efficiency |
| Gu 2013 [27] | Railway | Headway reduction | Peak power problem | Peak demand reduction<br>Energy consumption reduction |
| Koning 2002 [37] | Railway | Increase capacity<br>Headway reduction | Performance comparison | Find performance differences |
| Mazzanti 2018 [40] | Metro lines | Safety ensurance | Scheduling | Deadlock avoidance |
| Pochet 2016 [46] | Suburban railways | Increase management efficiency<br>Increase railway capacity<br>Headway reduction | Rescheduling | Restore feasibility |
| Wang 2014 [56] | Railway | Increase capacity | Trajectory planning | Reduce energy consumption |
| Zafar 2012 [60] | Railway | Headway reduction<br>Higher operating speed<br>Higher operating frequency | Interlocking model | Safety<br>Cost-effectiveness<br>Efficiency |
| Giuliari 2000 [25] | Railway | Increase capacity<br>Reduce transit times<br>Reduce energy consumption | conflict detection<br>& resolution | improved operations |
| Giannettoni 2004 [24] | Railway | | | |

### 2.1.1. Application fields

The concept of moving block signalling has been investigated by the European Union for applications in the railway network, for example in the COMBINE project [25]. However, the moving block principle has also been applied in several subway and metro line systems [5, 40, 46]. The application field of metro lines is very similar to railway networks, but differences can be found mainly in the fact that railway networks are larger in size and complexity. In subway lines, the moving block principle has been widely developed with regard to train control [46]. It should be noted that within the application of metro and subway lines, this signalling system is referred to as Communication Based Train Control (CBTC) [5, 46, 40]. This implies that the focus is slightly more on the train control system, rather than the traffic management system.

In [5], it is explained that CBTC has its basic guidelines, but that each manufacturer can implement its own architecture and modify the system to their own needs and constraints. CBTC can be used for both moving block and virtual coupling systems, meaning it can either apply an automatic train supervision system (ATS) or fully autonomous train operations (ATO) [5]. Figures 2.1 and 2.2 show a visualization of train control under CBTC in an ATS (2.1) or ATO (2.2) architecture, showing the limits of authority, braking curves, and information flows [5].

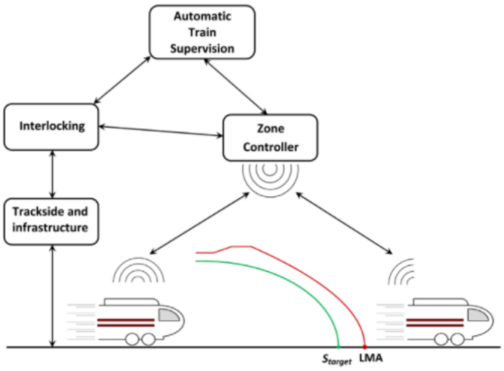


Figure 2.1: CBTC, extracted from [5]

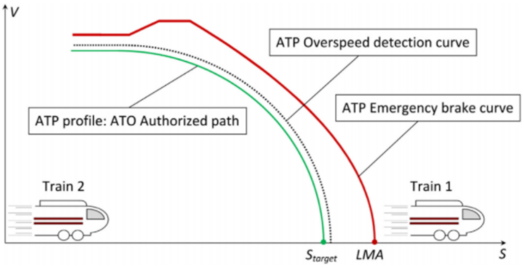


Figure 2.2: CBTC, extracted from [5]

### 2.1.2. Motivations for using moving block signalling

It is of importance to have a clear understanding of why dispatchers and train operators would favor a moving block signalling system over traditional fixed block. In the papers listed in Table 2.1, several motivations for applying the moving block concept are named. First, and most prominently, the benefit of headway reduction is mentioned [5, 27, 37, 46, 56, 60]. As described in [27], the moving block principle was introduced a few decades ago to reduce headways between two successive trains, separating them by their current braking speed plus a safety margin. This reduced headway can also be described as an increase in capacity [56]. With an increasing operational density of railway systems, traditional fixed block systems cannot meet the needs for extra capacity, whereas moving block signalling systems can achieve a higher performance [56, 37].

Besides increased capacity, literature claims several additional benefits of moving block applications, as can be observed in Table 2.1. These advantages are discussed in a more elaborate description of the potential of moving block signalling systems in the next chapter, in Section 3.1.

### 2.1.3. Problem types and objectives

The papers listed in Table 2.1 have different problem types and objectives. Most related to this thesis are the papers that consider a (re)scheduling problem. The authors of [25] provide a summary of the COMBINE project, as executed on behalf of the European Union in 2000. This work describes conflict detection and resolution methods under moving block signalling for improved railway operations. The authors of [40] and [46] focus on scheduling problems too, but in the CBTC approach. Rescheduling problems are extensively discussed in the next section.

Some of the papers from Table 2.1 focus more on the train control aspect, like the work of [5]. The authors use fuzzy algorithms to optimize distance control between successive trains, with the goal of minimizing energy consumption. In [56], the researchers apply pseudospectral methods for optimal trajectory planning. In their work, trajectory planning is presented as an optimal control problem for train driving. Similar to the work of [5], reduction of energy consumption is a primary research objective. In addition to these two papers, the research theme of [27] also revolves around energy consumption problems. The authors investigate peak demand reduction strategies in a moving block operated train network. Regarding the three works described here, it can be said that in relation to train control under moving block, energy efficiency is a recurring theme of research that is considered of importance.

In relation to papers that are more focused on traffic management, objectives consist of performance and improved operations [37, 25], feasibility of timetables [46], and deadlock avoidance [40]. According to Table 2.1, it could be said that energy efficiency is related more to train control, and operational efficiency is related more to traffic management.

Insight has been provided in how moving block signalling has been reviewed in current literature, both in relation to train control and traffic management. As stated in the introduction, this thesis focuses on the latter. Therefore, it is of importance to investigate methods used in this field.

## 2.2. Real time railway traffic management

This section assesses literature related to conflict detection and resolution methods, which are often referred to as problem solving methods for the real time railway traffic management problem (RTRTM problem), or Train Timetable Rescheduling (TTR). In [3], TTR is defined as follows: ”TTR (also called train dispatching or conflict detection and resolution) consists of adjusting in real-time an existing timetable that has become infeasible due to unpredicted disturbances or disruptions. The aim of TTR is to quickly re-obtain a feasible timetable of sufficient quality.” This description clearly indicates the difference between a real time- and a regular railway scheduling problem. In an RTRTM problem, the original timetable, as designed following a scheduling problem, has become infeasible and the goal is to reach renewed feasibility.

In current literature, multiple methods and models with different assumptions and solution methods have been applied to come to feasible solutions for rescheduling. Table 2.2 shows an overview of papers published in the research field of real time railway traffic management. The table is organized according to the following characteristics:

- **Signalling system** Indicates the type of signaling system assumed in the described paper, for example fixed block, moving block or a hybrid form.
- **Infrastructure level** Indicates whether the used infrastructure level is macroscopic, mesoscopic or microscopic
- **Optimization architecture** Describes if the solution is found via a centralized or a non-centralized approach
- **Problem type** This category describes what type of problem the paper is dealing with, for example retiming or rerouting or a combination of multiple problem types
- **Optimization objective** Indicates to which objective the paper is solving, like delay minimization
- **Model** Shows what type of model is used to represent reality
- **Solution method** Shows what solution method has been applied to solve the objective function
- **Remarks** This category is filled in when a paper has a very explicit or interesting distinction in relation to the other papers

### 2.2.1. Problem assumptions

The first three columns of Table 2.2 describe some important assumptions and scope definition aspects of the listed papers. First of all, the signalling system, second of all the infrastructure level and finally the optimization architecture.

**Signalling systems**

The assumed signalling system describes whether the concerned railway network uses a fixed block system, a moving block system or a hybrid or extended form of either of those two options. The characteristics of both signalling systems have been briefly explained in the introduction of this thesis. In Table 2.2, it is clearly visible that a majority of the papers regarding real time railway conflicts still applies fixed block signalling systems. However, some hybrid forms can also be found. The paper [24] describes an overview of the European project COMBINE 2, where both moving block and fixed block systems are used. This approach emerges from a realistic view on implementation of moving block signalling systems, namely that railway networks will go through a transition phase, where different control areas are either equipped with fixed block or moving block signalling.

The authors of [58, 59] describe a quasi moving block signalling system. This implies that the railway network is still divided into fixed block sections, but the safety distance between trains is variable according to current speeds.

The only paper in Table 2.2 applying a fully moving block signalling system is [25], which is the overview paper of the COMBINE project. It can thus be seen once again that moving block signalling systems are highly under-researched when it comes to conflict detection and resolution.

**Infrastructure level**

When taking on an approach towards solving railway traffic management problems, it is of importance to establish and assume an infrastructure level. This can either be macroscopic, mesoscopic or microscopic. Macroscopic models consider a coarse outline of the network, considering stations or end points and the connections between them. Microscopic models consider a detailed outline of the network, considering all section limits, junctions, switches, and station platform details. A mesoscopic model can be seen as a hybrid form of macro- and microscopic, for example only considering some junctions that are deemed important for the specific problem.

Regarding current literature as displayed in Table 2.2, most of these papers take on a microscopic approach. This is not surprising, since proper conflict resolution calls for details on when a specific train will reach a specific point in infrastructure to ensure high solution quality [38].

**Architecture**

Regarding the fact that this research acknowledges the differences between centralized and non-centralized optimization, it is valuable to analyse the architectures of the traffic management and train control systems applied in the papers of Table 2.2. In a centralized architecture, all information is sent to one overarching computer of calculator that uses all data to come to a solution. In large, complex problems, this can lead to high computation times or fault propagation, meaning it could be desirable to decompose the system and come to a solution via a non-centralized approach. The concept of non-

Table 2.2: Published literature in the research field of real time railway traffic management

| Reference | SS | IL | Architecture | Problem type | Objective | Model | Solution method |
|---|---|---|---|---|---|---|---|
| Corman 2011 [15] | FB | Mi | Centralized<br>Hierarchical | Rescheduling | Delay minimization | AG based job shop | Iterative<br>Branch & Bound |
| Corman 2014 [10] | FB | Mi | Centralized<br>Distributed | Rescheduling | Schedule optimality<br>Reduced computation time<br>Delay minimization | AG based job shop<br>Border graphs | Branch & Bound<br>Heuristics |
| D'Ariano 2007 [16] | FB | Mi | Centralized | Conflict resolution | Delay minimization<br>Restore feasibility | AG Based job shop (fixed & variable speed) | Iterative scheduling algorithm |
| D'Ariano 2008 [18] | FB | Mi | Centralized | Rerouting<br>Reordering | Delay minimization<br>Improve punctuality | AG based job shop | Branch & Bound<br>Local search |
| Giannettoni 2004 [24] | Mixed | Mi | Distributed<br>Hierarchical | Traffic management | Management efficiency | graph theory | - |
| Giuliari 2000 [25] | MB | Mi | Centralized | Traffic management | Management efficiency | - | - |
| Luan 2018 [39] | FB | Mi | Centralized | Reordering<br>Retiming<br>Speed profile generation | Delay reduction<br>Speed management | MILP<br>MINLP | Custom two step approach |
| Luan 2020 [38] | FB | Mi | Distributed | Retiming<br>Reordering<br>Rerouting | Increase efficiency<br>Reduce delays | MILP | MILP solver |
| Mazzarello 2007 [41] | Mixed | Mi | Distributed<br>Hierarchical | Conflict resolution<br>Speed profile generation | Improve punctuality<br>Increase capacity<br>Save energy | AG | Multi step algorithm<br>Heuristics |
| Narayanaswami 2015 [43] | FB | Ma | Distributed | Rescheduling | | | |
| Pellegrini 2015 [45] | FB | Mi | Centralized | Rescheduling<br>Rerouting | Minimize delay propagation | MILP | RECIFE-MILP |
| Xu 2017 [58] | QMB | Mi | Centralized | Rescheduling | Minimize delay<br>Increase optimality | AG based job shop | Commercial solver |
| Xu 2021 [59] | QMB | Mi | Centralized | Rescheduling<br>Rerouting | Efficiency (?) | AG based MILP | Two step algorithm<br>Commercial solver |
| Keita 2020 [32] | FB | Mi | Centralized (?) | Rescheduling<br>Rerouting | Limit delay propagation | MILP | RECIFE-MILP<br>Benders' decomposition |
| Corman 2010 [12] | FB | Mi | Centralized<br>Distributed | Rescheduling<br>computation time | Minimize delay<br>Reduce | AG<br>Border graphs | Add & remove |
| Corman 2017 [14] | FB | Mi | Centralized | Delay management | Delay minimization<br>Passenger satisfaction | MILP | Heuristics |

SS = signalling system IL = infrastructure level FB = Fixed Block MB = Moving block QMB = Quasi moving block Mi = microscopic Ma = macroscopic

centralized approaches will be further detailed in Section 2.3. However, it is already worth noticing that most papers from Table 2.2 take on a centralized approach, meaning non-centralized methods have not been as excessively investigated.
Overall, both moving block signalling systems and non-centralized optimizations for real time traffic management are not very widely represented in current literature.

### 2.2.2. Problem types and objectives

The fourth and fifth column of Table 2.2 describe the problem type and its corresponding objective.

**Problem types**
As described in the introduction of this work, several actions can be taken when handling railway traffic. These actions mainly include retiming, reordering, rerouting, cancelling, and short turning. The term rescheduling is also used frequently, which implies a selection or all of the aforementioned actions. From Table 2.2, it can be concluded that rescheduling problems concerning retiming and reordering are the most recurring problem types within current literature. Rerouting is covered slightly less. The work [58] has first been performed, covering retiming and reordering only. The recommendations for future work in [58] include the extension to a rerouting problem, which has then been executed in their next work [59].

In the works of [39] and [41], the traffic management method not only includes conflict detection and resolution methods, but also speed profile generation. This is out of scope for the rest of this research, but it is interesting to have some understanding of this principle. As is explained in [41], a CDR module (Conflict Detection Resolution) is responsible for train rescheduling and rerouting, by detecting conflicts and creating a new schedule. The SPG module (Speed Profile Generator) then uses this information to compute a speed profile that will achieve the goals set by the CDR. The SPG thus has a lower place in system hierarchy than CDR [41]. In [39] they have embedded speed profile generation in a pre-processing stage of an integrated conflict resolution method. This means that the traffic management problem and the speed profile generation, also called train control problem, are resolved in a highly integrated manner. Figure 2.3 shows an example of a train speed profile spread over two cells, which is their way of representing block sections. For details of their method, refer to [39]. The authors of [39] also make an interesting note that there is the possibility to have SPG higher in the hierarchy then CDR, although it is not very common.

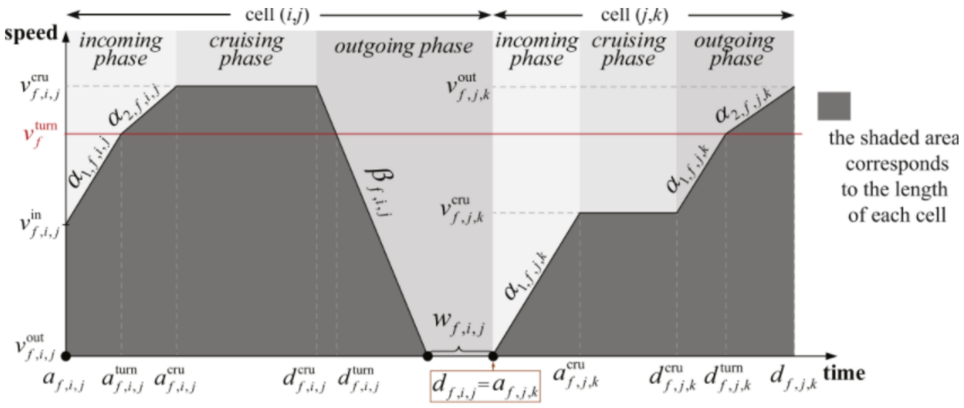


Figure 2.3: Example of a train speed profile in two cells (block sections), retrieved from [39]

**Objectives**
Similar problems can be solved for different objectives. In Table 2.2, multiple objectives can be observed, including delay minimization, restoration of feasibility, saving energy, increasing passenger satisfaction, increase capacity, increase punctuality, and reducing computation time. In relation to the conflict detection and resolution, the minimization of delay is the most prominent objective appearing in

current literature. Regarding the principles of railway traffic management, as explained in the introduction of this thesis, this is a very logical objective to be appearing regularly. This is taken into account in the rest of this thesis.

### 2.2.3. Problem formulations

Problem formulations are a representation of reality. There are three formulation types, that can complement each other or be combined, that keep recurring in the current literature as displayed in Table 2.2. These formulations are the following, and will be clarified in this section.

- Job shop scheduling problem
- Alternative graph
- Mixed Integer Linear Problem (MILP) or a variation thereof

**Job shop scheduling problem**

Job shop scheduling problems are, according to current research, very suitable to represent railway optimization problems. The classic job shop problem consists of $n$ jobs that have to be processed on $m$ machines under the following assumptions, as described by the authors of [4]:

- a machine can only process one job at a time
- the processing of a job on a machine is called an operation
- an operation cannot be interrupted
- a job consists of at most $n$ operations
- the processing order of a job is given according to this job
- the operation sequences of a machine are unknown

In the work of [44], this job shop problem is applied to the railway conflict resolution problem, where trains are viewed as jobs and tracks or track sections are seen as machines. Additional constraints can be added to more accurately represent real life conditions, as has been done in a majority of papers using this problem formulation. Additional constraints can include connections between trains [18], headway constraints, priorities [15] or variable operation times [58]. This formulation will be more extensively explained in Chapter 4.

**Alternative graph**

The alternative graph (AG) formulation is a recurring approach in railway traffic management problems. This subsection provides a brief description, since an elaborate explanation of this method is given in Chapter 4. In the AG approach, the problem is formulated as a set of nodes, a set of directed arcs (fixed arcs) and a set of pairs of directed arcs (alternative arcs), as explained in [16]. Every node is associated with an operation, whereas each arc represents a precedence relation constraining the starting time of each operation. Every fixed arc represents the running time of train through a block section, however two trains cannot occupy one block section at the same time. Therefore, the choice of alternative arcs represent the order of trains, where the arc length represents the minimum headway constraint. Figure 2.4 shows the basic principle of alternative graph theory in railway applications.

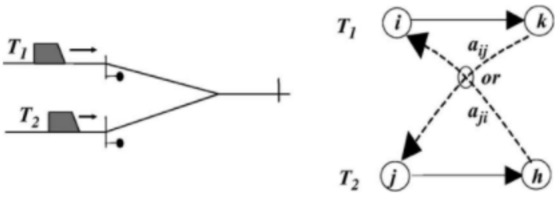
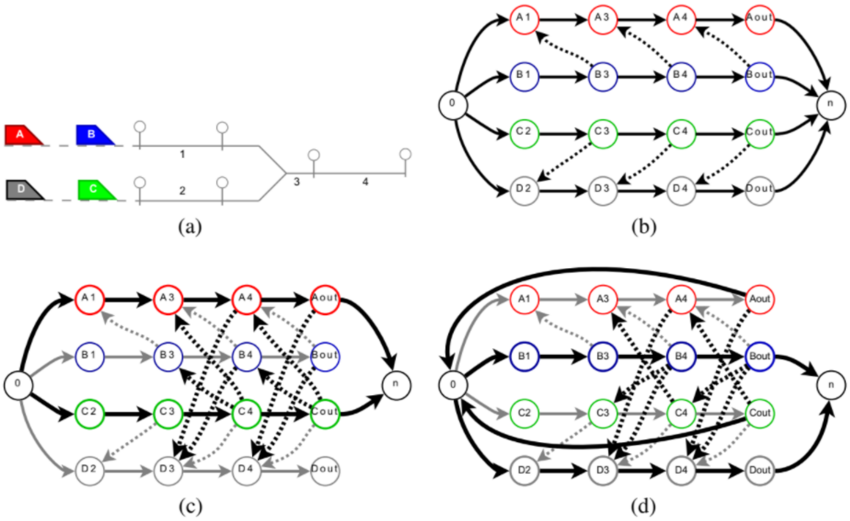


Figure 2.4: The principle of alternative graph formulation [16]

When looking at multiple sections, junctions or tracks, alternative graph representations can get more complicated. This is nicely visualized in the work of [15], where the authors also make use of an AG based model, see Figure 2.5. This example shows four trains and four sections. As can be noticed from Table 2.2, AG theory appears prominently in current literature on railway traffic management [15, 10, 16, 18, 41, 58, 59, 12].



. (a) Railway area and trains, (b) initial graph selection, (c) first iteration and (d) second iteration. Example of the iterative framework via alternative graphs.

Figure 2.5: An alternative graph model for multiple trains and sections [15]

**MILP**

Another recurring type of problem formulation is Mixed Integer Linear Problem, or MILP. MILPs consist of a set of input variables and decision variables, which can either be continuous or binary, that make up a problem formulation and a set of constraints that the problem is subjected to. The goal is to reach an optimal value of the objective function, of which the following examples are found in literature of Table 2.2:

- minimize the sum of mean absolute delay [39, 38]
- minimize total weighted delays [45, 32]
- minimize total times of changing siding lines as well as train tardiness [59]
- minimize the total time spent in the system by all passengers [14]

The objective function as described in the work of [14] shows merging two streams of research, namely the network manager's and train operator's point of view. It takes into account both delay minimization and the influence on passengers and passenger satisfaction. Whilst the alternative graph model and job shop scheduling problems are highly focused on delay minimization, the MILP more easily provides room for other objectives to be implemented as well.

Now that these three main formulations have been discussed, it should be noted that a fair share of the papers from Table 2.1 use an alternative graph based job shop model [15, 10, 16, 18]. This can be viewed as a very suitable model for handling railway delay minimization problems. However, MILP formulations also have advantages as described above. This is important knowledge for the rest of this work and will be taken into account in Chapter 4.

## 2.2.4. Problem solution methods

In Table 2.2, a large variety of solution methods is presented, containing tailored algorithms, heuristics and commercial solvers.

In terms of algorithms, branch and bound is a favourable method for solving job shop scheduling problems [12, 16, 18]. In [12], the authors use the branch and bound method used in [16] as a building block, but tailor the solution method to what they call an add remove algorithm. A share of the papers

from Table 2.1 uses tailored solution algorithms containing two or more distinguishable steps. For example in [39], where multiple problems formulations are investigated, a variety of tailored algorithms is applied. First, a custom two level heuristic, based on a genetic algorithm, to solve the railway real time rescheduling problem. Second, a tailored two step approach that enhances the performance of MILP solvers for solving the rescheduling problem, taking into account train speed profile options [39]. Something similar has been executed by the authors of [59], where additional rules are implemented in the solution algorithm, to improve MILP solver solutions of the rescheduling problem. The authors of [41] first use heuristics for rescheduling, then an additional algorithm for rerouting to further optimizing the solution.
A very specific state of the art method to solve real time railway traffic management problem is RECIFE-MILP [45]. This method has been developed as part of the decision support tool RECIFE ((REcherche sur la Capacité des Infrastructures FErroviaires). RECIFE-MILP is a heuristic algorithm based on a Mixed Integer Linear Programming model. The performance of this method is assessed in [45] and also taken further by other authors, like in [32]. In this paper, the RECIFE-MILP algorithm is enhanced with a three step benders decomposition [32].

In Table 2.2, it can be seen that job shop scheduling problems, and therefore alternative graph formulations, are often solved with branch and bound (inspired) methods or heuristics. MILPs are often solved with more tailored methods and algorithms or commercial MILP solvers.

Overall, looking at Table 2.2, it can be stated that a huge majority of the papers on real time railway traffic management applies fixed block signalling systems. Problems are usually formulated on a microscopic level, for example using the alternative graph formulation. The leading objective is delay minimization, other objectives like passenger satisfaction or energy savings are secondary objectives. The centralized optimization architecture is also the most frequently occurring in Table 2.2, however there are some papers that apply non-centralized strategies. The next section will dive more into non-centralized architectures in the railway sector, not only taking into account papers on real time traffic management, but also other railway applications.

## 2.3. Non centralized strategies in railway traffic management

This section focuses on research devoted to non-centralized methods in railway applications. As stated in the introduction, this research does not yet cover a non-centralized approach. However, it is deemed valuable to investigate what these approaches are and why they are desired, partly to make interesting recommendations for future research in Chapter 7.
Referring again to the definition from [3], ”The aim of TTR is to quickly re-obtain a feasible timetable of sufficient quality.” A keyword in this sentence is ”quickly”, referring to the fact that in real time traffic management, solutions need to be generated fast. One of the ways to accomplish this is to apply non-centralized methods for optimization, since this can significantly reduce the computation time of problem solving. Non-centralized methods can include decentralized, distributed or multi agent systems. These terms can have different meanings in different fields of application, but in this thesis the following definitions will be employed:

**distributed system:** system decomposed into subsystems, where subsystems in the same layer of hierarchical can communicate
**decentralized system:** system decomposed into subsystems, where subsystems in the same layer of hierarchical can not communicate
**multi agent system:** a system of multiple intelligent subsystems, not specified whether this functions in a distributed or decentralized manner

This section will review literature that executes strategies for either three types of non-centralized optimization. Note that the literature discussed in this section does not necessarily concern real time operations only. The reason for this is to demonstrate a broader view on the possibilities for non-centralized methods. Table 2.3 shows a list of papers all concerned with non-centralized approaches, categorized according to the following properties:

- **Problem type** This category describes what type of problem the paper is dealing with, for example real time rescheduling or maintenance planning
- **Motive for Non-Centralized Optimization** This category indicates the motive(s) mentioned to apply a non-centralized method
- **Type of Non-Centralized Optimization** Describes if the paper executes distributed, decentralized or multi agent optimization
- **Type of Decomposition** In a non-centralized architecture, a problem or representation thereof is divided into subunits. This category describes the type of partitioned groups the problem is divided into
- **Method of Non-Centralized Optimization** The solution method used to solve the optimization problem

### 2.3.1. Non-centralized problem types

Looking at the second column of Table 2.3, the majority of the papers considers rescheduling problems. However, papers with other problem types are also reviewed.

The authors of [23] apply distributed optimization approaches for maintenance planning of railway tracks. In their paper, it is explained that scheduling of maintenance operations for large scale railway networks is a challenging task. This is also supported by [61], who propose an optimization strategy combining track scheduling and maintenance planning.

Another recurring problem type is concerned with energy management of railway networks. The work of [35] explains that the modern railway system is a massive grid connected complex system with distributed active loads, sources and storage of energy. This implies that energy management in railway systems can be aided with distributed methods. In addition to this, the work of [19] explains that distributed optimization has been applied in different aspects of electrical engineering and can therefore be very suitable for energy management.

Inspiration from papers concerning scheduling or rescheduling problems can be complemented by methods used in other problems, as described above. Regarding papers on (re)scheduling, it should be noted that in the papers in Table 2.3, traditional fixed block systems are favoured. This is not as explicitly addressed as in the previous section, but still holds.

### 2.3.2. Non-centralized approaches

When reviewing the applications for non-centralized strategies for problem solving, it is of importance to understand the different motives for- and types of non-centralized approaches. Additionally, the different possible decomposition types of the problem are interesting aspects as well.

The second column of Table 2.3 indicates the reason for applying a non-centralized approach as described in the corresponding paper. Frequently occurring motivations are increased computational efficiency, or decreased computation time, [12, 32, 38, 43, 24, 34, 6, 23, 61] and increasing the scalability or large scale suitability of the system [12, 43, 10, 41, 23, 35]. The decrease of computation time is highly related to papers concerning rescheduling problems. Due to the real time nature of these problems, a solution is required in a very short computation time for resolving train conflicts as quickly as possible. However, the computation time for finding an optimal solution increases exponentially when enlarging the scale of the problem instances. In relation to this, distributed optimization has gained a lot of attention to face the need of fast and efficient solutions for problems arising in the context of large-scale networks [38].

Regarding increased suitability for large scale operations, the authors of [23] explain that traditional solution schemes can be highly dependent on primary problem structures, and are therefore not always flexible enough to apply in large scale networks. The authors of [35] even define a quantified performance metric to prove that decentralized approaches are more scalable than centralized methods.

Another important motivation includes (re)establishment of feasibility, which also strongly relates to rescheduling problems [10, 34, 11]. In large scale networks, there is a risk of local solution not being globally feasible, and smart choices in non-centralized optimization can help preventing this disadvantage [33]. This is stressed by the authors of [33], stating that they will specifically focus on global railway control instead of local problem solving.

Table 2.3: Published literature in the research field railway traffic management using distributed methods

| Reference | Problem Type | NCO motive | NCO type | Decomposed into | NCO method |
|---|---|---|---|---|---|
| Corman 2010 [12]* | Rescheduling | Improve computation efficiency<br>Increase large scale suitability | De | Two areas | Coordination level<br>Information exchange |
| Keita 2020 [32] | Rescheduling | Increase computational efficiency | Mat | subproblems | benders decomposition |
| Luan 2020 [38] | Retiming<br>Reordering<br>Rerouting | Improve computational efficiency | Di | Subareas<br>Singular trains<br>Time intervals | ADMM algorithm<br>PR algorithm<br>CDRSBK algorithm |
| Narayanaswami 2015 [43] | Rescheduling | Increase scalability<br>Reduce computational complexity | MA | Agents | Bidding heuristic |
| Corman 2014 [10] | Rescheduling | Increase feasibility<br>Increase scalability | De?? | Subareas | Feasibility checking |
| Giannettoni 2004 [24] | Conflict resolution (rescheduling) | Distribute dispatching responsibility<br>Decrease complexity<br>Increase computational efficiency | De | Subareas | Information exchange<br>Feasbility checking |
| Mazzarello 2007 [41] | conflict resolution (rescheduling)<br>speed profile generation | Expand system suitability<br>Increase model scalability | De | Subareas | Feasibility checking<br>Aggregate information exchange |
| Corman 2012 [11] | Rescheduling | Establish global feasibility<br>Increase level of coordination<br>Increase global optimality | De? | Subareas | Feasibility checking<br>Branch & Bound algorithm |
| Hassanbadi 2013 [30] | Train control problem | Reduce complexity<br>Increase reliability | Di | Agents | |
| Kersbergen 2016 [33] | Rescheduling<br>Rerouting | Decrease computation time | Di | Subareas | Model based partitioning<br>MPC based algorithm |
| Kersbergen 2014 [34] | Rescheduling | Ensure feasibility<br>Decrease computation time | Di | Successive<br>Subareas<br>subproblems | Tailored MPC based algorithm |
| Cavone 2020 [6] | Rescheduling | Reduce computation time | Di | Subareas | MPC based algorithm<br>bi level heuristics |
| Faris 2018 [23] | Maintenance planning | Reduce computation time<br>Increase scalability | Di | Lagrangian based subproblems | PALR algorithm<br>ADMM algorithm<br>DRSBK algorithm |
| Jiang 2020 [19] | Energy management | | Di | | ADMM algorithm |
| Khayyam 2016 [35] | Energy management | Handle complexity<br>Large scale suitability | De | Subnetworks | - |
| Roberts 2002 [53] | Fault detection | Economic and efficient diagnosing | Di | Subareas | Neuro fuzzy networks |
| Zhu 2020 [62] | Power flow optimization | Stabilize learning process | De | Subsystems<br>Agents | Deep learning |
| Zhang 2021 [61] | Rescheduling<br>Maintenance planning | Efficient operation<br>Decrease computation time<br>Realize feedback correction | Mat | Lagrangian based subproblems | Rolling horizon algorithm |

NCO = Non Centralized Optimization De = Decentralized Di = Distributed MA = Multi Agent Mat = Mathematical

Other motives are related to distributing responsibilities [24], efficient fault diagnosis [53] and stabilizing the learning process in a deep learning application [62].

Different non-centralized architectures can be found in the papers reviewed in Table 2.3. In this table, a distinction has been made solely between distributed, decentralized or general multi agent oriented approaches. However, the way a non-centralized approach is executed can differ between papers.

### 2.3.3. Non-centralized problem solution methods

Regarding problem solution methods, a variety of different algorithms and heuristics can be found in Table 2.3, whereof some recurring strategies can be named.

First, a share of the papers from Table 2.3 applies a decentralized solution procedure, where an overarching layer over the whole network performs feasibility checks on local solutions [11, 10, 24, 41]. A visualisation of this principle is given in Figure 2.6, from the paper [11]. The authors of [11] explain that decision taken locally may influence the quality and feasibility of train schedules in other areas. The goal of the overarching coordinator is to ensure global feasibility of train schedules, which could imply imposing constraints on subareas. It should be noted that global feasibility is not necessarily proportionate to global optimality. The authors of [11] extend the role of the regional coordinator by exploring possibilities to pursue global optimality as well.

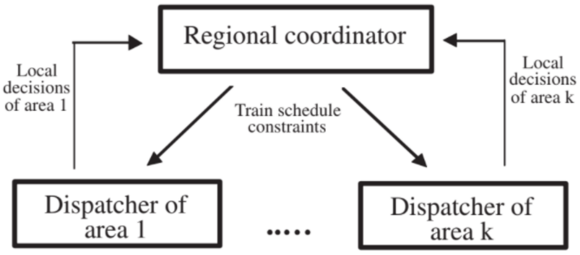


Figure 2.6: System architecture of separate areas and an overarching feasibility check, extracted from [11]

Second, the concept of MPC (model predictive control), also called rolling or receding horizon [61], is a frequently used principle in distributed applications [34, 33, 6, 61]. As stated in [33], ”MPC is a control methodology that, at discrete time instants, determines the control inputs for the system that minimize a cost function based on a prediction of the evolution of the state of the system under control.” In distributed MPC system, suitable for large and complex systems like railway networks [34], the system is divided into subsystems, where each subsystem is controlled by a separate MPC controller. In the works [34, 33], written by the same author, a specific method is used to decompose the system, which the writers refer to as model-based partitioning.

Lastly, various types of algorithms and heuristics can be used to solve a non-centralized problem. A recurring algorithm is the Alternating Direction Method of Multipliers (ADMM algorithm) [38, 23, 19]. The ADMM algorithm applies Lagrangian relaxation to iteratively solve subproblems. The authors of [19] motivate their choice for this algorithm by stating it can be used whilst protecting the privacy of different parties.

## 2.4. Conclusion

After analyzing all literature in the preceding sections, an overview representing the state of the art regarding conflict resolution under moving block signalling can be made based on four aspects. These aspects are:

- Problem type
- Signalling system

- Solution architecture
- Application field

After assessing the literature represented in Table 2.1, 2.2, and 2.3, the highly relevant papers can be categorized according to these four characteristics, as has been done in Table 2.4. The first row of this table shows related works that consider railway conflict resolution problems for fixed block signalling systems, with a centralized solution architecture. As can be seen from the table, this area is widely researched in various sources. The second row shows related works that consider train control under moving block, in metro/subway applications. This is not as prominently present in literature as the previous problem type. The third row depicts literature papers that focus on various traffic management problems that have adopted a non-centralized solution architecture. However, signalling systems are not always explicitly mentioned, and if they are it is mostly fixed block. The last row of Table 2.4 shows the combination of problem type, signalling system, solution architectures and application field that is underresearched and will be investigated in this thesis. This table thus answers research question 1 and consequently shows the gap in current literature, namely that there has been no research performed that covers:

- Conflict resolution
- Under moving block signalling
- For railway applications
- (Acknowledging the difference between centralized and centralized optimization)

This stresses and confirms the contribution of this thesis as mentioned in the introduction, namely that this research will introduce a conflict resolution approach for moving block signalling systems. Non-centralized approaches will not yet be explored in this work, however this thesis will be performed with the knowledge of non-centralized problem solving whilst keeping the possibilities and added value of non-centralized options in mind.

| Problem Type | Signalling System | Solution Architecture | Application Field | Related Works |
|---|---|---|---|---|
| Conflict resolution | Fixed Block | Centralized | Railway | [16, 18, 17, 15, 10, 14, 45, 32, 39] |
| Train control | Moving Block | Centralized | Metro/subway | [5, 46, 40] |
| Traffic management | - | Non-centralized | Railway | [24, 41, 43, 23, 38, 61] |
| Conflict resolution | Moving Block | Centralized (Non-Centralized in mind) | Railway | **This thesis!** |

Table 2.4: State of the art and overview and research gap, to be filled by this thesis

# 3

# Conflict resolution under moving block signalling

This chapter will provide insights in the principle of moving block signalling and its corresponding considerations for conflict resolution. First, it is important to understand the information flows present in a moving block signalling system. Subsequently, attention will be drawn to performance measurement for conflict resolution. This will be done by formulating key performance indicators to evaluate the effectiveness of the defined conflict resolution approach. At the end of this chapter, research questions 2a-2d will be answered in the conclusion.

## 3.1. The potential of moving block signalling systems

The introduction of this research already explained the basic differences between fixed and moving block signalling systems. Chapter 2 has shown how moving block has been represented in current literature. This section will provide some more detailed information on railway signalling systems and the advantages of shifting to a moving block signalling system. Table 3.1 shows the characteristic differences of fixed and moving block signalling systems. The three most essential differences are provided in the top three rows of Table 3.1. In this research it is assumed that moving block signalling operates with solely trainside measurements. It is also possible to combine track- and trainside information, leading to a hybrid, or mixed, signalling system [24]. More information on this can be found in [24]. Because of these three main characteristic differences, moving block signalling contains several

| Fixed block | Moving block |
|---|---|
| Information limited by trackside measurements | Information continuous by trainside measurements |
| Safety maintained by trackside signals | Safety maintained by continuous monitoring and feedback |
| Capacity restricted by number of block sections | Capacity restricted by minimum headway between trains (absolute braking distance + safety margin) |
| | Operational cost reduction by saving costs to maintain track-side signalling equipment |
| | Potential for higher speeds [60] |
| | Potential for decrease in conflicts and increase in traffic fluidity [58] |
| | Potential for decrease in traction energy [25] |
| | Potential for higher competitiveness with other transport modes [25] |

Table 3.1: The characteristic differences between fixed and moving block signalling systems

potential advantages that can improve railway operations. Knowledge on these advantages has been gathered from the papers already considered in Chapter 2, and will now be further elaborated.
A very important advantage is cost reduction, because of the decrease in track side equipment maintenance. This is currently a large share of operational costs, which can be largely decreased when moving equipment trainside. Current literature also mentions several other moving block benefits. In [58], where they adopt a quasi-moving block signalling system, it is explained that intelligent speed management decrease conflicts and increase the fluidity of traffic. Especially in terms of reordering of trains in heterogeneous timetables, where fast trains, like intercities, can be given the opportunity to avoid slow driving trains, like sprinters. Another advantage of moving block is explained by the authors of [60], who highlight that the potential to drive much closer together and shortening the minimum distance between trains leads to a more efficient system with higher possible driving speeds. Related to both the driving speed and intelligent management of speed is the potential of decreasing traction energy, as explained in [25]. Moving block signalling allows more smooth control of traffic and reduces the number of times that trains need to come to a complete stop. This leads to reduced traction energy. Lastly, the authors of [25] mention another very important motivator for moving towards moving block signalling systems. Due to the increased railway capacity, there is an increase potential for railway transport to compete with other transport modes.

## 3.2. Information flows in a moving block signalling system

As explained in the introduction of this research, a moving block signalling system requires continuous speed and position measurements to be sent to a dispatcher, or traffic management system. In turn, trains need to obtain adjusted speed profiles for traffic fluidity and safe distancing. It should be noted that moving block signalling and ERTMS level 3 are often mentioned in combination with automated train operations (ATO), both in current literature [5, 46] and at ProRail. However, in this research it is not distinguished whether the trains have a certain degree of automation or not, since the scope of this research is limited to traffic management and does not consider train control.

Figure 3.1 [41] shows the information loop in ERTMS level 3 for a moving block case. This figure is used to create a general understanding of information in moving block signalling systems, it is possible that other works use or assume a different version. Figure 3.1 mentions intermediate steps from train to TMS and from TMS to train. Speed and position information in the train is collected by sensors and given to GSM-R (Global System for Mobile communication - Rail) [42]. Next, that is forwarded via Radio Based Communication (RBC) to TMS. The TMS translates these current measurements of speed and position into multiple types of information. This information includes train advisory speeds, new arrival- and departures times, and update train orders and routes. This is forwared via the same communication train to the train driver (if there is one). This research focuses on the top right corner of this figure, namely the traffic management system.
Besides a clear visualization of the ERTMS level 3 working principle, this figure demonstrates another

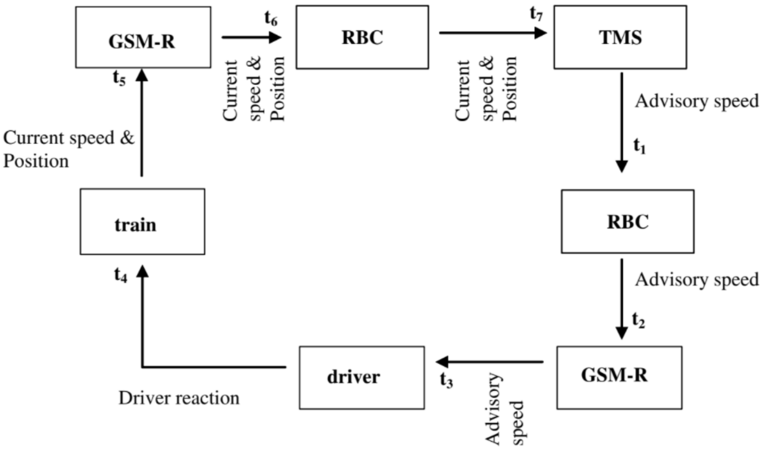


Figure 3.1: The full information loop in ERTMS level 3 (for a moving block case) from [41]

interesting aspect. For every communication step, there is a communication delay, adding up to the total time it takes to go from a measurement by the sensor to an implemented advisory speed. These are important aspects in terms of conflict resolution and real time traffic management. Real time operations call for fast solutions and it is desirable to keep the entire time elapsed through the communication loop as small as possible [41]. In terms of TMS, the focus area of this research, it is desired to develop fast solution algorithms that can solve the conflict resolution problem in short time periods. Slow functioning algorithms can be a limiting factor in this information flow cycle, which is an undesired effect [41]. This will be elaborated more in Section 3.4, that focuses on key performance indicators.

This thesis thus describes the development of a decision support system to optimally solve rail traffic conflicts under moving block signalling. Figure 3.2 shows a more detailed view of the information flows related to this research. As can be seen in Figure 3.2, the data that enter the system and the output that leaves the system will need alterations and translations in order to be useful. This research does not focus on how exactly this data is processed and (re)translated. It is thus assumed that everything that is needed to use the decision support system is present in the desired format.

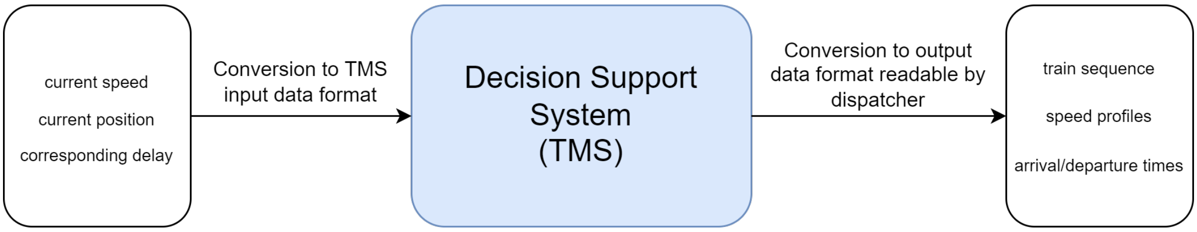


Figure 3.2: Conceptual visualization of information flows for a decision support system for traffic management. Note: the output information are possibilities, not mandatory or definite data

Since this research deals with conflict resolution, the decision support system has the goal of creating a conflict free network from a conflicted network. The dispatcher can then use the output of the system to make decisions on accepting, adjusting or cancelling the proposed solution of the decision support system. It should be stressed that the dispatcher still makes his or her own choices and can either act exactly as the support system suggests, or decide to use another solution after all. This makes it challenging to formulate a suitable output for the dispatcher to use, which is why the output Figure 3.2 is optional. Ways to communicate TMS output to the dispatcher have been designed, however this remains a challenging and iterative task and there is no universal consensus throughout all railway applications. This has become evident from an interview with an expert on this topic, the author of [41]. The reasons this is challenging are the following, also derived from correspondence with this same expert:

- **Unambiguity** the output should primarily be understandable for the dispatcher. It should contain enough information to explain and justify the solution, but not so much information that it becomes unclear or difficult to distinguish what's important. This also has a relation to the next point of challenge.
- **Timely decision implementation** Information should be clear and substantiated, but the dispatcher needs to be able to make quick decisions. Since the goal is to perform real time traffic management and fast decisions are desired, showing too many information that complicates the decision can be paradoxical. Therefore, it can be said there is a trade off between clarity and decision speed within the information shown to dispatchers.
- **User interface tuning** Given the aforementioned tradeoff, it can be a good solution to allow tuning of what is visible to the dispatcher, regarding the dispatchers point of view and preferences. However, this needs thorough research and testing.
- **Effective TMS configuration** Besides options to tune the user interface, solutions could also be developed to tune input parameters and objectives according to dispatcher preferences and specific situations.

It is out of scope for this thesis to dive deeply into the pros and cons of all types of communication to the dispatcher. However, for future research it is a valuable aspect to look into.

## 3.3. Considerations for conflict resolution in a moving block signalling system

In this research, mathematical approaches will be used to perform conflict resolution on railway tracks. Since a mathematical approach is always a representation of reality that can not have one hundred percent accuracy, it is important to be aware of several considerations. These will be discussed in this section.

Considerations can be divided into two types. First, general railway considerations that are of interest for a majority of railway related problems. Second, railway considerations that are especially important to conflict resolution problems. Aspects to take into account for railway problems in general are the following:

- Level of detail (macroscopic/mesoscopic/microscopic)
- Directional use of tracks (uni/bidirectional)
- Train type distinction (passenger/freight trains)

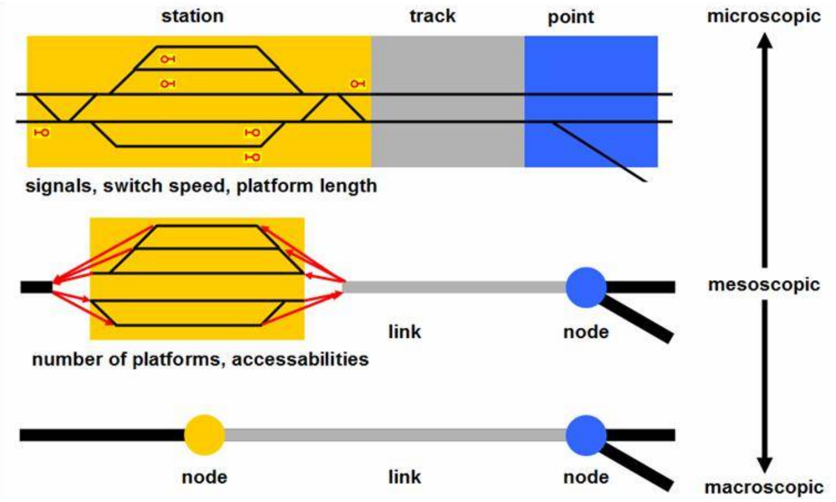


Figure 3.3: Illustration showing different detail levels in railway problem approaches, extracted from [22]

Regarding the first aspect, Figure 3.3 shows the three different detail levels that can be applied to railway problems. These are the same detail levels also found in the papers from Chapter 2, see table 2.3. Because microscopic approaches are highly detailed and take network aspects into account very accurately, a downside is that it usually implies high computation times. Macroscopic approaches are less detailed and suitable to solve large problems much quicker. The drawback is that details are missing from the model, meaning solutions might be infeasible on microscopic level.
For this research, it is desired to go with a microscopic approach. The drawback of possible high computation times is overshadowed by the the fact that solutions need to be feasible on microscopic level. Moreover, since moving block signalling is considered with detailed information on the current speed and position of a train, this approach is most suitable. In addition, this approach is in line with current literature, where most conflict resolution problems are solved microscopically, see Table 2.3. This detail level should be taken into account in Chapter 4, where a mathematical approach will be developed. Additionally, this research will start with looking only at trains driving in the same direction, meaning that bidirectional use of tracks are out of scope. Furthermore, it is assumed that all train types are passenger trains, meaning no explicit distinction is made between passenger and freight trains. It should be noted that considering the difference between passenger and freight trains can be valuable aspect

in conflict resolution problems [11]. Since freight trains usually have an on-demand schedule inserted in buffer paths of the timetable, but also drive at lower speed, this can lead to interesting insights or choices when resolving conflicts. This is left as a recommendation for future research.

Following this, aspects to consider that are specifically related to conflict resolution are as described below:

- Types of delay causes (disturbance/disruption)
- Types of delay considered (primary delay/secondary delay)
- Definition of a conflict

Railway conflicts can emerge due to disturbances or disruptions on the tracks. The authors of [3] provide some clear definitions describing the difference between these two concepts. They state that ”disturbances are relatively small perturbations of the railway system that can be handled by modifying the timetable, but without modifying the duties for rolling stock and crew” [3]. And ”disruptions are relatively large incidents, requiring both the timetable and the duties for rolling stock and crew to be modified” [3]. As stated in the introduction of this thesis, the conflict resolution actions that are taken into account are retiming and reordering of trains. Because of this, only disturbances are analysed in this study. Following the definitions of [3], these two conflict resolution actions might not be enough to solve the problem when disruptions are present in the system.

Regarding delay types, there are two types to be distinguished, also described by [3]. First, there is primary delay, which is caused by the fact that a railway process, such as driving from one station to another or dwelling at a station, lasts longer than planned. Second, there is secondary delay, or propagated delay, which is caused by primary delays being passed on to another train. It is important to state that this research considers track occupation conflicts that occur when two or more trains try to occupy the same portion of track at overlapped time periods. Rescheduling actions can not remove initial delays that caused the conflict, but it can mitigate the effects of the initial delay. This means that conflict resolution focuses on mitigating propagated delay over other trains, in other words secondary delay.

Lastly, it is important to consider the definition of a conflict. Since a moving block signalling system is assumed, the moment of conflict is slightly altered with respect to traditional fixed block systems. Traditionally, a track occupation conflict, within terms of blocking theory, occurs when two trains claim the same block section simultaneously [11]. Figure 3.4, extracted from [48], shows the blocking time of a train in a fixed block signalling system. It can be seen that this blocking time is dependent on the fixed distance between to signals, and consists of:

- Setup time ($t_s$)
- Sight reaction time ($t_{sr}$)
- Approach time ($t_{ap}$)
- Running time ($t_{run}$)
- Clearing time ($t_c$)
- Release time ($t_{rel}$)

If these blocking times of two trains overlap, it means the section is claimed simultaneously by two different trains and a conflict occurs. This is visualized in a conceptual way in Figure 3.6. The overlap of blocking times is demonstrated with dotted lines. Train A has primary delay, because the block length is stretched over the time axis. Train B can inherit this delay, because of the overlapping blocking times. In order to prevent this, rescheduling measures can be taken. In a moving block signalling system, blocking times are slightly different. Figure 3.5 shows the blocking time of trains in a moving block signalling system, which now consists of:

- Sight reaction time ($t_{sr}$)

- Approach time ($t_{ap}$)
- Clearing time ($t_c$)
- Release time ($t_{rel}$)

This shows that in moving block signalling, the setup time and the running time are no longer contributing factors in the blocking time of a train. This means that all components of the blocking time exist of train related parameters, rather than infrastructure, which nicely shows the principle of moving block. The blocks move along with the train and are not related to fixed points anymore (there are some exceptions, which will become clear later in this research). In theory, the definition of a conflict is the effect of two blocking times overlapping. In practice, for fixed block this means block overlapping caused by two trains claiming the same infrastructure section of fixed length. For moving block this means trains driving closer together than the braking distance plus a safety margin, causing the blocking times to overlap. This is visualized in Figure 3.7, where a primary delay of train A causes the trains to drive too close together. This is kept in mind throughout the rest of this research.

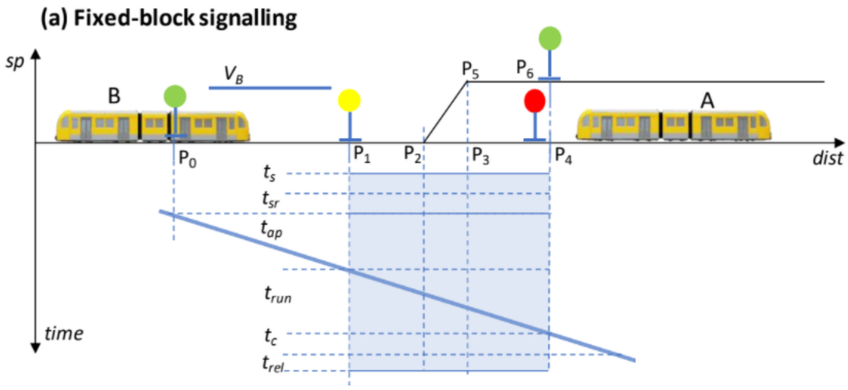


Figure 3.4: Blocking times under fixed block signalling represented in a time distance diagram, extracted from [48]

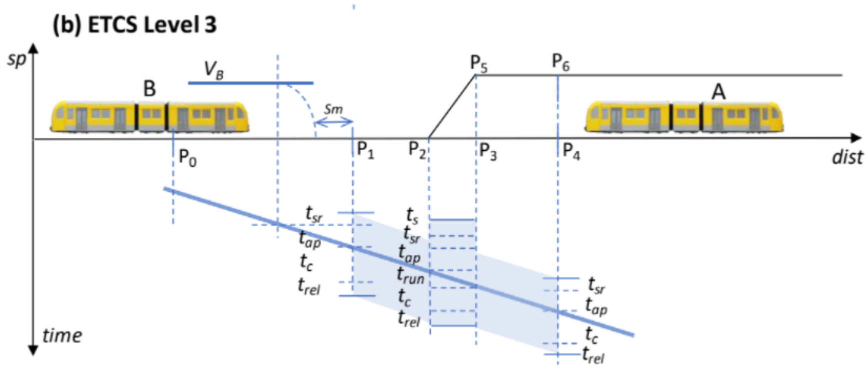


Figure 3.5: Blocking times under moving block signalling represented in a time distance diagram, extracted from [48]

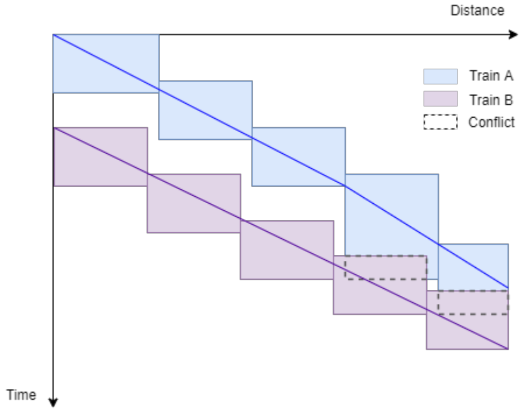


Figure 3.6: Conflicting blocks in a fixed block signalling system

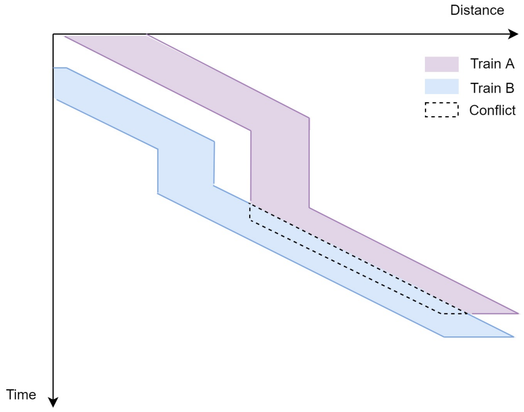


Figure 3.7: Conflicting blocks in a moving block signalling system

## 3.4. Key performance indicators for conflict resolution under moving block signalling

When designing a decision support system for conflict resolution under moving block signalling, it is valuable to look at the performance of this system. This section will discuss performance measurement in two steps. First, a set of possible KPI's will be discussed, that could be used now or in the future. Second, a selection of these KPI's will be made to use in this research.

### 3.4.1. Relevant key performance indicators (KPI's)

When designing a decisions support system, there are multiple options to consider when evaluating on the performance of the system. A collection of relevant KPI's for conflict resolution has been formulated and will be discussed in this section. These KPI's are based on literature review and additional indicators have been added by ProRail as deemed important to their business. Note that not all these KPI's will be used in this research to evaluate the designed method. A selection of KPI's will be made, which will be discussed in the next section.
Table 3.2 shows a collection of possible Key Performance Indicators. The relevant KPI's can be divided into two categories:

- **Computation performance.** This KPI category describes assessment in relation to the solving procedure.
- **Solution quality**. This KPI category focuses on after the solving procedure is done and is related to the characteristics of the final solution

Some of the KPI's from Table 3.2 are especially interesting in relation to moving block signalling.

| Computation performance | Solution quality |
|---|---|
| Computation time | Total propagated delay |
| Solution stability | Punctuality |
| Scalability | Speed feasibility |
| Number of iterations | Number of affected trains |
| | Number of reorderings |
| | Number of retimings |
| | Energy use |
| | Safety risks |
| | Cancelling limit reached |
| | Crew scheduling limit reached |

Table 3.2: Possible KPI's to evaluate a conflict resolution method

For example, as explained in Section 3.1, moving block signalling has great potential for decreasing energy use. When rescheduling trains, one could translate retimings and corresponding speed profile generations to an amount of traction energy. However, since energy saving is not within the scope of this research, this KPI will not be discussed here. Also, safety risks and safety boundaries within the solution could be evaluated. Because of unknown system disturbances on the trains or communication delays, it could be debated how close trains can actually drive together. Since this research remains in simulation environment where no sudden disturbances are present it is assumed that the minimum headway is sufficient to maintain safety. Other possible KPI's include speed feasibility, which indicates whether the provided solution is feasible in terms of allowed, reachable or desired speeds for trains. This KPI is not considered within the scope of this research, but is something ProRail could look into in the future. The solution can also be investigated in terms of how many trains are left with a significant amount of delay, even after rescheduling, so that crew needs to be rescheduled or the train needs to be cancelled. However, since this research only analyses disturbances and not disruptions, it is assumed no cancelling or crew rescheduling is necessary.

### 3.4.2. KPI selection

From Table 3.2, a selection is made to evaluate the decision support system of this research. These KPI's can provide useful insight in the performance of the system and are presented in Table 3.3. This section will describe the motivation and relevance of the selected Key Performance Indicators. In short, these KPI's each have a relation to one important aspect of railway conflict resolution; being real time traffic management, large scale system design, conflict resolution methods or railway passengers.

| Computation performance | Solution quality |
|---|---|
| Computation time | Total propagated delay |
| | Number of affected trains |

Table 3.3: Selected KPI's to evaluate a conflict resolution method

**Computation time**

As explained, conflict resolution algorithms are mainly designed to operate in real time traffic management solutions. In real time applications, computation time is an important Key Performance Indicator when designing new methods. In these situations, feasibility could even be favoured over optimality and solutions need to be generated fast. The quicker the algorithm, the sooner the train driver can receive directions. This KPI is used in a large variety of literature papers [39, 45, 38, 13, 3], highlighting the importance of this Key Performance Indicator.

**Total propagated delay**

When designing conflict resolution methods for railway operations, the most prominent KPI in all research is the total delay (before and after rescheduling). Eventually, from a dispatching point of view, minimizing the delay is the primary goal. Additionally, minimizing the total delay has its secondary benefits in the form of optimized efficiency or a lower degree of change with respect to the original schedule, which can be related to passenger satisfaction. This KPI is thus used in many research papers [16, 18, 12, 11, 10, 45, 41].

In this research, the KPI related to delay will be presented in the form of “delay reduction”. The reduction of delay will be measured with respect to not running a rescheduling algorithm and letting the railway network rely on its buffertimes. This is called the unrescheduled case. Buffer times are time supplements in the timetable that can prevent delay propagation or allow trains to catch up with their delays between stations. In the unrescheduled case, it is assumed that safe distance will remain respected, because of the functioning of the moving block signalling sytem, and conflict will therefore be resolved eventually. However, it expected to take much longer and to result in higher propagated delays with respect to taking rescheduling measures.

**Number of affected trains**

In collaboration with ProRail, it has been established that looking at the number of affected trains in a rescheduling solution can be a very valuable KPI. This has multiple reasons. First of all, it can manage passenger satisfaction. Trains always drive according to a fixed schedule, and the less trains affected the more trains still drive regularly, which is pleasant for train passengers. Second of all, the amount of communication can be kept at a manageable level if the number of affected trains does not exceed certain limits, which in turn can be beneficial for the information loop and its corresponding communication delays, as was presented in Figure 3.1.

### 3.4.3. Performance of state of the art methods

In Chapter 2, state of the art methods for conflict resolution have been elaborately addressed. This section will provide a brief analysis on how approaches in current works perform regarding the KPI's described in the previous section.

First of all, it should be noted that different researches have different focuses and approaches. This analysis does not serve as a guideline for some ideal situation. Additionally, it is very difficult to make one on one comparisons between literature and this thesis. However, it can provide useful insights in current trends, recurring phenomena or order of magnitude of values.

Regarding computation time, the majority of current researches manages to solve conflict resolution problems within a maximum of two minutes. In [16] computation times can even be less than a second. In the works of [18], [10] and [45], computation times lie around 20 seconds to one minute. In [59], computation times are somewhat higher, around 200 seconds. The authors still consider the approach suitable for real time applications. In addition to this, literature frequently mentions the phenomenon of computation times increasing for larger networks or higher situational complexity [38]. This was already discussed in Section 2.3 as well.

Looking at propagated delay, current literature either stresses the maximum delay, the average delay, the total delay or a combination of any of those. In the works of [41] and [45] rescheduling measures can lead to up to 50-60% of situation improvement. In the work of [16], both the average and maximum delay are analysed, where the average delay reduction is significantly higher than the reduction of maximum delay. Of course, this can be influenced by the research objective. For example, in the work of [11], the amount of delay that is reduced varies very highly per train type, since priority rules and hierarchy are used to reschedule trains. The amount of reduced delay thus depends on the goal of the rescheduling and can also be influenced by what type of rescheduling measures are used.

The work of [16] shows clear overviews of the number of affected trains. Overall, around 10% of the trains in their considered network are affected by the rescheduling measures. In another work of the same author, around 40% of the trains in the network are affected. However, the number of trains affected by the measures really depends on the number of trains in the network, the size of the network considered, the type of measured applied and the amount of buffertime in the schedule.

## 3.5. Conclusion

In this chapter, research questions 2a-d have been answered. In conclusion, it can be said that a moving block signalling system is characterized as a a radio-based dynamic supervision system of train braking curves which migrates all vital track-side safety devices on board to reduce maintenance costs while increasing capacity, traffic fluidity, and inter-modal competitiveness. Within this supervision system, a decision support system can be designed to perform conflict resolution. In order to do this, the following considerations are respected. First, in terms of general railway considerations, it is assumed to be working with a microscopic approach on unidirectional tracks and only considering passenger trains. Second, in terms of conflict resolution considerations, this research considers minimizing the secondary delay caused by disturbances. Within conflict resolution, the construction of blocking times within moving block signalling systems is respected. In terms of Key Performance Indicators, the conflict resolution model will be evaluated on computation time, total delay, and number of affected trains.

# 4

# Definition of a centralized conflict resolution model for moving block rail operations

This chapter will analyze and describe how conflict resolution under moving block signalling can be formulated mathematically. First, multiple possibilities for conflict resolution modelling will be assessed. This is followed by a mathematical representation for scheduling problems, namely job shop scheduling. This formulation is then extended to an alternative graph model, which will provide important theory behind rescheduling for moving block systems. Once this is completed, the next section will describe the assumptions and considerations to be made as well as all the parameters that should be taken into account when formulating a mathematical model. Lastly, the model will be presented as an optimization problem by its objective function and corresponding constraints. The chapter will end with a short conclusion, where subquestion 2e is answered.

## 4.1. Conflict resolution methods

When developing a conflict resolution model, there are multiple possible options to be considered. As has been explained in Chapter 3, conflict resolution focuses on mitigating secondary delay, in other words delay propagation, of trains. In current literature, as can be seen in Table 2.3 in Chapter 2, there are two main approaches used to represent conflict resolution problems:

- MI(N)LP - Mixed Integer (non) Linear Problem
- Alternative graph based job shop scheduling problem

Table 4.1 shows characteristics of these two approaches with respect to input parameters, objectives, and solution methods. The main difference between the two methods is the fact that MILP formulations can have a variety of (multi)objectives, whilst alternative graph based job shop scheduling problems

| | AG based JSSP | MIL(N)P |
|---|---|---|
| Input | train timetables<br>train routes<br>processing times<br>initial delay | dependent on objective |
| Objective | minimize maximum propagated delay | variable, multi objective possible |
| Solvable with | commercial solver, (customized) algorithms, heuristics | commercial solver, (customized) algorithms, heuristics |

Table 4.1: Characteristics of two possible modelling approaches

have an objective that is directly linked to the minimization of secondary delay. For this thesis, the choice has been made to formulate the problem as an alternative graph based job shop scheduling problem. All details of this model formulation will be provided in the upcoming sections. There are six reasons to be named that support the decision for this formulation, which are the following:

- This model is a State of the Art method used in relevant papers from Table 2.2 [11, 10, 16, 18, 45, 41]
- This model has been implemented in the European COMBINE and COMBINE 2 projects, which form an important inspiration for this thesis
- This model is very suitable for rescheduling problems because it directly links to delay [11]
- This model shows the potential to be tailored to moving block signalling systems by adjusting existing formulations for fixed block signalling systems
- This model is behind the current TMS (Traffic Management System) used by ProRail, developed in collaboration with OnAir
- The job shop scheduling problem extended to moving block operations can be an inspiration to other application fields where scheduling problems are deemed relevant

## 4.2. The job shop scheduling problem

Train scheduling, and therefore also rescheduling, can be represented as a job shop scheduling problem [16]. As already briefly explained in Chapter 2, in the classical job shop problem the goal is to schedule a set of jobs to be performed on a set of machines. Constraints are used to further shape the problem, expressing if jobs need to be processed by specific machines, if there is a hard or flexible time limit and whether there is a predefined order in which jobs need to be processed. The traditional objective function of a job shop scheduling problem is to process the entire set of jobs as fast as possible [2].

Train rescheduling problems in literature have been formulated as a job shop scheduling problem in a fixed block signalling system [16, 18, 11, 10]. This thesis expands this representation to a moving block situation. Table 4.2 shows an overview of the job shop problems in both the traditional way and applied to fixed and moving block railway systems.

| | Traditional job shop | Job shop for railway scheduling (Fixed Block) | Job shop for railway scheduling (Moving block) |
|---|---|---|---|
| To be processed | Jobs | Trains | Trains |
| To perform processing | Machines | Track sections of predefined length | Infrastructure locations of infinitesimal length |
| Safety | Machine processes one job at a time, job can start after safe interval | Track section processes one train at a time, train can enter after safe interval | Infinitesimal locations processes one train at a time, separated by a safe interval |
| Processing time | Processing time | Running time | Clearing time |
| Order | Job order can only change between machines | Train order can only change between track sections that contain switches or stations | Train order can only change between track sections that contain switches or stations |

Table 4.2: Overview of job shop representations for scheduling problems

As can be seen in Table 4.2, a job shop scheduling problem in fixed block railway applications defines trains as jobs and track sections as machines. The constraints ensure that every track processes one

train at a time. This satisfies the fixed block safety system, or blocking constraint, by allowing only one train per block section. The order of trains can only change in track sections where there are switches, eg switches on the tracks or stations. The time it takes to process a job is the running time of a train through a section. Figure 4.1 shows a visual representation of the job shop scheduling problem for fixed block signalling. It is shown that every machine is a track section between two signals of predefined length $L$ and every train is a job.
When a moving block signalling system is assumed, the job shop scheduling components become slightly different. The jobs are still trains, but since there are no fixed track sections anymore, this research defines machines in a new manner. This new approach can be found in Table 4.2 and is supported by the visual in Figure 4.2. A predefined length $L$ can not be given to tracks in a moving block signalling system, since there are no longer physical sections that can only process one train at a time. Therefore, a machine can be represented by an infinitesimally small track section that can process one train at a time. This means that the processing time for a job is no longer the running time through a section, but the time it takes for a train to pass its own length. This is defined as the clearing time of a train. This nicely supports the principle of moving block signalling, since the next train can start passing a point as soon as another train has completely passed (plus a safety margin). This means trains are separated by only a headway interval, and no longer an entire fixed section length. This newly defined principle of the job shop scheduling problem will be used in the rest of this chapter.

Figure 4.1: A visual representation of the job shop scheduling problem for fixed block signalling systems

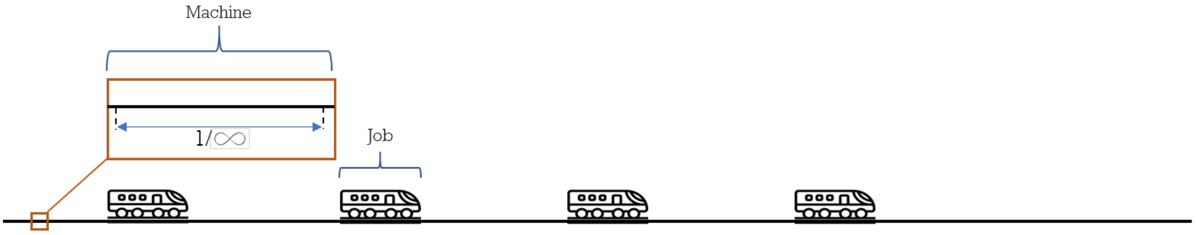


Figure 4.2: A visual representation of the job shop scheduling problem for moving block signalling systems

## 4.3. The alternative graph model

A job shop scheduling problem can be modelled with alternative graph theory. Alternative graphs have been used in recent papers concerning railway scheduling [41, 16, 18, 11, 10, 58, 59]. Alternative graphs are a state of the art method, considered to be very suitable for railway job shop scheduling problems. As described in [10], ”the main value of this formulation is the detailed and flexible representation of network topology and signalling system”. The approach has been briefly addressed in Chapter 2. This section describes alternative graph modelling and the relevance to this research more elaborately. First, traditional alternative graph modelling is explained, followed by its extension to moving block signalling systems.

### 4.3.1. The principle of alternative graph theory

Alternative graph theory is an extension to disjunctive graph theory and one of the first papers to explain this concept is [16]. When this approach is applied, the entire system is represented as nodes and two types of arcs, namely fixed arcs and alternative arcs. Each node in the graph represents the starting time of an operation, for example entering a track section, station or junction. Each fixed arc in the graph represents the precedence relation between two operations, where the weight of the arc implies the processing time of the operation. Figure 4.3 shows a simple visualization of this principle.

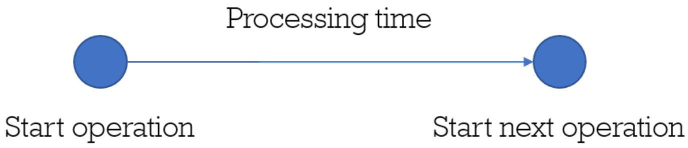


Figure 4.3: A simple visualization of representing train operations as a graph

Besides fixed arcs, alternative arcs are added to the graph. Alternative arcs play a crucial role and are very useful when it comes to conflict resolution for the following reason. If two trains, say train A and train B, require the same resource simultaneously, a conflict is detected. Alternative arcs are placed between the end time of this operation for train A and the starting time of this operation for train B and vice versa. Only one of these arcs is chosen, meaning that only one of these trains can start the operation first. In other words, there is now a precedence relation between the ending of the operation for one train and the start of the operation for another. The weight of the alternative arc represents the headway time between the two trains, ensuring the starting time of the operation by train B is a safe interval away from that of train A, or the other way around.

In a practical example, say train A and train B are required to pass a junction, see Figure 4.4 (let's assume train A = T1, train B = T2). The graph consists of all the operations that need to be performed for every train. This means train 1 passing the section entry, represented by node $i$, and passing the section exit, represented by node $k$. Note that the operation represented by node $k$ can also be described as entering of the next section. These same operations are valid for train 2, represented by node $j$ and $h$. The alternative arcs between node $k$ and $j$ and between node $h$ and $i$ now ensure that only one train can go first, and the other train has to wait for a safe time interval of $a_{ij}$ or $a_{ji}$. Figure 4.5 shows train A going first, Figure 4.6 shows train B going first. Eventually, the goal is to choose one arc over another. How and why this choice is made will be elaborated in the next section.

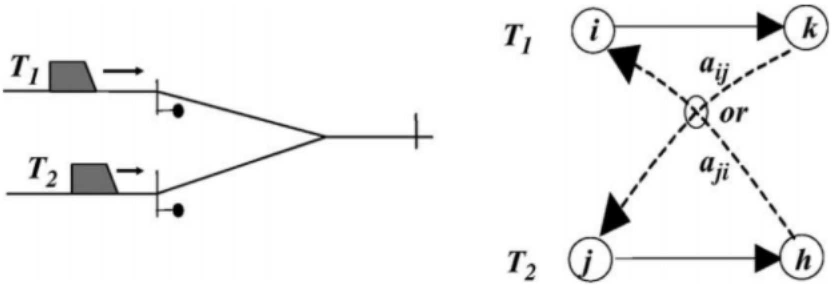


Figure 4.4: Alternative graph approach for two trains approaching a junction, extracted from [16]

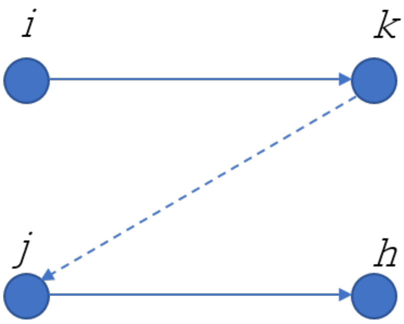


Figure 4.5: Solution to the alternative graph in Figure 4.4 with train A preceding

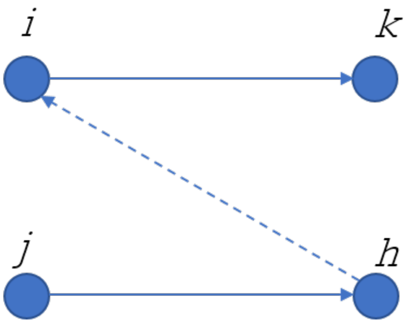
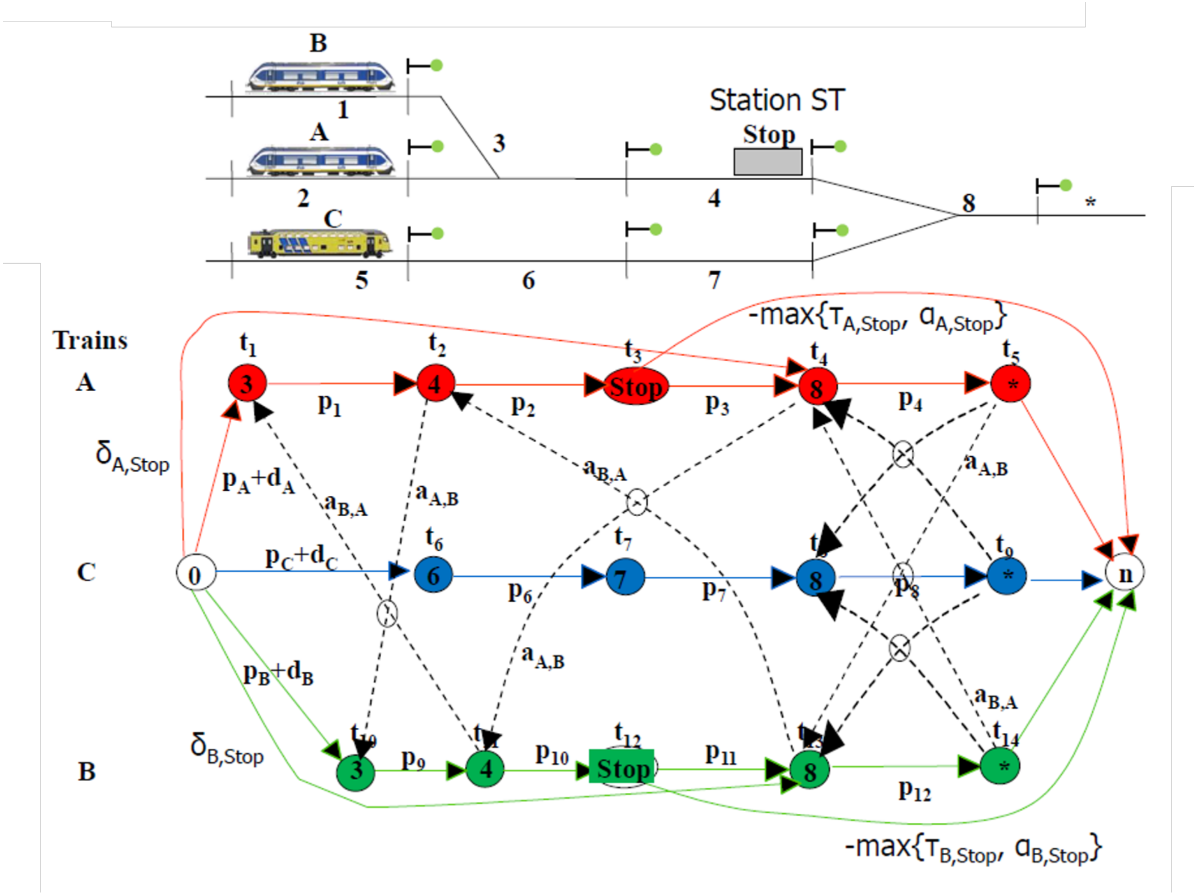


Figure 4.6: Solution to the alternative graph in Figure 4.4 with train B preceding

### 4.3.2. Conflict resolution and delay minimization with alternative graph theory

The alternative graph formulation thus consists of nodes, fixed arcs, and alternative arcs, representing all operations that need to be processed to move trains through the network. Since the goal of the job shop scheduling problem is to process all jobs in the shortest possible time, this means moving all trains from source to destination in the shortest possible time. This can be done by minimizing the secondary delay in the network. The less delay is inherited by a train from other trains, or even no delay at all, the sooner their destination will be reached. But, all this should be done without any conflicts. Let's consider Figure 4.7, a slightly more complicated example.



t = arrival time at node, d = initial delay, p = processing time, a = headway time

Figure 4.7: Alternative graph formulation for three trains approaching junctions and a station

Figure 4.7 shows three trains in a network that have overlapping routes at some point. Train A and B approach a track convergence guarded by a switch, after which they will approach a station. Subsequently, the track converges again at a point that is also approached by train C. The goal is to guide these trains through the network conflict free, whilst minimizing the propagated delay.
The alternative graph that represents this situation shows all operations of every train as nodes and all the precedence relations as arcs. As can be observed in Figure 4.7, alternative arcs are placed between all points of potential conflict. For trains A and B, this includes every operation starting from the entrance of section 3 until the end of the considered network. For trains A and C this is only the convergence at the entrance of section 8, as is the case for trains B and C.
As is visible in the figure, two other important elements of alternative graph theory are introduced here, node $0$ and node $n$, which are described below. It should be noted that the graph elements related to the stop are not explained yet, but will be later in this chapter (Section 4.4.2).

- Node $0$. This node is added for modelling purposes and represents the moment that trains enter the considered network. The edge weight of the edge between node $0$ and the first train operation is the initial delay plus the entrance time.
- Node $n$. This node is added for modelling purposes and represents the maximum secondary delay in the network.

The graph can thus be described as:

$$G(N, F, A) \tag{4.1}$$

where

$N$ = set of nodes
$F$ = set of fixed arcs
$A$ = set of alternative arcs

Figure 4.7 thus represents trains A, B and C moving from node $0$ to node $n$ in the graph $G$. In alternative graph theory, the longest path from node $0$ to node $n$ is equal to the maximum value of propagated delay in this network. Thus, the objective is to minimize this longest path, which is equal to:

$$min \quad t_n - t_0 \tag{4.2}$$

This means that alternative graph theory relies on the following principle:

*Find the subset $S$ of $A$ that minimizes $t_n$*

The optimal sequential choice of alternative arcs will thus yield the desired train rescheduling. This illustrates once more why the alternative graph formulation is linked directly to delay minimization and why it is suitable for conflict resolution.

## 4.4. Alternative graph theory for a moving block signalling system

Since this thesis assumes a moving block signalling system, attention is now devoted to representing moving block operations as an alternative graph. This is a novel approach, designed for this thesis. As is visible in Table 4.2, in the moving block job shop problem there are no predefined beginning and end points of fixed track sections throughout the network. This means that nodes representing the starting time of a train entering a track section are no longer valid. However, trains still pass stations, junctions, and switches, that can be linked to the timetable and a physical component of infrastructure. This leads to an important aspect of the alternative graph formulation for a moving block signalling system: part of the nodes represent physical components of the infrastructure, part of the nodes represent moving points on a train. This could also be seen as part of the nodes representing operations performed by machines of length $L$ and part of the nodes representing operations performed by machines of infinitesimal length. An elaborate description of an alternative graph under moving block operations will be explained in this section.

### 4.4.1. Model components

**Nodes**

The graph formulation will be extended with *an additional node type, called a virtual node,* and thus consists of the following types of nodes:

- **Static node** $N_s$ Nodes related to physical points from the infrastructure with dimensions, like switches. They represent operations performed by machines of fixed length L.
- **Virtual node** $N_v$ Nodes related to the driving trains, that represent operations performed by machines with infinitesimal length.

**Arcs**

Since safety in a moving block signalling systems is maintained via constant headway preservation, this safety aspect needs to be respected in the mathematical approach as well. This leads to another important addition to the alternative graph formulation, namely *a third arc type, called a conditional arc.* This extends the formulation in current literature and splits the set of alternative arcs, denoted $A$, into a set of *decisional alternative arcs*, denoted $D$, and a set of *conditional alternative arcs*, denoted $C$. Since “decisional alternative arcs” are similar to and have the same function as what is denoted “ alternative arcs” in the state-of-the-art alternative graph method, this thesis may denote decisional alternative arcs in that same manner. Conditional arcs are always explicitly noted to be “conditional”. The graph thus consists of the following types of arcs:

- **Fixed arc between two static nodes** $(i, j) \in F$ Arc representing the precedence relation between two operations
- **Decisional alternative arc** $(i, j) \in D$ Arc representing the order in which two trains pass a point in infrastructure
- **Conditional alternative arc** $(v, w) \in C$ Arc representing the order in which two trains drive on a track, following the determined order of the last passed point of conflict and thus obeying the alternative arc decision

**Arc weights**

The addition of new node and arc types has an impact on the different types of arc weights. It should be noted that for readability purposes and alignment with the model description as in current literature, the headway between two trains in a static section is denoted with $a$. The headway related to conditional arcs is denoted $c$. The weights of the arcs are as follows:

- **Weight of an arc between two static nodes** $f_{ss}$ will be defined by the running time of a train in a fixed section
- **Weight of an arc between two virtual nodes** $f_{vv}$ will be defined by the clearing time of a train passing an infinitesimal point and an added safety margin
- **Weight of an arc between a static and a virtual node** $f_{sv}$ will be the running time from the last passed static node to the first generated virtual node
- **Weight of** $a_{ij}$ will be defined by the headway between two trains passing a static section
- **Weight of** $c_{vw}$ will be defined by the headway between two trains proceeding after a static section

Figure 4.8 shows a visual representation of two trains on a track in a moving block signalling system and its corresponding alternative graph. The head and tail of a train, including a safety margin, are represented by virtual nodes (blue). Train A and train B are pursuing the same route from $0$ to $n$, whilst keeping a safe distance from each other. The arc weight $f_{vv}$ is indicated in the figure and is equal to the clearing time of the train, eg the time to pass an infinitesimally small location with its entire length. The safe distance is represented by conditional arcs with weight $c$. Note that in this situation, the order of the trains is already fixed, and that the conditional arcs obey this order and have a safety function.

Figure 4.9 shows two trains on a converging track and it’s corresponding alternative graph. In this figure, a fixed section is present, namely the switch that the trains have to pass. This switch is represented with static nodes (yellow). The fixed arc weight $f_{ss}$ between the static nodes is the running time needed to pass the fixed section. In order to pass the switch, one of the trains will precede the other. Therefore, one of the alternative arcs with arc weight $a$ must be chosen. After the switch, the safe distance needs to be maintained, hence the virtual nodes after the fixed section are separated by conditional alternative arcs again, with arc weight $c$. These conditional arcs will obey to the choice of alternative arc once this has been established.

Figure 4.10 shows two trains on a diverging-converging track and it's corresponding alternative graph. Up to the divergence, the train order is as has been on the straight track. When the track converges again, a new train order can be chosen.

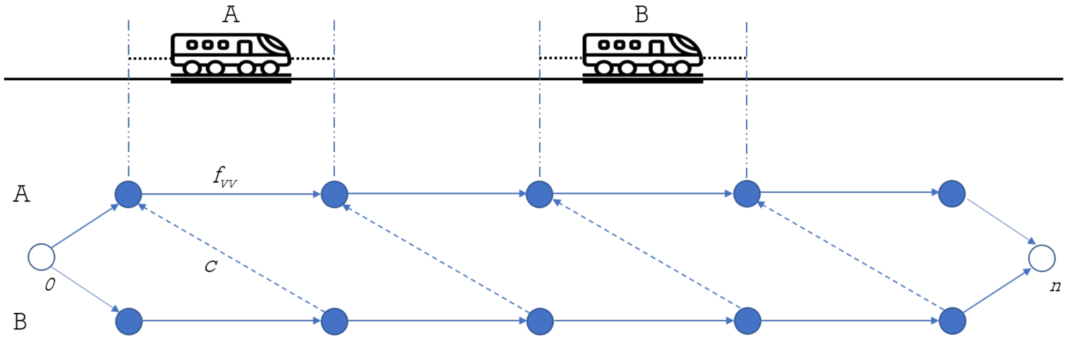


Figure 4.8: The visual alternative graph formulation for two trains on a single unidirectional track under moving block signalling

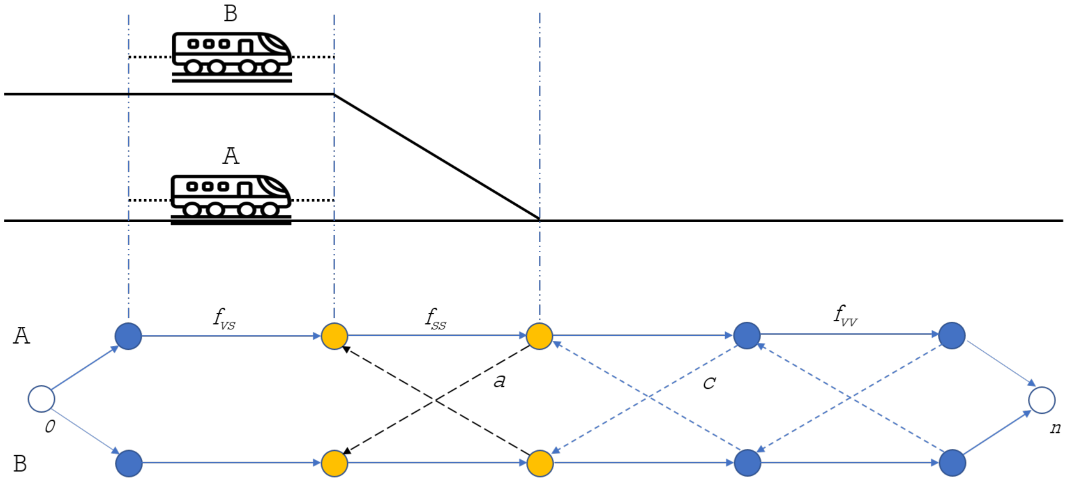


Figure 4.9: The visual alternative graph formulation for two trains on a converging track under moving block signalling

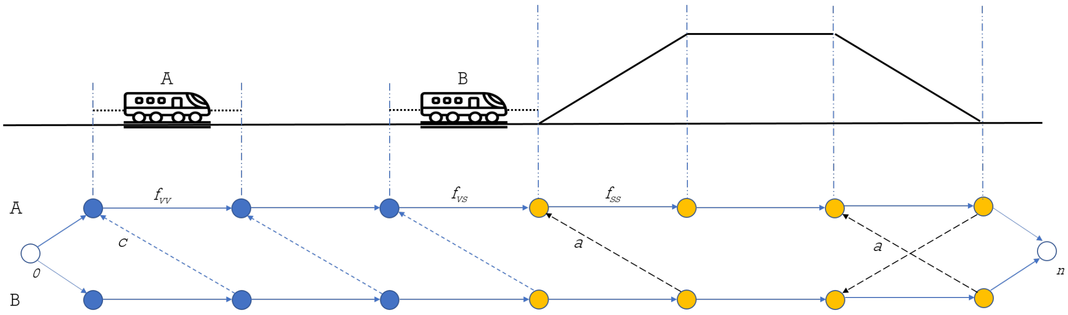


Figure 4.10: The visual alternative graph formulation for two trains on a diverging-converging track under moving block signalling

### 4.4.2. Station modelling

When dealing with conflict resolution and the minimization of secondary delays, stations play a crucial role in the network. In between stations, trains can inherit delays from other trains or recover some of their propagated delay. However, the actual arrival or departure delay at the station is of high importance. This section will explain how stations are modelled.

All trains have scheduled arrival and departure times, which may not be met if a conflict has occurred. Arrival times are modelled as a fixed arc from the node that represents train arrival to node $n$. The weight of this arc is equal to $-max(p_i, \tau_i)$, where $p_i$ is the scheduled arrival time of train $i$ and $\tau_i$ is the earliest possible arrival time of train $i$. In that way, if a train arrives at the final station exactly on time, the delay is equal to zero, but if a train arrives later than the scheduled time, the delay is taken into account. It should be noted that in a properly designed timetable, $p_i$ is always larger than $\tau_i$, since a timetable where the earliest possible arrival time is later than the scheduled arrival time is undesirable. In this work, it assumed that timetables are properly designed and $max(p_i, \tau_i)$ is equal to $p_i$.

Regarding departure times, trains are not allowed to leave from a station earlier than there scheduled departure time. This is modelled as a fixed arc from node $0$ to the node that represents train departure. The weight of this arc is equal to the scheduled departure time.

The longest path from node $0$ to node $n$ is influenced by these timetable arcs. A visualization of station modelling is given in Figure 4.11

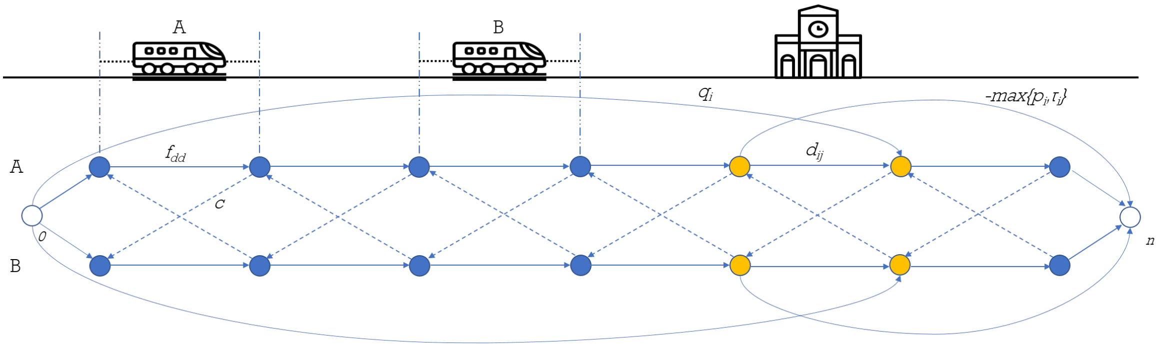


Figure 4.11: Station modelling in the alternative graph formulation

### 4.4.3. Arc weights

This section will provide a more detailed description of how arc weights are obtained. Before discussing all arcs separately, it should be made clear that there is a distinction between:

- **Scheduled time**. This is the processing time for operations as originally planned. Scheduled times always have some buffer to ensure timetable stability [13].
- **Actual time**. This is the processing time for an operation as obtained by the speed and position measurements on the train.
- **Minimum time**. When rescheduling trains, it can be beneficial to alter, usually decrease, the originally scheduled processing times for quicker delay recovery. However, there are still minimum processing times that need to be respected.

These three terms will be used throughout the rest of this section when explaining arc weights.

**Fixed arcs: clearing times**

The time it takes for a train to pass an infinitesimal infrastructure location can be obtained using train speed $v_t$ and train length $l_t$. The clearing time $t_c$ of a train is equal to:

$$t_c = \frac{v_t}{l_t} \tag{4.3}$$

The speed $v_t$ can vary, depending on whether the calculated clearing time is scheduled, actual or minimal. When constructing the alternative graph minimal clearing times, possibly dependent on the actual clearing times, are used. This will give trains the chance of catching up with delays. It is assumed that trains have an implemented buffer time of 5%, meaning the minimum clearing time is the scheduled clearing time minus 5%.

**Fixed arcs: running times**
The time it takes a train to travel through a static section can be obtained using the train speed $v_t$, train length $l_t$, and section length $l_s$. It is important to note that a static section becomes available for next train once the previous train has completely left the section. This means that the clearing time of the section exit node should also be taken into account. A running time $t_r$ is constructed as:

$$t_r = \frac{v_t}{l_s + l_t} \tag{4.4}$$

When constructing the alternative graph, minimum running times are used for the same reasons as described above. A buffer time of 5% is once again taken into account.

**Fixed arcs: dwell times**
In general, the time a train is held at a station can be obtained according to comfortable times for passenger entrance or connection times between trains. It may also include coupling and decoupling of rolling stock units [12]. Therefore, dwell times can widely vary and can be dependent on preferred timetable stability or passenger flows. In this thesis, connection constraints for transfer between different trains are not taken into account. The authors of [12] hold a minimum dwell time at stations of 50sec when not taking into account transfers, which is considered suitable for this thesis as well. In the construction of the alternative graph, minimum dwell times are used, because this can enhance delay recovery.

**Alternative arcs and conditional arcs: headway times**
Headway times are dependent on train speed, braking properties, and situational factors like the reaction time. Figure 4.12, extracted from [50] shows headway distances in moving block signalling systems. A safety margin (Sm) is established from the rear end of the leading train (supervised location, SvL) and current speed and position measurements are used to calculate a suitable braking curve up to the end of authority (EoA). This thus establishes the movement authority (MA) for the train [50]. All details on headway variation and optimization are related to train control, which is not within the scope of this thesis, but for more information see [50]. When constructing the alternative graph, the arc weights of the alternative and conditional arcs will be minimum headway times. This will enhance quick delay recovery, making use of the maximum railway capacity by allowing trains to drive as close together as is safely possible.
These minimum headway times vary, depending on whether it is a headway arc between fixed nodes or virtual nodes. In Chapter 3, the difference between fixed and moving block headways has been explained using Figures 3.4 and 3.5. To complete the mathematical formulation, these headways are constructed as follows. First, for headway between two fixed nodes:

- Setup time ($t_s$) = 7 seconds
- Sight reaction time ($t_{sr}$) = 6 seconds
- Approach time ($t_{ap}$) = time to cross braking distance + 50 meter safety margin
- Running time ($t_{run}$) = running time over a length of 100m (assumed length of a fixed section)
- Clearing time ($t_c$) = clearing time as described above
- Release time ($t_{rel}$) = 3 seconds

Second, for headway between two virtual nodes:

- Sight reaction time ($t_{sr}$) = 6 seconds
- Approach time ($t_{ap}$) = time to cross braking distance + 50 meter safety margin
- Clearing time ($t_c$) = clearing time as described above
- Release time ($t_{rel}$) = 2 seconds

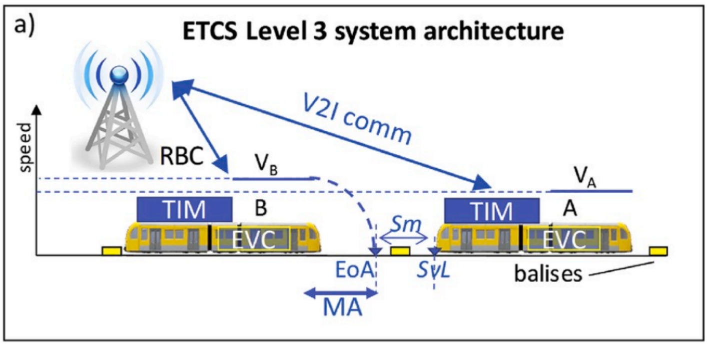


Figure 4.12: Illustration of the headway between two trains in a moving block signalling system.

### 4.4.4. Generation of virtual nodes

The addition of a virtual node to the alternative graph formulation alters the way the graph is constructed. When using static nodes only, the entire graph can be constructed beforehand, since infrastructure components do not move. However, virtual nodes are train related and therefore dependent on the speed and positions measurements of the signalling system. This means that these measurements need to be used to generate virtual nodes over a certain time period and its corresponding clearing times. Because of the novelty of this mathematical approach, this research will not yet provide in depth construction of how this node generation will computed exactly. This is left as a recommendation for future research.

Nonetheless, this section provides a simple example of how this node generation could be performed. A visualization of this is given in Figure 4.13. Say train A is pursuing its route. The moving block signalling system detects train A's speed and position. The length of train A is known. With this knowledge, TMS can generate virtual nodes that are at least a train length apart plus an appropriate safety margin for the measured operational properties. Additionally, it can use the measurements to compute the clearing time of the train, in order to create the correct arc weight between those two nodes. The virtual nodes, taking into account their appropriate safety margins and their predicted arc weights are then placed between the current position of the train and the end of considered network for generation, for example the nearest station like in Figure 4.11.

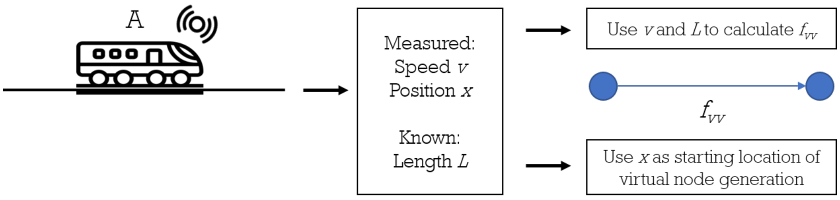


Figure 4.13: A simple example of virtual node generation using speed and position measurements and train properties

### 4.4.5. Modelling assumptions

In order to be able to use this model for implementation, several assumptions need to be made. This subsection describes assumptions related to general aspects, static node sections, and station modelling. Each assumption will be briefly explained.

#### General

It is assumed that:

- the original timetable is known
- all information to calculate arc weights is known, which includes:
  - train length
  - train speed profile
  - train route

**Static node sections**

For this graph formulation, it is assumed that the following infrastructural assets will be modelled as static nodes:

- switch, but also components that exist of multiple switches, such as:
  - station
  - junction

In addition to this, the following nodes in the graph formulation will be approached as a static node:

- node $0$
- node $n$

All operations between two static nodes will be approached as a fixed section.

**Station modelling**
Since rerouting is not part of the rescheduling measures in this work, station stops are modelled as static sections belonging to the corresponding train route. Other parts of the station, not planned to be used by the considered trains, are not taken into account.

## 4.5. Problem formulation

This section presents the objective function and corresponding constraints, together forming the mathematical model that represent conflict resolution under moving block signalling, based on [16, 18, 11]. For each equation, a short explanation is provided to elaborate on the contribution to the model.

**Indices and sets:**

$G(N, F, A)$ = graph

$N$ = set of nodes
$F$ = set of fixed arcs
$A$ = set of alternative arcs

$i, j...$ = indices for nodes
$s$ = index for static node
$v$ = index for virtual node
$\theta$ = index for trains

$N_s \subset N$ = set of static nodes
$N_v \subset N$ = set of virtual nodes
$F_{ss} \subset F$ = set of fixed arcs between two static nodes
$F_{sv} \subset F$ = set of fixed arcs between a static and a virtual node
$F_{vv} \subset F$ = set of fixed arcs between two virtual nodes

$D \subset A$ = set of decisional alternative arcs
$C \subset A$ = set of conditional alternative arcs

**Parameters:**

$f_{ij}$ = weight of fixed arcs (running time or clearing time)

where:
$f_{i_s j_s}$ = running time
$f_{i_s j_v}$ = running time (to nearest virtual node)
$f_{i_v j_v}$ = clearing time

$a_{ij}$ = weight of decisional alternative arcs
$c_{ij}$ = weight of conditional alternative arcs

where:
$a_{ij}, c_{ij}$ = headway time

$p_\theta$ = scheduled arrival time of train $\theta$
$q_\theta$ = scheduled departure time of train $\theta$
$d_{ij,\theta}$ = scheduled dwell time of train $\theta$

**Decision variable:**

$t_i$ = actual start of operation at node $i$

**Objective function:**

$$min \quad t_n - t_0 \tag{4.5}$$

**Subject to**:

$$t_j - t_i \geq f_{ij} \qquad \forall \quad (i,j) \in F \tag{4.6}$$

$$t_j - t_i \geq a_{ij} \vee t_k - t_h \geq a_{hk} \qquad \forall \quad ((i,j),(h,k)) \in D \tag{4.7a}$$

$$\text{if} \quad a_{ij} \quad \text{is selected, then} \quad t_w - t_v \geq c_{vw} \qquad \forall \quad (v,w) \in C_{a_{ij}} \tag{4.7b}$$

$$\text{if} \quad a_{hk} \quad \text{is selected, then} \quad t_z - t_u \geq c_{uz} \qquad \forall \quad (u,z) \in C_{a_{hk}} \tag{4.7c}$$

$$t_i \geq q_{i,\theta} \tag{4.8}$$

$$t_j - t_i \geq d_{ij,\theta} \tag{4.9}$$

Where the indices $h, i, j, k, u, v, w, z$ are in the set of nodes $N$ and $\theta$ is in the set of trains $\Theta$.

Equations 4.5, 4.6 and 4.7a in this formulation are based on the mathematical description as by D'Ariano and Corman in many of their works [16, 18, 11]. Equations 4.7b and 4.7c show the contribution of this research, explicitly describing the conditional arcs. The contribution of this work is also represented in the different subsets in the set of nodes $N$ and fixed arcs $F$, as described under "indices and sets".

Equation 4.5 represents the minimization of the maximum arrival time at node $n$, which equals the maximum propagated delay in the network.

Equation 4.6 represents that the starting time of the operation at the next node (node $j$) is at least the starting time of the previous operation (node $i$) plus the processing time of that operation ($f_{i_s j_s}$, $f_{i_s j_v}$ or $f_{i_v j_v}$)

Equation 4.7a represents that only one decisional alternative arc can be chosen. This implies that only one of two trains can start an operation first, as was visually represented in Figures 4.5 and 4.6. For the selected alternative arc, it is stated that the starting time of the operation at node $j$ is at least the starting time of the operation at node $i$ plus the safe time interval (headway) before a next operation can start. The headway is denoted with $a_{ij}$ and $a_{hk}$.

Equations 4.7b and 4.7c show the dependence of the conditional alternative arcs. If one decisional arc is selected, the conditional arc that has the same direction must be selected for all conditional arcs dependent on that decisional arc. This implies the same train order must be maintained after this has been established by the decisional alternative arcs. Additionally, these constraints show that the starting time of the operation at node $w$ is at least the starting time of the operation at node $v$ plus the safe time interval (headway) before a next operation can start, similar to constraint 4.7a. The headway is denoted with $c_{vw}$ and $c_{uz}$. Recall that conditional arcs are placed between trains driving on the same track without any trackside signals and these arc weights thus represent an important safety constraint.

Equation 4.8 represents that the starting time of the operation at node $i$, when the operation is a station departure, is at least the scheduled departure time at node $i$ for train $\theta$. This constraint could be viewed as already included in constraint 4.6, but is written separately to show station departure times as an important component of the model.

Equation 4.9 represents the starting time of the operation at node $j$ (station departure) is at least the starting time of the operation at node $i$ (station arrival) plus the scheduled dwell time of train $\theta$ between nodes $i$ and $j$. This constraint could be viewed as already included in constraint 4.6, but is written separately to show station dwell times as an important component of the model.

## 4.6. Conclusion

This chapter has described how a mathematical model can represent conflict resolution in a moving block signalling system, thereby answering subquestion 4e. A formulation has been obtained using the alternative graph method, consisting of nodes and arcs, which has been extended for moving block suitability. This means adding virtual nodes and conditional arcs as an extension to the state of the art model. Subsequently, an optimization problem can be formulated with the objective of minimizing the arrival time at the final node. The objective function is subjected to four constraints, where the presence of conditional arcs is explicitly stressed in the second constraint. This, in addition to the extended graph properties, illustrates the novelty and contribution of this mathematical model.

# 5

# Verification of a centralized conflict resolution model under moving block signalling

The mathematical model that has been developed and described in chapter 4 is novel. Therefore, this chapter will provide a proof of concept to illustrate the working principle of the model and show that the modelling approach is sufficient to solve the conflict resolution problem. In order to provide for a proof of concept, the mathematical model from Chapter 4 has been implemented in a Python model. A conceptual representation of the working principle of this model is shown in Figure 5.1.
This chapter describes how this implementation has been developed and executed. First, this chapter focuses on the creation of the alternative graph. Second, objective function modelling will be discussed. Third, solution approaches will be described and the method to solve the optimization problem will be presented. Next, the working principle will be illustrated via three illustrative examples. This chapter will close with a model verification of the proof of concept, followed by a conclusion. Overall, this will answer research question 3a-3c.

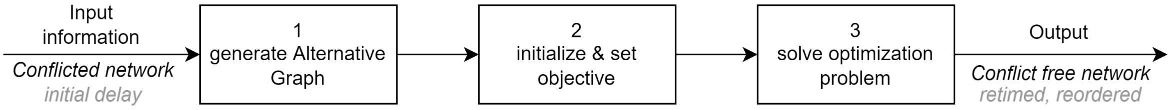


Figure 5.1: Conceptual representation of the rescheduling process

## 5.1. Generating an alternative graph

When implementing the mathematical model, a more detailed look towards the graph formulation is necessary. As explained, the graph consists of nodes, fixed arcs, and alternative arcs. This section will explain what to take into account when generating suitable nodes and arcs for the moving block job shop scheduling problem.

Before diving into the details of the alternative graph, it is important to establish which input parameters are required in order to fulfill a Python implementation of this problem. From the parameters listed in the problem formulation in Section 4.5, it can be derived that the input must contain:

- timeable points
- blocking times
- headway times
- timetable times

- initial delays

Table 5.1 explains the necessity of these input parameters and how they can be used to translate the problem into an alternative graph formulation. More details on how the alternative graph is build is provided in the upcoming subsections.

| Input | Motivation |
|---|---|
| timeable points | These points are points where an operaration is performed, and will thus form the nodes of the graph. Nodes can be distinguished using the names of the timeable points and the name of the train that is passing them. |
| blocking times | Blocking times are very important input variables to aid the alternative graph formulation. They will provide the edge weights of all the fixed edges representing an operation time. |
| headway times | A minimum headway time needs to be given to provide the edge weights of the alternative and conditional arcs. |
| timetable times | The planned arrival and departure times will provide edge weights for fixed edges representing operations at stations (see 4.3 station modelling). |
| initial delay | It should be known whether there are intial delays for one or more trains, which influences the time instant at which they enter the graph. |

Table 5.1: Input parameters to create an implementation of the mathematical model

### 5.1.1. Nodes

Nodes are linked to a unique ID, that can contain information about the operation this node represents. Also, the type of node should be clearly indicated to be static or virtual. Next to the ID and type of the node, there are two more attributes given to the nodes. First, the type of event this node is representing. This is necessary in order to be able to distinguish arriving, departing or passing events throughout the model. Second, the node value is indicated as a node property. The value of a node is equal to the starting time of the operation that is represented by the node. Therefore, this is the decision variable of the model, represented by $t$ in equation 4.5. Nodes are thus implemented in the following manner:

| Object | Node | | | |
|---|---|---|---|---|
| Attributes | Name | Type | Event | Value |
| Possibilities | from input | static<br>virtual | passing<br>arriving<br>departing | t<br>(decision variable) |

### 5.1.2. Arcs

Arcs consist of a beginning and end node, or source and destination, which are important arc defining properties. Of course the arc weights should also be defined and linked to all correct arcs. As explained in Section 3.4.2, arc weights differ according to the node types that connect the arc and can be equal to running time, clearing time, dwell time or timetable arrival/departure time. This will also be indicated in the model and is called the service of the arc. For the modelling approach, it is crucial to indicate whether an arc is fixed, alternative or conditional. Note that the alternative and conditional arcs contain all the same attributes as fixed arcs, and can therefore be created in the same manner. Arcs are thus created in the following way:

| Object | Arc | | | | |
|---|---|---|---|---|---|
| Attributes | Source | Destination | Weight | Type | Service |
| Possibilities | Node object | Node object | from input | fixed | running time |
| | | | | alternative | clearing time |
| | | | | conditional | dwell time |
| | | | | | arrival time |
| | | | | | departure time |
| | | | | | headway time |

### 5.1.3. Trains

For a complete model, it is necessary to create a set of trains. For every train, it is of importance to know which operations need to be performed and what the precedence relations between these operations are. That means that trains consist of a set of nodes and fixed arcs and will be represented as follows:

| Object | Train | |
|---|---|---|
| Attributes | Nodes | Arcs |
| Possibilities | Node object | Arc object |

Table 5.2: Table: *Note: these node and arc objects represent all operations and operation processing times of corresponding train

### 5.1.4. Graph

Eventually, all train properties of all trains will be added to a graph, which thus again consists of nodes and arcs. However, when adding all components in the set of trains to the graph, this will only contain fixed arcs. This means that the alternative and conditional arcs are yet to be created and added to the graph. Alternative arcs should be placed between predicted and possible conflicting nodes, meaning nodes where trains start sharing or share the same route and train order can be changed. Conditional arcs should then be placed throughout the entire route to maintain both train order and safe distance. This will then make the graph formulation complete. Below, a pseudocode is provided for the formulation of the full alternative graph.

```
Graph = Graph
for train in trains:
    add train.nodes to Graph
    add train.edges to Graph
for train in trains:
    if node(train_A), node(train_B) on same route:
        if order change possible:
            add alternative arc pair
        if order change not possible:
            add conditional arc pair
for arc in alternative arcs:
    add arc pair to Graph
for arc in conditional arcs:
    add arc pair to Graph
```

## 5.2. Initializing the model

Before moving towards optimization of the alternative graph, it should be noted that in this proof of concept, the analysed network is approached as one snapshot of the current situation, in which the optimal solution can be found. If this model were to be implemented in practice, it will have to become a dynamic decision support system, that will be activated once a conflict has been detected. This is left as a recommendation for further research.

After converting a set of input parameters to an alternative graph as described in the previous section, the optimization problem can be formulated. First of all, the decision variable $t$ should be initialized. As mentioned, the decision variable $t$ is equal to the value of a node in the graph. Before rescheduling, the node values are equal to their current values (for current position) and current predicted values (for future position) of the train, as measured by the on board monitoring devices on the trains. This means that for every train, the node values are equal to the time instant at the current node plus the planned processing time of every operation. The initialization in pseudocode thus looks as presented below.

```
Initialize node values:
for train in trains:
    for node in train.nodes:
        node.value = node.value(previous node) + edge.weight(between nodes)
```

This means a complete alternative graph including initial node values is now formulated. The goal is to solve the optimization problem where the node value of the final node, $t_n$, is minimized, as has been presented in equation 4.5. The minimization of the final node value is subjected to the constraints presented in equation 4.6, 4.7a, 4.8 and 4.9. This will be more elaborately discussed in the next section.

## 5.3. Solution approaches

In order to find the optimal value for the objective function, the minimum value for $t_n$, a solution approach should be chosen. In Table 2.3 of Chapter 2, three main directions for solving optimization problems can be found:

- (Tailored) algorithm
- Heuristic
- Commercial solver

In the traffic management system (TMS) developed by Mazzarello [41], heuristics were used to find the solution to the alternative graph formulation. This TMS has been developed ever since and connected to the ProRail conflict simulator FRISO. Multiple heuristic possibilities exist within the solver and can be chosen depending on desired outcome of solution. In the future, these approaches can be tailored to moving block rescheduling. However, that is not currently available and it is out of the scope of this thesis to design a heuristic or tailored algorithm. This is left as a recommendation for future research. Therefore, for this research the choice has been made to use a commercial solver for optimization. The choice of commercial solver is Gurobi, which will be elaborated on further in the next section.

## 5.4. Solving the optimization problem with a commercial solver: Gurobi optimizer

Gurobi optimizer is a product released by Gurobi optimization [28] and a widely used commercial solver. Gurobi is designed for MI(L)P problem solving and is supported by multiple programming languages including Python. In order to be able to solve the conflict resolution problem with Gurobi, there are two aspects that need attention. First, choosing a suitable solution algorithm provided by Gurobi. Second, tailoring the mathematical formulation to ensure it matches the Gurobi environment. These aspects will be elaborated in the upcoming subsections.

### 5.4.1. Gurobi solution algorithms

Gurobi provides two main solution algorithms, a simplex algorithm and a barrier algorithm. The algorithms have different characteristics and suitability, of which an overview is provided in Table 5.3. Gurobi first applies a presolve using a branch and cut, and then solves the remaining problem using either of these two algorithms.

| Simplex algorithm | Barrier algorithm |
|---|---|
| suitable for simple models | suitable for large models |
| marginally numerically sensitive | suitable for complex models |
| | numerically sensitive |

Table 5.3: Characteristics of the two main algorithms provided by Gurobi optimizer

Since railway conflict resolution is a quite complex problem, it seems beneficial to use the barrier algorithm. Additionally, when using real life data in a case study, which will be done in Chapter 6, the total problem will become quite large as well. However, this algorithm might complicate the solution process more than necessary and it is more numerically sensitive than the simplex method [29]. Figure 5.2 and 5.3 show the difference in complexity visually. For more information on the working principles of the algorithms, refer to [7, 57].

The choice has been made to initially solve the conflict resolution problem with the simplex algorithm. That is, with the option to switch the commercial solver to barrier algorithm if computation times get to high, for example in the case study. There is also an option to perform a stricter presolve in the branch and cut to bring down computation times. However, trial and error has proven that this performs worse in terms of solution quality than switching to the barrier algorithm.

It should be noted that due to the fact that Gurobi is a commercial solver, the exact properties of their algorithms are not revealed. Therefore, to have full control over the solution procedure, tailored algorithms can be designed to solve the problem, as was already briefly mentioned at the beginning of this section. Tailored algorithms and heuristics are already available for decision support systems suitable for fixed block signalling systems. These could be extended to moving block suitability. However, that is not the scope of this thesis.

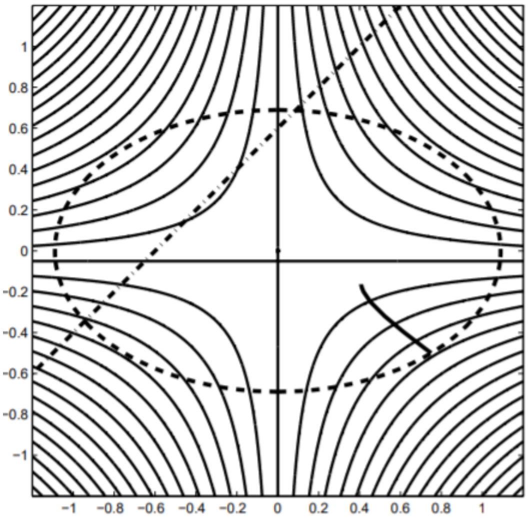


Figure 5.2: A visualization of the working principle of a barrier algorithm, extracted from [57]

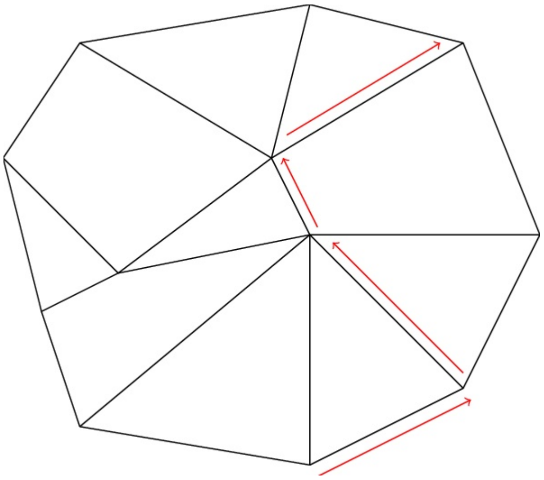


Figure 5.3: A visualization of the working principle of the simplex algorithm, extracted from [7]

### 5.4.2. Model tailoring

Gurobi is a MI(L)P solver and is thus very suitable to solve problems with an objective function as equation 4.5 as presented in Section 4.5. Gurobi can minimize the objective function by finding the optimal values for the decision variables of the optimization problem. Gurobi takes integer and binairy decision variables. The primary decision variable in this optimization problem is:

$t_i$ (integer)

This decision variable, the actual arrival time of a train at node $i$, represents the retiming aspect of the conflict resolution problem. The variable $t_i$ will be optimized to minimize $t_n$.

Next to retiming, the reordering aspect is also represented in the mathematical model. This involves the selection of alternative arcs and the obedience of conditional arcs. In pseudocode, constraint 4.7a can be written as follows:

```
Alternative arc = arc_a
Conditional arc = arc_c
Choose a_ij or a_hk
if arc_a = a_ij:
    arc_c = c_ij
elif arc_a = a_hk:
    arc_c = c_hk
```

However, Gurobi does not take if-statements into its optimization constraints. Therefore, this constraint will be represented by introducing two additional decision variables:

$z_i$ (binary)

$y_i$ (binary)

Decision variable $z_i$ is related to alternative arcs, decision variable $y_i$ is related to conditional arcs. Every pair of alternative (or conditional) arcs will be multiplied with a binary decision variable, that can either be 1 or 0. If an alternative arc is chosen, the corresponding binary value will be 1. Logically, the corresponding value of the other arc will then be 0, meaning only one arc can be chosen. In order to impose this logic into Gurobi solver, two extra constraints are added to the model, stating the sum of binary variables related to one pair of arcs is equal to 1. In simple form, this looks as follows:

$$\begin{aligned} &z_i = [z_{i,i}, z_{i,j}] \\ &headway_{min} \leq a_{ij} * z_{i,i} + a_{hk} * z_{i,j} \\ &sum(z_i) = 1 \end{aligned} \tag{5.1}$$

Besides imposing the choice between alternative arcs, the obedience of conditional arcs also needs to be maintained. In order to ensure that the correct conditional arc is selected, the big M method is introduced. The big M method is a method used in linear programming that is suitable for navigating constraints in the correct direction [36]. The principle of big M can be used to ensure the choice of conditional arc $c_{hk}$ is made undesired after choosing alternative arc $a_{ij}$.
A very large number M is introduced in the model. The choice of binary variable $y_i$ is now dependent on $z_i$ and a large number M in the following manner:

$$\begin{aligned} &M = 1000 \\ &y_i = [y_{i,i}, y_{i,j}] \\ &headway_{min} \leq c_{ij} * (y_{i,i} + M(1 - z_{i,i})) + c_{hk} * (y_{i,j} + M(1 - z_{i,j})) \\ &sum(y_i) = 1 \end{aligned} \tag{5.2}$$

In this way, if $z_{i,i}$ is equal to 1 and $z_{i,j}$ is equal to zero, the term $y_{i,i} + M(1 - z_{i,i})$ becomes $y_{i,i}$, whilst the other term on the same side of equation 5.2 becomes $y_{i,j} + M$. Since the optimization is a minimization problem, choosing the first term will be favoured over the second. This means that if alternative arc $a_{ij}$ is chosen, conditional arc $c_{ij}$ is chosen as well and vice versa. In the same manner as in equation 5.1, the sum $y_i$ is equal to one, meaning only one of the conditional arcs will be chosen.
Furthermore, to ensure that timetable constraints represented by the arrival and departure time arcs

as explained in Section 4.4.2 are taken into account explicitly, the fixed arc constraint is split into three constraints. Overall, this means that the Gurobi implementation with all decision variables, objective function and constraints looks as described below. It can be seen that implemented constraint 1a covers equations 4.6 and 4.9, constraint 1b and 1c cover equation 4.8 and constraints 2a,2b,3a and 3b make up equation 4.7a.

Note that besides the objective function and constraints, several functions have been used in the Python model to aid both the graph formulation and the optimization problem. These can be found in Appendix D.

Decision variables:

$t_i$ = node value of node $i$, type = integer for $node_i$ in $nodes$
$z_i$ = represent arc choice, type = binary for $arc - pair_i$ in $alternative - arc - pairs$
$y_i$ = represent arc obedience, type = binary for $arc - pair_i$ in $conditional - arc - pairs$

Objective function:

$$min \quad t_n \tag{5.3}$$

Constraint 1a:

$$\begin{gathered} t_i - t_j \geq edge.weight \\ for \quad edge \in Graph.edges \quad \text{(fixed, processing times)}, \quad i,j \in Graph.nodes \end{gathered} \tag{5.4a}$$

Constraint 1b:

$$\begin{gathered} t_i - t_j \geq edge.weight \\ for \quad edge \in Graph.edges \quad \text{(fixed, arrival times)}, \quad i,j \in Graph.nodes \end{gathered} \tag{5.4b}$$

Constraint 1c:

$$\begin{gathered} t_i - t_j \geq edge.weight \\ for \quad edge \in Graph.edges \quad \text{( fixed, departure times)}, \quad i,j \in Graph.nodes \end{gathered} \tag{5.4c}$$

Constraint 2a:

$$\begin{gathered} (t_j - t_i) * (z_{i,i}) + (t_i - t_j) * (z_{i,j}) \geq edge.weight \\ for \quad edge \in Graph.edges \quad \text{(alternative)}, \quad i,j \in Graph.nodes \end{gathered} \tag{5.5a}$$

Constraint 2b:

$$sum(z_i) = 1 \tag{5.5b}$$

Constraint 3a:

$$\begin{gathered} (t_j - t_i) * (y_{i,i} + M * (1 - z_{i,i}) + (t_j - t_i) * (y_{i,j} + M * (1 - z_{i,j}) \geq edge.weight \\ for \quad edge \in Graph.edges \quad \text{(conditional)}, \quad i,j \in Graph.nodes \end{gathered} \tag{5.6a}$$

Constraint 3b:

$$sum(y_i) = 1 \tag{5.6b}$$

## 5.5. Illustrative examples

The considerations, formulations and choices from the previous sections ensure that a complete Python implementation can be realised. Three illustrative examples have been formulated in collaboration with ProRail to support the proof of concept. In this section, these illustrative examples will be explained and both the problem and solution for all of these examples will be presented.

### 5.5.1. Two trains on a converging track

The first illustrative example shows two trains on a converging track. This example is also used in a fair amount of literature using the alternative graph approach [16, 18, 10], therefore it is considered valuable to evaluate the same example as often provided in state of the art literature. Additionally, two trains on a converging track is one of the simplest examples to show the principle and contribution of alternative arcs, since it asks for a decision where only one of the two trains can go first. Unlike current literature, the system limit in this example is placed not directly after the track convergence but on the straight track, in order to show the conditional arcs and virtual nodes.

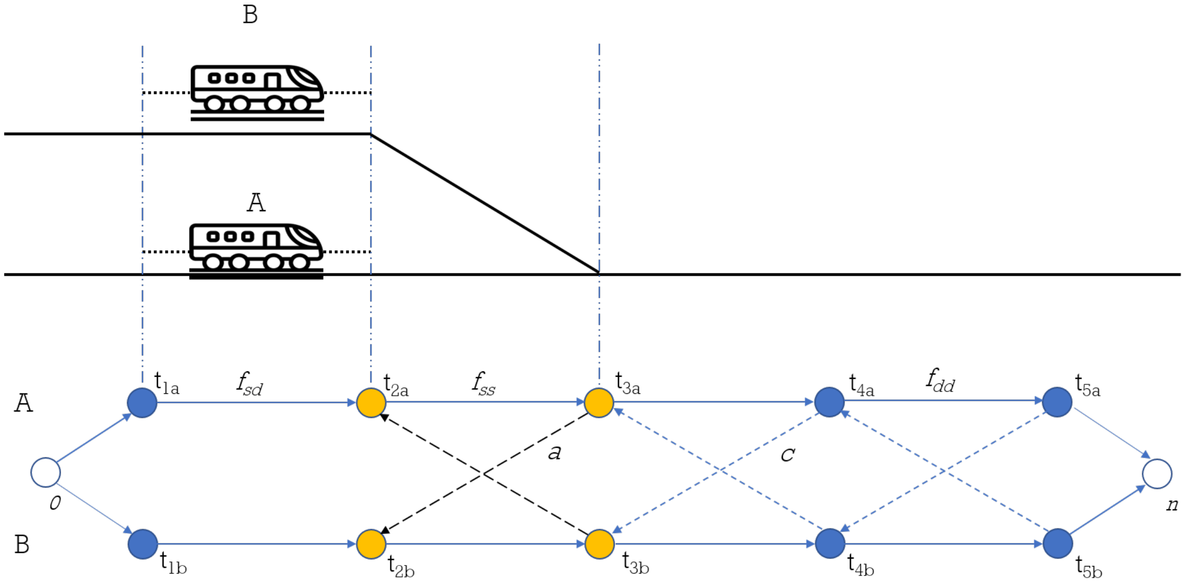


Figure 5.4: Illustrative example 1: two trains on a converging track

The situation of illustrative example 1 can be described as follows. Two trains are approaching a switch that converges the double track to a single track. Because of an initial delay of train A, the two trains approach the switch simultaneously, thereby causing a conflict. Making rescheduling decisions can facilitate an efficient solution to this conflict.

In the illustrative examples, arbitrary block occupation times have been formulated and used as described below. Note that these illustrative example function to show the working principle of the mathematical model and do not contain any real values. Therefore, the units of the used parameters is considered arbitrary as well. In addition, it is assumed that trains have slack time enclosed in their timetable, meaning that in between stations they can catch up with delays. Also, the time $t_0$ at node 0 is set at zero. The input parameters for the Python implementation for illustrative example 1 are as presented in Table 5.4.

$f_{ss}$ = 2
$f_{sv}, f_{vv}$ = 1
$a_{ij}, c_{ij}$ = 0.5
buffertime = 5%

Table 5.4: Input parameters for illustrative example 1

| | Train A | Train B |
|---|---|---|
| Nodes | A, B_A, C_A, D_A, E_A | F, B_B, C_B, D_B, E_B |
| Processing times | 1, 2, 1, 1 | 1, 1, 2, 1, 1 |
| Initial delay | 3 | 0 |
| Headway time | 0.5 | 0.5 |

This problem can now be translated to a graph in the Python script following the model as described in Chapter 4 and the methodology as described in Sections 5.1-5.3. The alternative graph formulation as visualized in Figure 5.4, can now be translated into the Python script.

The scheduled situation, consisting of all scheduled node values, is presented in Table 5.5. For a clearer overview of the example's performance, a final station Z has been added to the nodes. Note that in all tables of this section, bold node letters represent a station. In this example, as described in Table 5.4, train A has been given an initial delay of 3. This alters the scheduled situation to a conflicted situation, with predicted node values as presented in Table 5.6. If this conflict is noted, the moving block signalling system will provide safety and ensure that the safe distance between trains will be reestablished. This has been implemented in the model by alternative and conditional arcs. However, the optimization problem not only regains safety, but minimizes the total propagated delay. Tables 5.7 and 5.8 show the abilities of the optimization problem in combination with the alternative arc model. Table 5.7 shows the arrival time at the final station if no reordering of trains is executed. Table 5.8 shows the arrival time at the final station of reordering has been implemented. It is evident that train reordering reduces the delay propagation. This is made more explicit in Table 5.9, which shows the total delay propagation in different situations. It can be observed that rescheduling in this case leads to a delay reduction of 0.9 (or: 22.8%).

In addition to table representation, Figures 5.5, 5.6 and 5.7 show the rescheduling process as well. Enlarged versions of these figures can be found in Appendix A. In Figure 5.5, the situation without rescheduling measures is depicted. This means that the scheduled train order is maintained and train A will be driving in front of train B. However, the safe distance between trains will be restored, because of the safety measures of the moving block signalling system. This leads to $t_n$ equal to 3.949, following from train B's arrival delay at node Z. Figure 5.6 shows the alternative graph before the rescheduling process, clearly showing that a choice between alternative arcs is allowed. This then leads to the solution in Figure 5.7, which shows train B preceding train A, leading to $t_n$ equal to 3.049. This means train A gains a consecutive delay of 3.049, but the propagated delay is reduced to zero.

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A,F | 0 | 1 | - |
| B | 1 | 2 | A |
| C | 3 | 4 | A |
| D | 4 | 5 | A |
| E | 5 | 6 | A |
| **Z** | 6 | 7 | A |

Table 5.5: Illustrative example 1 - scheduled situation

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A,F | 3 | 1 | - |
| B | 4 | 2 | - |
| C | 6 | 4 | - |
| D | 7 | 5 | - |
| E | 8 | 6 | - |
| **Z** | 9 | 6 | - |

Table 5.6: Illustrative example 1 - conflicted situation

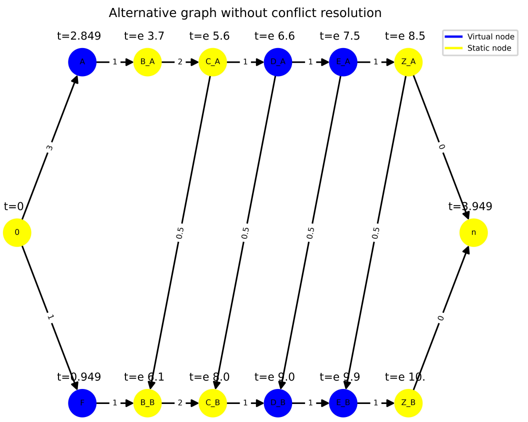


Figure 5.5: The graph representation for illustrative example 1 if no rescheduling is applied

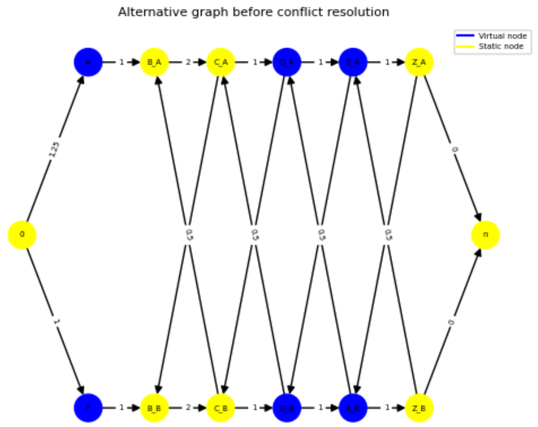


Figure 5.6: The graph for illustrative example 1 before rescheduling

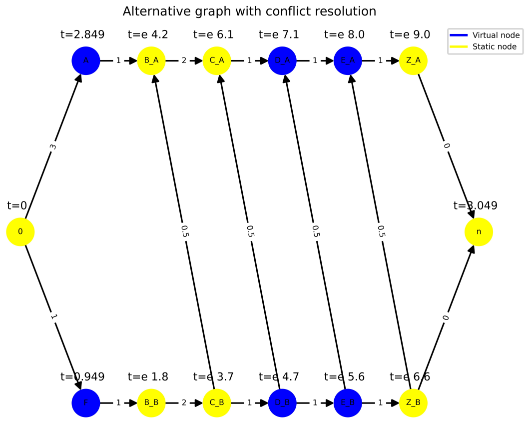


Figure 5.7: The graph representation for illustrative example 1 if rescheduling is applied

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A,F | 2.8 | 0.9 | - |
| B | 3.7 | 6.1 | A |
| C | 5.6 | 8.0 | A |
| D | 6.6 | 9.0 | A |
| E | 7.5 | 9.9 | A |
| **Z** | 8.5 | 10.949 | A |

Table 5.7: Illustrative example 1 - without rescheduling

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A,F | 2.8 | 0.9 | - |
| B | 4.2 | 1.8 | B |
| C | 6.1 | 3.7 | B |
| D | 7.1 | 4.7 | B |
| E | 8.0 | 5.6 | B |
| **Z** | 9.0 | 6.6 | B |

Table 5.8: Illustrative example 1 - with rescheduling

| | Final station | |
|---|---|---|
| Delay | Consecutive | Propagated |
| | A | B |
| Without rescheduling | 0 | 3.949 |
| With rescheduling | 3.049 | 0 |

Table 5.9: The total propagated delay in illustrative example 1 in different situations

### 5.5.2. Two trains passing two stations

The second illustrative example shows two trains passing two stations. The reason to include an example containing stations in the proof of concept is twofold. First, it is valuable to verify whether the arcs for station modelling are functioning properly and it adds some complexity to the example situation. Second, passenger trains and passenger satisfaction are a fair share of concern in railway operations and therefore it is valuable for ProRail to show an illustrative example involving stations.

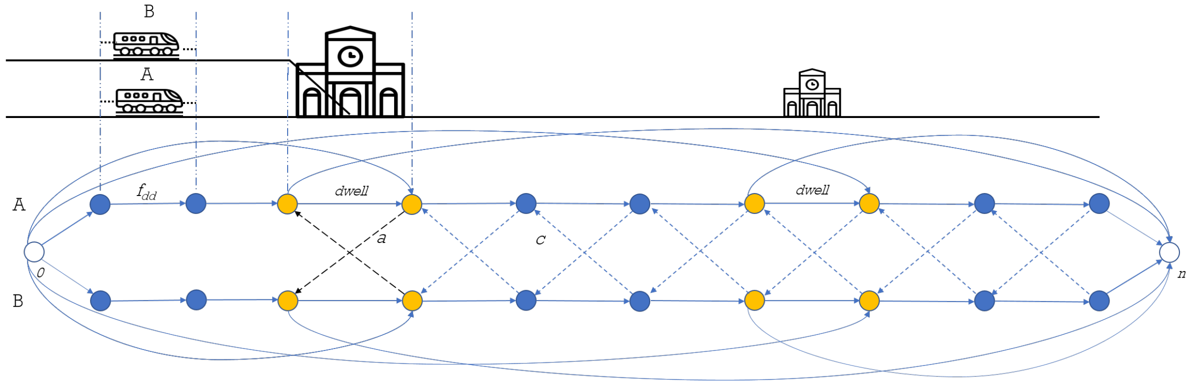


Figure 5.8: Illustrative example 2: two trains passing two stations

The situation of illustrative example 2 can be described as follows. Two trains are converging towards a station, where they are arriving simultaneously because of an initial delay of train A. After the first station has been passed, the trains are driving on a straight single track towards a second station. Since this example involves two stations, arrival and departure times (timetable times) are now added to the input parameters. These timetable times are formulated such that they are met when the train drives according to scheduled running- and clearing (processing) times. This gives the input parameters as presented in Table 5.10.

Table 5.10: Input parameters for illustrative example 2

| | Train A | Train B |
|---|---|---|
| Nodes | A, B, C_A,... J_A | K, L, C_B,... J_B |
| Processing times | 1, 1, 2, 1, 1, 1, 2, 1, 1 | 1, 2, 1, 1, 1, 2, 1, 1 |
| Initial delay | 4 | 0 |
| Headway time | 0.5 | 0.5 |
| Arrival times | 2, 7 | 5, 10 |
| Departure times | 4, 9 | 7, 12 |

In the same manner as illustrative example 1, a final station Z has been added, the graph has been generated in Python, and the optimization has been performed. The results are again visualized in both table and figure form. Enlarged versions of the figures can be found in Appendix A. As visualized in Table 5.11, 5.13 and Figure 5.9, train A was scheduled to be the preceding train. After conflict, the unrescheduled situation leads to $t_n$ equal to 3.69. Table 5.14 and Figure 5.11 show the situation after conflict resolution has been performed.

Evidently, it shows that the rescheduled situation is equal to the unrescheduled situation. However, this actually proofs an important working principle of the model. If reordering would have been applied in this example, train A would have a delay of 5.5 when arriving at the first station, leading to $t_n$ equal to 5.5. This is a higher maximum delay than $t_n$ equal to 3.69. Therefore, this example nicely shows that no rescheduling measures are taken when it worsens the situation instead of improves.

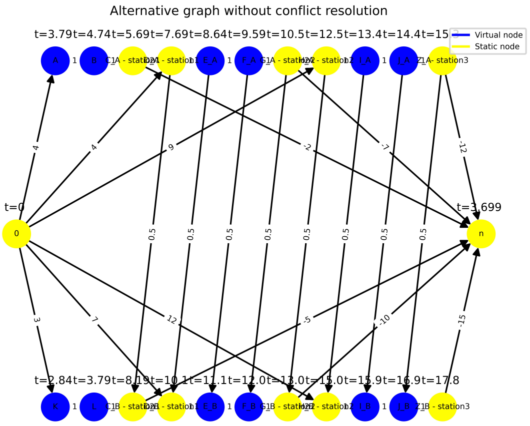


Figure 5.9: The graph representation for illustrative example 2 if no rescheduling is applied

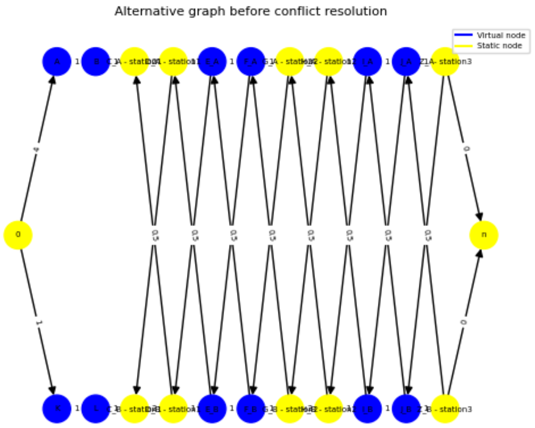


Figure 5.10: The graph representation for illustrative example 2 before rescheduling

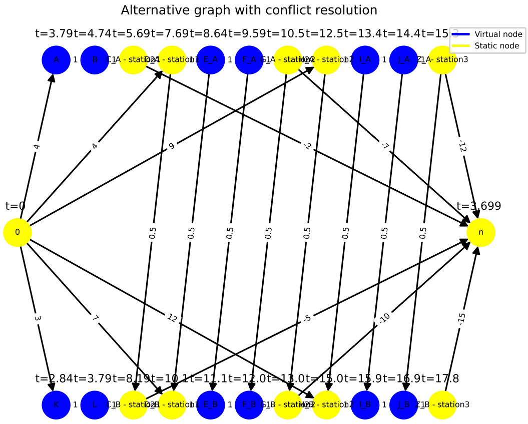


Figure 5.11: The graph representation for illustrative example 2 if rescheduling is applied

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A, K | 0 | 3 | - |
| B, L | 1 | 4 | - |
| **C** | 2 | 5 | A |
| **D** | 4 | 7 | A |
| E | 5 | 8 | A |
| F | 6 | 9 | A |
| **G** | 7 | 10 | A |
| **H** | 9 | 12 | A |
| **I** | 10 | 13 | A |
| J | 11 | 14 | A |
| **Z** | 11 | 15 | A |

Table 5.11: Illustrative example 2 - scheduled situation

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A, K | 4 | 3 | - |
| B, L | 5 | 4 | - |
| **C** | 6 | 5 | - |
| **D** | 8 | 7 | - |
| E | 9 | 8 | - |
| F | 10 | 9 | - |
| **G** | 11 | 10 | - |
| **H** | 13 | 12 | - |
| **I** | 14 | 13 | - |
| J | 15 | 14 | - |
| **Z** | 16 | 15 | - |

Table 5.12: Illustrative example 2 - conflicted situation

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A, K | 3.8 | 2.8 | - |
| B, L | 4.7 | 3.8 | - |
| **C** | 5.7 | 8.2 | A |
| **D** | 7.7 | 10.1 | A |
| E | 8.6 | 11.1 | A |
| F | 9.6 | 12.0 | A |
| **G** | 10.5 | 13.0 | A |
| **H** | 12.5 | 15.0 | A |
| I | 13.4 | 15.9 | A |
| J | 14.4 | 16.9 | A |
| **Z** | 15.3 | 17.8 | A |

Table 5.13: Illustrative example 2 - without rescheduling

| Node | t | | Precedence |
|---|---|---|---|
| | Train A | Train B | |
| A, K | 3.8 | 2.8 | - |
| B, L | 4.7 | 3.8 | - |
| **C** | 5.7 | 8.2 | A |
| **D** | 7.7 | 10.1 | A |
| E | 8.6 | 11.1 | A |
| F | 9.6 | 12.0 | A |
| **G** | 10.5 | 13.0 | A |
| **H** | 12.5 | 15.0 | A |
| I | 13.4 | 15.9 | A |
| J | 14.4 | 16.9 | A |
| **Z** | 15.3 | 17.8 | A |

Table 5.14: Illustrative example 2 - with rescheduling

| | Station 1 | | Station 2 | | Final station | |
|---|---|---|---|---|---|---|
| Delay | Consecutive | Propagated | Consecutive | Propagated | Consecutive | Propagated |
| | A | B | A | B | A | B |
| Without rescheduling | 0 | 3.2 | 0 | 3 | 0 | 2.8 |
| With rescheduling | 0 | 3.2 | 0 | 3 | 0 | 2.8 |

Table 5.15: The total propagated delay in illustrative example 2 in different situations

### 5.5.3. Two trains passing a station, a third train crossing the route

The third illustrative example consists of two trains passing a station, similar to illustrative example 2, followed by a third train that crosses the route of the two trains. This example distinguishes itself from the previous examples, because the third train does not remain on the same track as the other two trains. It is interesting to also investigate the crossing of two routes, instead of only looking at trains on the same route. The reason for this is that, apart from converging junctions, crossings are also frequently appearing points of possible conflicts on railway tracks, which is why this illustrative example is valued by ProRail.

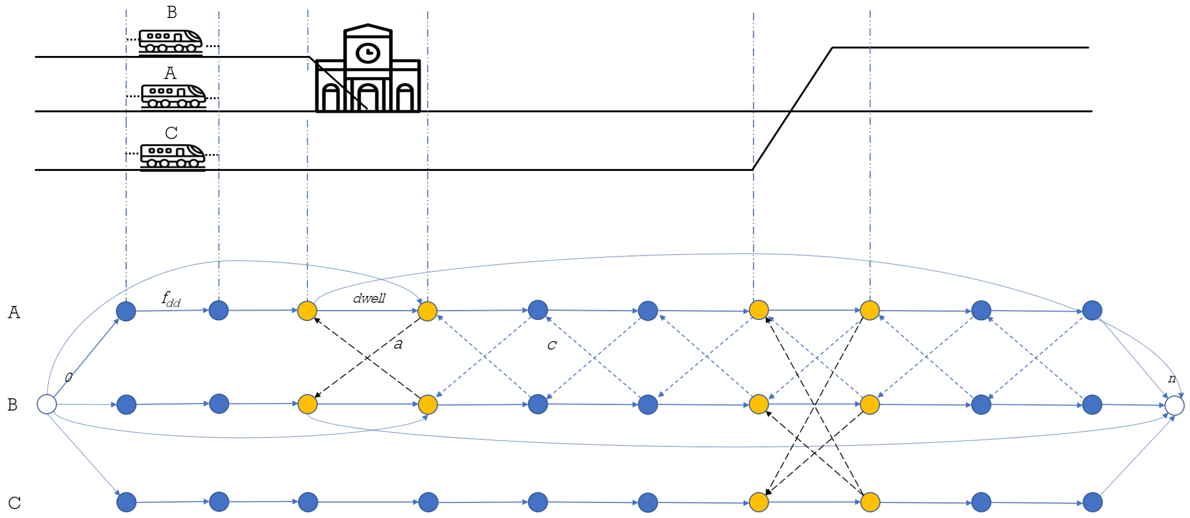


Figure 5.12: Illustrative example 3: two trains passing a station and a third train crossing the route

Adding a third train adds to the complexity of the problem, first of all since there are more possibilities in train reordering and second of all because there are more trains to be affected by propagated delay. The results are presented in the same manner as in the previous two illustrative examples. Enlarged versions of the figures can be found in Appendix A. Looking at the switch, the scheduled order is A-B-C, also depicted in Table 5.16. When giving train A an initial delay of 5, maintaining this order leads to a propagated delay of 3.4 for train B and 5.8 for train C at the final stations Z and Y respectively, see Table 5.20. Running the optimization problem leads to changing the train order to A-C-B, which decreases the maximum propagated delay from 5.8 to 4.8 (17%), as can be observed in Table 5.20. It can also be observed that, although the maximum propagated delay is decreased, to the total propagated delay at the final station slightly increases from 9.2 to 9.6. However, this model is designed to reduce maximum propagated delays and therefore functions properly. In future research it can be valuable to play with different objectives or use multi objective functions.

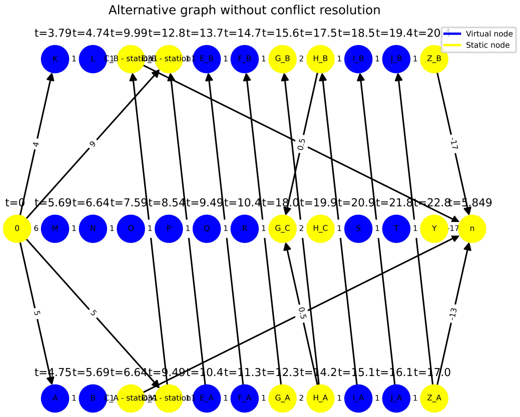


Figure 5.13: The graph representation for illustrative example 3 if no rescheduling is applied

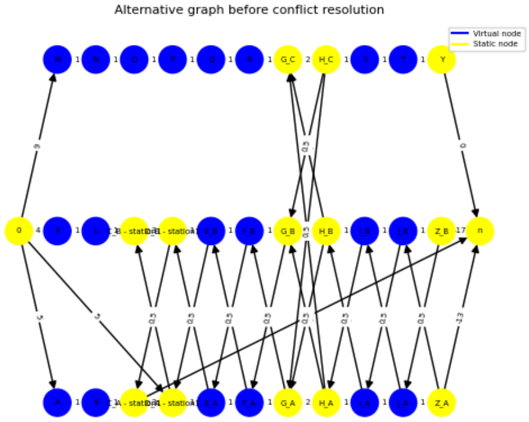


Figure 5.14: The graph representation for illustrative example 3 before rescheduling

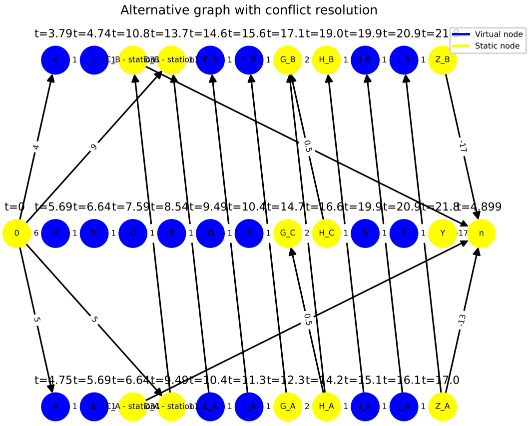


Figure 5.15: The graph representation for illustrative example 3 if rescheduling is applied

| Node | t | | | Precedence |
|---|---|---|---|---|
| | Train A | Train B | Train C | |
| A, K, M | 0 | 4 | 9 | - |
| B, L, N | 1 | 5 | 10 | - |
| **C**, O | 2 | 6 | 11 | A |
| **D**, P | 5 | 9 | 12 | A |
| E, Q | 6 | 10 | 13 | A |
| F, R | 7 | 11 | 14 | A |
| G | 8 | 12 | 15 | A, B |
| H | 10 | 14 | 17 | A, B |
| **I**, S | 11 | 15 | 18 | A |
| J, T | 12 | 16 | 19 | A |
| **Z, Y** | 13 | 17 | 20 | A |

Table 5.16: Illustrated example 3 - scheduled situation

| Node | t | | | Precedence |
|---|---|---|---|---|
| | Train A | Train B | Train C | |
| A, K, M | 5 | 4 | 9 | - |
| B, L, N | 6 | 5 | 10 | - |
| **C**, O | 7 | 6 | 11 | - |
| **D**, P | 10 | 9 | 12 | - |
| E, Q | 11 | 10 | 13 | - |
| F, R | 12 | 11 | 14 | - |
| G | 13 | 12 | 15 | - |
| H | 15 | 14 | 17 | - |
| **I**, S | 16 | 15 | 18 | - |
| J, T | 17 | 16 | 19 | - |
| **Z, Y** | 18 | 17 | 20 | - |

Table 5.17: Illustrated example 3 - conflicted situation

| Node | t | | | Precedence |
|---|---|---|---|---|
| | Train A | Train B | Train C | |
| A, K, M | 4.8 | 3.8 | 5.7 | - |
| B, L, N | 5.7 | 4.7 | 6.6 | - |
| **C**, O | 6.6 | 10 | 7.6 | A |
| **D**, P | 9.5 | 12.8 | 8.5 | A |
| E, Q | 10.4 | 13.7 | 9.5 | A |
| F, R | 11.3 | 14.7 | 10.4 | A |
| G | 12.3 | 15.6 | 18 | A, B |
| H | 14.2 | 17.5 | 19.9 | A, B |
| I, S | 15.1 | 18.5 | 20.9 | A |
| J, T | 16.1 | 19.4 | 21.8 | A |
| **Z, Y** | 17 | 20.4 | 22.8 | A |

Table 5.18: Illustrated example 3 - unrescheduled

| Node | t | | | Precedence |
|---|---|---|---|---|
| | Train A | Train B | Train C | |
| A, K, M | 4.8 | 3.8 | 5.7 | - |
| B, L, N | 5.7 | 4.7 | 6.6 | - |
| **C**, O | 6.6 | 10.8 | 7.6 | A |
| **D**, P | 9.5 | 13.7 | 8.5 | A |
| E, Q | 10.4 | 14.6 | 9.5 | A |
| F, R | 11.3 | 15.6 | 10.4 | A |
| G | 12.3 | 17.1 | 14.7 | A, C |
| H | 14.2 | 19.0 | 16.6 | A, C |
| **I**, S | 15.1 | 19.9 | 19.9 | A |
| J, T | 16.1 | 20.9 | 20.9 | A |
| **Z, Y** | 17.0 | 21.9 | 21.8 | A |

Table 5.19: Illustrated example 3 - rescheduled

| | Station 1 | | | Final station | | |
|---|---|---|---|---|---|---|
| Delay | Consecutive | Propagated | | Consecutive | Propagated | |
| | A | B | C | A | B | C |
| Without rescheduling ($t_n = 5.8$) | 0 | 4 | - | 0 | 3.4 | 5.8 |
| With rescheduling ($t_n = 4.8$) | 0 | 4 | - | 0 | 4.8 | 4.8 |

Table 5.20: The total propagated delay in illustrative example 3 in different situations

## 5.6. Model verification

This section will describe a model verification study that has been performed using the illustrative examples. The goal of this study is to ensure that the model is functioning properly before moving on to the case study in the next chapter. The verification study will be executed by testing the model with different test cases, created by the variation of certain input parameters. An expected outcome is determined for every test case, which can then be verified for every illustrative example. The used input parameters and their expected outcomes are described in Table 5.21.

| Nr | Test case description | Expected outcome |
|---|---|---|
| 1 | Increase minimum headway | Increase of tn |
| 2 | Increase buffer times | Decrease of tn (or equal) |
| 3 | Departure time >(arrival time +dwell time) | Train leaves at departure time |
| 4 | Running times of train A »running times of train B | Train B precedes train A |
| 5 | Force alternative arc choice | Conditional arc obeys |

Table 5.21: Test cases for verification study and their expected outcome

Test case number 1 describes to increase the minimum headway distance between trains. Since this should cause trains to wait longer before they are allowed to pass a node, this effect should appear in the objective value $t_n$ as an increase.
Test case number 2 states to increase the percentage of buffer time related to running time recovery. As explained in 4.4.3, scheduled times contain slack time meaning the minimum running times are shorter than scheduled running times. Consequently, trains can catch up with delays obtained earlier on the route. Increasing the percentage of slack time could result in a decreased objective value $t_n$. However, it could also be that timetable constraints are unfavourable and that increased buffer times are not able to decrease $t_n$ even further. In that case $t_n$ remains equal.
Test case number 3 involves the inspection of the timetable arc functionality. If the scheduled departure time is increased to more than the arrival time + dwell time, the actual train departure should increase to this newly scheduled departure time.
Test case number 4 verifies whether the model favours network efficiency. By largely increasing the running times of one train, it should be undesirable to keep another train waiting behind. As it were, the model should prevent fast driving trains to get stuck behind slow trains.
Test case number 5 will check the functioning of conditional arc obedience. When an alternative arc choice is forced, by altering the binary decision variables by hand, the conditional arc should follow this imposed train order.

Tables 5.22, 5.23 and 5.24 show the verification study as executed for all illustrative examples. Note that test case number 3 is not applicable to illustrative example 1, since that scenario contains no stations but the final station. It can be concluded from these tables that the model behaves properly and handles varying inputs as expected.

| Nr | Test case description | Expected outcome | Outcome | Pass |
|---|---|---|---|---|
| 1 | minimum headway = 0.7 | $t_n$ >3.05 | $t_n$ = 3.25 | ✓ |
| 2 | buffertime = 10% | $t_n \leq 3.05$ | $t_n$ = 2.6 | ✓ |
| 3 | - | - | - | - |
| 4 | Process times B = 3*Process times A | Train A precedes train B | Train A precedes train B | ✓ |
| 5 | $z_i$ = [0 1] (Train A precedes) | Train A precedes train B | Train A precedes train B | ✓ |

Table 5.22: Verification study, applied to illustrative example 1

| Nr | Test case description | Expected outcome | Outcome | Pass |
|---|---|---|---|---|
| 1 | minimum headway = 0.7 | $t_n$ >1.5 | $t_n$ = 1.7 | ✓ |
| 2 | buffertime = 10% | $t_n \leq 1.5$ | $t_n$ = 1.5 | ✓ |
| 3 | departure time train B at node D = 8 | $t_{D_B} \geq 8$ | $t_{D_B}$ = 8 | ✓ |
| 4 | Process times B = 3*Process times A | Train A precedes train B | Train A precedes train B | ✓ |
| 5 | $z_i$ = [0 1] (Train A precedes) | Train A precedes train B | Train A precedes train B | ✓ |

Table 5.23: Verification study, applied to illustrative example 2

| Nr | Test case description | Expected outcome | Outcome | Pass |
|---|---|---|---|---|
| 1 | minimum headway = 0.7 | $t_n$ >4.99 | $t_n$ = 5.29 | ✓ |
| 2 | buffertime = 10% | $t_n \leq 4.99$ | $t_n$ = 4.29 | ✓ |
| 3 | departure time train A at node D = 10 | $t_{D_A} \geq 10$ | $t_{D_A}$ = 10.0 | ✓ |
| 4 | Process times C = 3*Process times B | Train C precedes train B | Train C precedes train B | ✓ |
| 5 | $z_i$ = [0 1], $z_{i+x}$ = [0 1] | Order is A-B-C | Order is A-B-C | ✓ |

Table 5.24: Verification study, applied to illustrative example 3

## 5.7. Conclusion

This chapter has discussed how an alternative graph can be constructed from necessary input variables and thus how the theoretical model from Chapter 4 can be put into practice. In most basic form, this is done in three step. First, timeable points are translated into nodes. Second, processes and processing times are translated into edges and edge weights respectively. Third, headway times are translated into alternative and conditional arcs between points of potential conflict. This answers subquestion 3a. After constructing the alternative graph, the optimization problem can be solved with Gurobi commercial solver, considered a suitable solver because of its capability to efficiently solve constrained optimization problems, and therefore answering subquestion 3b. Subquestion 3c has been answered by an elaborate analysis of three different illustrative examples followed by a verification study. The illustrative examples have shown that the solution approach can handle different types of problems and is able to reduce total propagated delay. These examples have constructed a proof of concept for the novel alternative graph approach, wherein the different node and arc types are distinguished and put to use. The verification study has displayed that the model correctly satisfies the constraints and model formulation for all considered test scenarios.

# 6

# Model validation and assessment on the Rotterdam-The Hague Dutch corridor

In this chapter, a case study is executed using the model implementation as described in Chapter 5. The case study is performed with real life data supplied by ProRail. This chapter first elaborates on the corridor that will be the focus of the case study, the Rotterdam - The Hague corridor. Subsequently, it is explained how the real life data have been obtained and modified for suitability. Next, this chapter describes what type of inputs will be given to the model and why this is considered valuable. This is followed by the results generated by the corresponding inputs. The results are presented on the basis of the KPI's as formulated in Chapter 3. This means subquestion 4 will be answered in this chapter, which will be done explicitly in the chapter conclusion.

## 6.1. Rotterdam - The Hague corridor

The considered area for the case study is the Rotterdam - The Hague corridor, see Figure 6.1 (corridor split in two for readability purposes). This means trains are analysed driving from Rotterdam Central Station to The Hague Central Station (only in this direction). This corridor is part of the larger ROSA corridor (Rotterdam, Schiphol, Arnhem). At Rotterdam Central Station, there are two platforms from which trains are leaving in the direction of The Hague, platforms 8 and 9. Sprinters leave from platform 8, intercities from platform 9. Figure 6.2 shows a zoom in on platforms 8 and 9. Figures 6.1 and 6.2 are extracted from the ProRail tool “InfraMonitor” and give an insight in the infrastructure layout of the case study. The sprinter and the intercity both follow a different route, which are the following:

- Sprinter: Rotterdam - Schiedam - Delft Campus - Delft - Rijswijk - Den Haag Moerwijk - Den Haag
- Intercity: Rotterdam - Schiedam -Delft - Den Haag

Table 6.1 shows routes of all trains leaving platforms 8 and 9 within one hourly cycle. Times are given as time instants between 0 and 60 minutes/3600 seconds. The 5100 and 5000 trains are sprinters and the 2400 trains are intercities, as can be seen from the correspondence to the routes as described above.

| | | C5100 | | C5000 | | C2400 | | A5100 | | A5000 | | A2400 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | min | sec | min | sec | min | sec | min | sec | min | sec | min | sec |
| Rtd | D | 6.0 | 360 | 21 | 1260 | 27 | 1620 | 36 | 2160 | 51 | 3060 | 57 | 3420 |
| Sdm | A | 10.0 | 604 | 25.1 | 1504 | 30.7 | 1842 | 40.1 | 2404 | 55.1 | 3304 | 60.7 | 3642 |
| | D | 11.1 | 666 | 26.1 | 1566 | 31.9 | 1914 | 41.1 | 2466 | 56.1 | 3366 | 61.9 | 3714 |
| Dtcp | A | 16.6 | 998 | 31.6 | 1897 | - | - | 46.6 | 2797 | 61.6 | 3697 | - | - |
| | D | 17.8 | 1068 | 32.7 | 1962 | - | - | 47.8 | 2868 | 62.7 | 3762 | - | - |
| Dt | A | 19.9 | 1193 | 34.8 | 2087 | 38.4 | 2306 | 49.9 | 2993 | 64.8 | 3887 | 68.4 | 4106 |
| | D | 20.7 | 1242 | 35.7 | 2142 | 39.4 | 2364 | 50.7 | 3042 | 65.7 | 3942 | 69.4 | 4164 |
| Rsw | A | 24.1 | 1443 | 39.1 | 2343 | - | - | 54.1 | 3243 | 69.1 | 4143 | - | - |
| | D | 24.9 | 1494 | 39.9 | 2394 | - | - | 54.9 | 3294 | 69.9 | 4194 | - | - |
| Gvmw | A | 27 | 1620 | 42 | 2520 | - | - | 57.0 | 3420 | 72.0 | 4320 | - | - |
| | D | 27.9 | 1674 | 42.9 | 2574 | - | - | 57.9 | 3474 | 72.9 | 4374 | - | - |
| Gv | A | 30 | 1837 | 45.6 | 2737 | 46.2 | 2770 | 60.6 | 3637 | 75.6 | 4537 | 76.2 | 4571 |

Table 6.1: Routes to The Hague of trains leaving within one hourly cycle from Rotterdam platforms 8 and 9

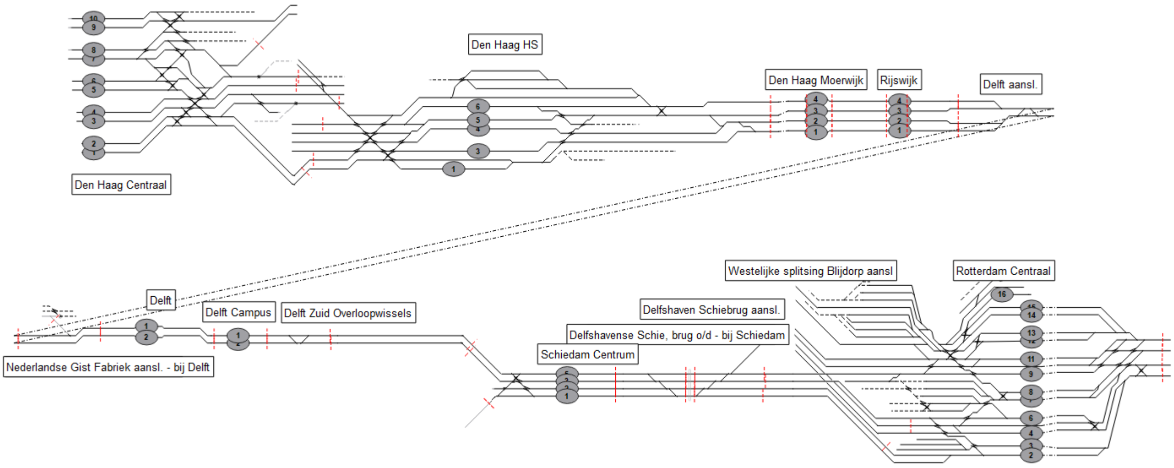


Figure 6.1: Microscopic overview of the Rotterdam - The Hague corridor, extracted from the ProRail tool ”inframonitor”

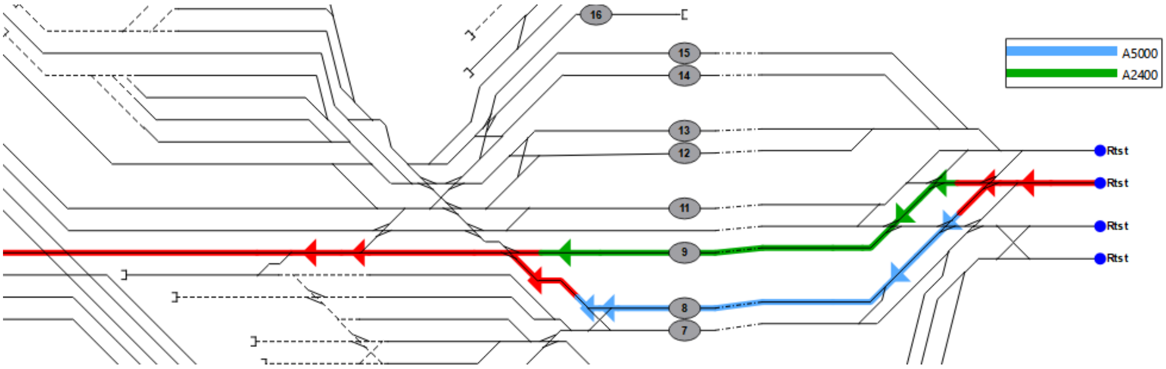


Figure 6.2: Microscopic overview of platforms 8 and 9 at Rotterdam Central Station, extracted from the ProRail tool ”inframonitor”. Green line: intercity, Blue line: sprinter, Red line: coinciding routes

## 6.2. Obtaining simulated data

In order to be able to implement the conflict resolution model within the Rotterdam - The Hague corridor, simulated data have been gathered and adjusted. This section will first discuss how the data has been obtained and secondly elaborate on how these data needed to be modified in order to create a suitable dataset to match the developed mathematical model and code.

### 6.2.1. Prorail simulation tools

The data used in this case study have been obtained using the ProRail simulation tool FRISO (Flexible Rail Infrastructure Simulation Environment, in dutch: Flexibele Rail Infra Simulatie Omgeving). FRISO is a microscopic rail traffic simulation tool in which trains can be simulated given different input timetables, rolling stock, and infrastructure modules. FRISO simulates an environment where the scheduled timetable is maintained, relying on buffer times for small disturbances. In order to perform conflict resolution that uses automated rescheduling measures, FRISO needs to be connected to either a human dispatcher a traffic management system. The Traffic Management System (TMS) connected to FRISO that is currently being tested in the simulation centre can solve conflicts in ERTMS level 2, in fixed block signalling systems. This means that data generated by FRISO are essentially based on fixed block infrastructure, and for this research they have been modified to resemble a moving block signalling system as accurately as possible. The next section will elaborate further on how these modifications have been performed.

Before discussing the modifications, it is of importance to understand how the simulation has been performed and what type of data have been generated. In the FRISO environment, all trains have been simulated to be driving conflict free according to their regular timetables. During the simulation, information is being gathered on the trains' behaviour, logged at different timeable points. These timeable points are different types of points, spread out over infrastructure. Information can for instance be logged at signals, welds, area boundaries, switches and a variety of other distinguishable infrastructure points. Eventually, occupation times of different infrastructural elements can be derived from these data to construct blocking times. For a description of the FRISO loggings and information types, refer to Appendix B.

### 6.2.2. Data set construction

For the construction of a suitable data set that matches the modelling approach, adjustments to the simulation output are necessary. First, not all the data in the FRISO logging are required input for the conflict resolution model. Second, and controversially, there are also some aspects that are missing from the logging generated by the FRISO simulation. Therefore, adjustments need to be made regarding the following four aspects:

1. There are more timeable points in the logging than necessary for model input
2. There is more measured information in the logging than necessary
3. There are no distinguished station stops in the logging
4. There are no distinguished fixed and virtual sections in the logging

The first modification has been performed as follows. In the logged data, a large variety of point types have been included. To decrease this amount of points, the choice has been made to only keep those that were timed at certain types. In collaboration with ProRail, the following points types are considered most important:

- Section (dutch: Section)
- Platform (dutch: Spoor)
- Switch (dutch: Wissel)
- Weld (dutch: Las)
- Heart dispatching point (dutch: Hart dienstregel punt)

Regarding the second matter, adjustments have been made to create a more clear data structure. This is done by selecting only the exact information necessary to create nodes, arcs, and their corresponding attributes. This includes the following information:

- Name of train
- Name of timeable point
- Type of timeable point

- Corresponding dispatching area
- Time instant of passing the timeable point with the train's front end
- Distance from starting point
- The following calculated values:
  - Reaction Time ($t_{reac}$)
  - Approach Time ($t_a$)
  - Release Time ($t_{rel}$)
  - Clearing time ($t_{cl}$)

Thirdly, an inclusion of station stops needs to be established. This has been executed with help of an additional data set provided by ProRail that contains all train events in the ROSA corridor. The station stops as determined in this additional dataset have been inserted into the main dataset.

Lastly, the dataset should be equipped with a distinction between static and virtual sections. This will also lead to resemblance of a moving block signalling system instead of fixed block. As already mentioned, one of the included aspects of information is the trains' clearing times ($t_{cl}$). These clearing times have been calculated in another research, that specializes in predicting blocking times for moving block systems based on data from fixed block systems. This research uses a tool called EG train ((Environment for the desiGn and simulaTion of RAIlway Networks), for more information see [50]. These clearing times can thus function as processing times between two virtual nodes, and running times can be set to zero. However, not all running times are zero, since the graph formulations still needs static sections to resemble switches and stations. More precisely, static sections need to be created in order for all other timeable points to be interpreted as virtual nodes. To serve this purpose, an additional column for running time ($t_{run}$) values has been added. Further, the full dataset has been filtered on only timing points related to switches and stations. Static sections have been composed in the two following manners:

- **Static section for switches** Since a switch finds itself within a static section that starts before the switch and ends after the switch, the first timeable point before and after the switch have been marked as the begin (point $i$) and end point ($j$) of a static section. The timeable point representing the actual switch was then removed. The running time of the static section then equal to the train front's end passing time of $j - i$.
- **Static section for stations** Stations are regarded as a static section starting at the station arrival and ending at the station departure. These are already two separate timeable points in the dataset. The running time of the static section is then equal to the dwell time of the train as provided by ProRail.

In final overview, the dataset that is imported into the script looks as illustrated below in Table 6.2 (note that this is only a few lines). This final dataset contains all information to construct the alternative graph and its corresponding arc weights as defined in Section 4.4.3.

| **Train name** | **Point name** | **Point Type** | **Area** | $t_{frontend}$ | **Distance** | $t_{reac}$ | $t_a$ | $t_{rel}$ | $t_{cl}$ | $t_{run}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| A2400-H-1 | 305 | SPOOR | Sdm | 3624.97 | 4238 | 8 | 34.80 | 1 | 120.61 | 0 |
| A2400-H-1 | _LAS-57B-V-1300 | LAS | Sdm | 3637.39 | 4307 | 8 | 14.71 | 1 | 113.40 | 0 |
| A2400-H-1 | Station_sdm_ar | STATION-AR | Sdm | 3642.29 | 4307 | 8 | 14.71 | 1 | 0 | 71.70 |
| A2400-H-1 | Station_sdm_dep | STATION-DEP | Sdm | 3714 | 4307 | 0 | 0 | 1 | 19.23 | 0 |
| A2400-H-1 | A54AT | SECTIE | Sdm | 3741.39 | 4458 | 8 | 3.68 | 1 | 19.23 | 0 |
| A2400-H-1 | _LAS-57B-V-300 | LAS | Sdm | 3755.00 | 4638 | 8 | 9.76 | 1 | 15.69 | 0 |
| A2400-H-1 | 57BT | SECTIE | Sdm | 3757.57 | 4678 | 8 | 11.64 | 1 | 15.20 | 5.28 |
| A2400-H-1 | _LAS-45A-L-100 | LAS | Sdm | 3762.86 | 4766 | 8 | 13.40 | 1 | 14.33 | 4.01 |
| A2400-H-1 | 45AT | SECTIE | Sdm | 3766.87 | 4837 | 8 | 14.79 | 1 | 13.74 | 0 |

Table 6.2: A few example rows of the final dataset for train A2400

## 6.3. Creating test cases for conflict resolution

In this section, the input to the case study will be constructed and explained. In order to analyse the performance of the conflict resolution model, different test cases in the form of initial delay will be given to simulate a conflict, similar to what has been done in the illustrative examples in Chapter 5. These test cases will be thoroughly constructed in this section.

The response of the conflict resolution system can be influenced by the initial delay in two main ways:

- The amount of initial delay. This is expected to influence conflict resolution for:
  - Amount of input delay can influence how much delay is propagated
  - Amount of input delay may influence the computational complexity of the problem

- The train types (sprinter or intercity) that experiences this initial delay. This is expected to influence performance for:
  - The speed difference between the two train types can influence how delay is propagated
  - The number of stops can influence how delay is propagated
  - The two aspects above may influence the computational complexity of the problem

It is thus valuable to vary the initial delay in terms of both quantity and train type. Besides the type of train, it is also useful to investigate the difference between the situation where only one train is initially delayed and the event of the consecutive train also having an initial delay. Additionally, this can also be done for not the consecutive but another selected train in the network having also an initial delay. Taking into account all these aspects, four test cases in which initial delay will be simulated have been formulated:

1. One delayed sprinter
2. Two delayed sprinters
3. One delayed intercity, one delayed sprinter
4. One delayed intercity

It should be noted that in the above cases, the delayed trains will always be trains that mostly influence the rest of the network, thus the trains scheduled to leave earliest with respect to the trains not initially delayed.

For all these four test cases, the amount of initial delay will be varied. When establishing the steps in which the initial delay will be increased, the timetable is taken into account. From Table 6.1, it can be seen that the scheduled gap between two trains leaving Rotterdam is not the same for all trains. This means that an initial delay of 10 minutes may cause different conflict resolution behaviour depending on what train is initially delayed. Therefore, the choice has been made to impose initial delays relative to the planned headway between two trains. In this way, the conflict resolution can be compared as accurately as possible for all different test cases. This leads to three delay scenarios, which are as follows:

1. Planned headway -30%
2. Planned headway
3. Planned headway +30%

This means the inputs for the case study are as presented in Table 6.3.

| Test case | Delay scenario | | Input delay value (sec) | | | |
|---|---|---|---|---|---|---|
| | | | C5000 | C5100 | C2400 | Total |
| 1. Delayed sprinter | 1.1 | ph -30% | 630 | | | 630 |
| | 1.2 | ph | 900 | | | 900 |
| | 1.3 | ph + 30% | 1170 | | | 1170 |
| 2. Two delayed sprinters | 2.1 | ph -30% | 630 | 280 | | 910 |
| | 2.2 | ph | 900 | 400 | | 1300 |
| | 2.3 | ph +30% | 1170 | 520 | | 1690 |
| 3. Delayed sprinter & intercity | 3.1 | ph -30% | 630 | | 378 | 1008 |
| | 3.2 | ph | 900 | | 540 | 1440 |
| | 3.3 | ph +30% | 1170 | | 702 | 1872 |
| 4. Delayed intercity | 4.1 | ph -30% | | | 378 | 378 |
| | 4.2 | ph | | | 540 | 540 |
| | 4.3 | ph +30% | | | 702 | 702 |

Table 6.3: The different inputs to be imposed on the network for the case study

## 6.4. Conflict resolution results

After running the case study model for all different test cases, results and model performance can be observed and analysed, which will be done in this section. First, a result overview will be presented and discussed, where some first conclusions can already be drawn. Second, results will be examined by the two most important KPI's, the propagated delay and the computation time. This will be done by a sensitivity analysis, expanding the insights in the performance of the conflict resolution model.

### 6.4.1. Result overview

An overview of case study results can be found in Table 6.4. The test case numbers from Table 6.3 can be recognized in Table 6.4, which has been extended with the results per test case and corresponding delay scenario. The following results are presented:

- **Minimum propagated delay at final station** This is the smallest value of propagated delay found at the arrivals at the final station. In other words, the arrival delay of the train that arrives with the least delay.
- **Maximum propagated delay at final station** This is the largest value of propagated delay found at the arrivals at the final station. In other words, the arrival delay of the train that arrives with the most delay.
- **Total propagated delay at final station** This is the sum of all trains' arrival delays at the final station.
- **Absolute reduction of total propagated delay** The difference in or seconds of propagated delays at final station with respect to the unrescheduled situation.
- **Percentage reduction of total propagated delay** The difference between propagated delay with respect to the unrescheduled situation, expressed in percentage.
- **Number of retimed trains** The number of trains that has been retimed with respect to the unrescheduled situation.
- **Number of reorderd trains** The number of trains that has been reordered with respect to the unrescheduled situation.
- **Computation time** The computation time elapsed during the solution process.

For every test case and corresponding delay scenario, more elaborate results containing arrival- and departure times and delays at all intermediate stations for all trains can be found in Appendix C. Also, it should be noted that, due to the fact that the optimal solution is not unique and the commercial solver has a slight variance, result contributions are only considered significant if the alteration exceeds 30 seconds. For example, if a train is said to be departing from a station 20 seconds later than scheduled, this is not considered a retiming measure. Similarly, if delay is reduced by 10 seconds, this is not considered a significant delay reduction. Table 6.4 is based on this assumption. From Table 6.4, some first conclusions can already be drawn.

First, the table shows that the minimum arrival delay at the final station is zero or negligible for all cases, which means there is always at least one train that reaches the final station without any delay inheritance. In the detailed presentation in Appendix C, it can be found that train A2400 always reaches the final station with a propagated delay of under 30 seconds, where from it can be concluded that the initial delays given to this case study are handled in such a way that delay is not propagated into the next hourly cycle.

Second, propagated delay is reduced for all test cases. From the detailed results in Appendix C it can also be seen that no consecutive delay has been generated for the initially delayed trains, who all arrive at the final station with less delay than they entered the network with. It should be noted that a consecutive delay for initially delayed trains could in some scenarios be beneficial, if it takes away propagated delay from other trains. However, such a scenario has not presented itself in this case study. Regardless, it can thus be said that for the provided input, the model's performance is satisfactory and is able to do what it is designed for i.e. reducing delays.

Third, the number of retimed trains is in all cases equal or larger than the number of reordered trains. This is in line with expectations, since reordering a train generally implies the train is also retimed. Additionally, retiming a train is a less severe measure and can already (partially) reduce propagated delay without reordering being necessary.

Lastly, it is evident that a reordering measure is always imposed on at least two trains, which is the minimum number of reordered trains if larger than zero. However, in this case study reordering is never performed for more than two trains. This implies that reordering more than two trains did not help the objective in this case study, but this might be different in more complex situations or if initial timetable conditions are different. The upcoming subsection will provide more insights in the case study results.

| **Test case** | **Propagated delay (at final station)** | | | **Total delay reduction** | | **Trains rescheduled** | | **Comp. time** |
|---|---|---|---|---|---|---|---|---|
| | **Min** | **Max** | **Total** | **Absolute** | **Percentage** | **Retimed** | **Reordered** | |
| 1.1 | 0 | 77 | 96 | 121 | 55.7 | 4 | 2 | 1.83 |
| 1.2 | 0 | 284 | 512 | 222 | 30.2 | 3 | 2 | 2.87 |
| 1.3 | 0 | 540 | 1058 | 205 | 16.2 | 3 | 2 | 1.36 |
| 2.1 | 0 | 150 | 242 | 236 | 49.3 | 4 | 2 | 2.21 |
| 2.2 | 0 | 284 | 510 | 230 | 31.1 | 3 | 2 | 8.41 |
| 2.3 | 0* | 540 | 1048 | 218 | 17.2 | 3 | 2 | 2.63 |
| 3.1 | 0 | 30 | 56 | 165 | 74.0 | 3 | 0 | 1.84 |
| 3.2 | 0* | 281 | 372 | 84 | 18.4 | 2 | 0 | 7.41 |
| 3.3 | 0* | 537 | 797 | 52 | 6.1 | 0* | 0 | 6.49 |
| 4.1 | 0 | 30 | 30 | 93 | 75.1 | 2 | 0 | 1.57 |
| 4.2 | 0* | 49 | 91 | 50 | 35.8 | 1 | 0 | 3.19 |
| 4.3 | 0* | 203 | 258 | 37 | 12.7 | 0* | 0 | 1.46 |

Table 6.4: Result summary per KPI of all corresponding test cases as described in Table 6.3
*Negligible (< 30 seconds)

### 6.4.2. Sensitivity analysis of absolute reduction of propagated delay to different test cases

This subsection investigates the sensitivity of absolute propagated delay reduction throughout the network, to the different test cases. This analysis will be conducted by comparing the output of different test cases and their delay scenarios, according to the following structure:

- **Test case 1 and test case 4** Analysing the test cases with only one delayed train
- **Test case 2 and test case 3** Analysing the test cases that have two delayed trains
- **Test case 1 and test case 2** Analysing the test cases with only sprinters having input delay
- **Test case 1-4** Analysing all test cases. After the assessment of the test cases as described above, comparing all test cases together might give extra insights

In this analysis, graphs will be presented that show the relation between initial delay and absolute delay reduction. The graphs distinguish the different test cases and their delay scenarios as formulated in Table 6.3. It should be noted that the delay scenarios are visualized in two ways, to provide results from the same test case in two different angles. The two types of presentation are follows:

- **Delay scenarios visualized in seconds** In these graphs, the amount of initial delay is given in seconds and is thus equal to the final column of table 6.3 for its corresponding scenario. In other words, the delay step (ph-30%, ph or ph+30%) is converted to the amount of seconds corresponding to that delay step.
- **Delay scenarios visualized as a step relative to planned headway** In these graphs, the amount of initial delay is visualized in planned headway step (ph-30%, ph, ph+30%) and is thus equal to the second column of table 6.3 for its corresponding scenario.

#### Test case 1 and test case 4

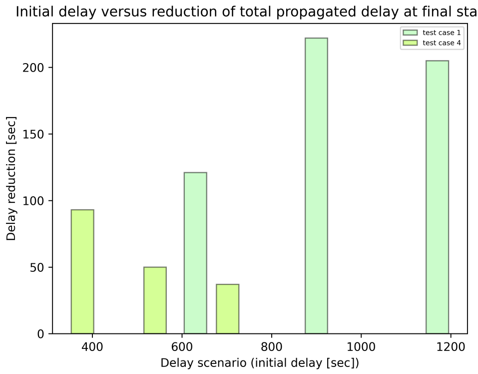


Figure 6.3: Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in seconds

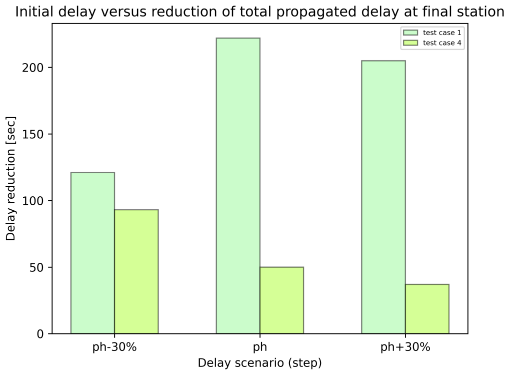


Figure 6.4: Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in step relative to planned headway

Figures 6.3 and 6.4 show the absolute delay reduction at the final arrival station in seconds for the corresponding test cases. In both figures, the y-axis displays the absolute delay reduction in seconds. As explained, the x-axis of Figure 6.3 displays the initial delay per scenario in seconds. In Figure 6.4, the x-axis displays the initial delay scenario as planned headway percentage.

It can be observed that for test case 4, the amount of delay reduction clearly reduces for higher input delays, whilst this trend is not so explicit for test case 1. The amount of absolute delay reduction is significantly higher for test case 1 compared to test case 4. This can be caused by two factors. First, the amount of input delay is higher for test case 1 than it is for test case 4, meaning there is more delay to reduce in the first place. Second, the conflict resolution measures of test case 1 affect more trains than the measures of test case 4, as can be deduced from Table 6.4, which can cause a higher rate of change compared to the unrescheduled situation and thus more delay reduction. Additionally, test case 1, with a delayed sprinter, applies reordering measures whilst test case 4, with a delayed intercity, does not.

### Test case 2 and test case 3

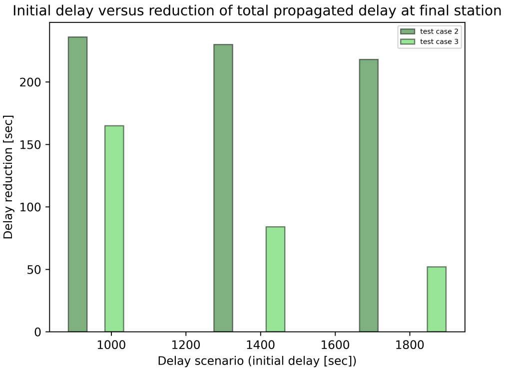


Figure 6.5: Sensitivity of absolute delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in seconds

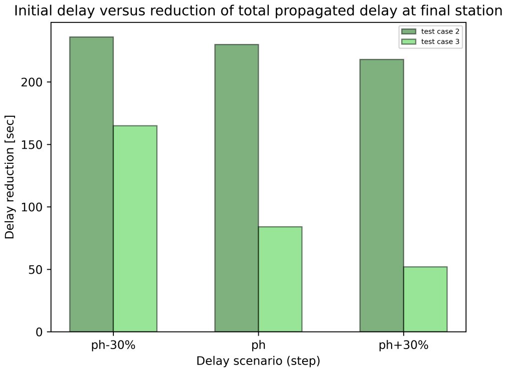


Figure 6.6: Sensitivity of absolute delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in step relative to planned headway

Figures 6.5 and 6.6 show the corresponding absolute delay reduction for test cases 2 and 3. When looking at both these test cases, test case 2 shows a slight decrease in delay reduction for higher initial delays. A more evident trend is that test case 2 yields higher delay reduction than test case 3. This is similar to what has been observed in the comparison between test cases 1 and 4. In both comparisons, the test case involving only initially delayed sprinters (cases 1 and 2) yield higher delay reduction than the test cases with delayed intercities (cases 3 and 4).

Again similar to the previous comparison, it can be observed in Table 6.4 that test case 2 imposes reordering as a rescheduling measure, whilst this is not implemented for test case 3. It thus seems that the effects of the model are most significant in cases 1 and 2, where reordering actions are taken. The fact that reordering takes place in test cases 1 and 2 and not in 3 and 4 could be explained in two ways. First of all, since intercities are faster trains with less stops, it is more easily beneficial to let an intercity precede in case of a delayed sprinter, and thus switch order. Whilst on the contrary, if an intercity is delayed, there is a higher chance that this train is still being held back by a slow sprinter in front of it, despite its initial delay, which could make reordering less desired. However, it could also be that the timetable properties under which this case study has been executed influence whether reordering is beneficial or not. Additionally, the advantages of reordering might change when considering a larger (or smaller) network. This could be examined further in a different case study that considers a corridor with different properties.

### Test case 1 and test case 2

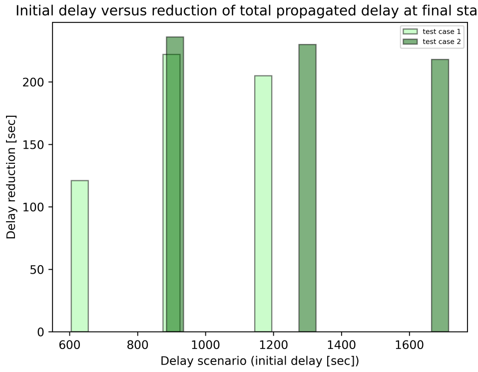


Figure 6.7: Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in seconds

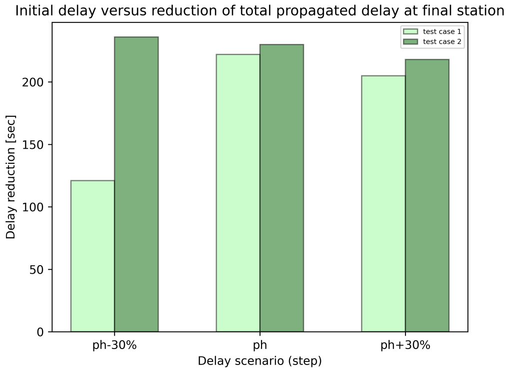


Figure 6.8: Sensitivity of absolute delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in step relative to planned headway

Figures 6.7 and 6.8 show the absolute delay reduction of test cases 1 and 2. Since these two test cases both consider only sprinters, it can be observed that in absolute numbers, the delay reduction of the two test cases is more similar than in the previous two comparisons. Whilst in test case 1, the absolute delay reduction increases from the first to the second delay scenario, this increase does not continue in the third delay scenario.

**Test case 1-4**

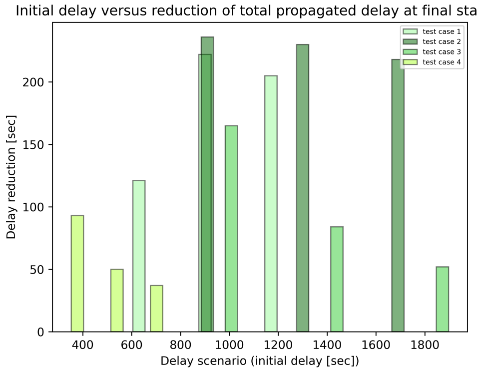


Figure 6.9: Sensitivity of absolute delay reduction to initial delay, comparing test cases 1-4, with delay scenarios visualized in seconds

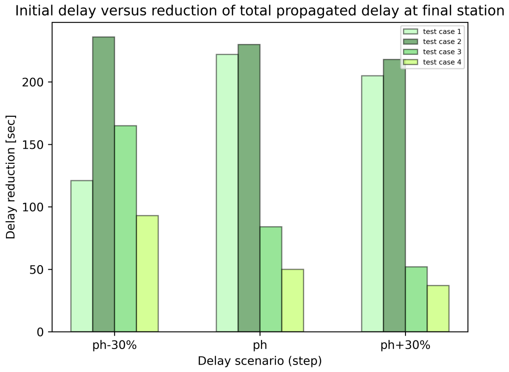


Figure 6.10: Sensitivity of absolute delay reduction to initial delay, comparing test cases 1-4, with delay scenarios visualized in step relative to planned headway

Figures 6.9 and 6.10 show the absolute delay reduction to the corresponding input delays for all four test cases. It can be seen from Figures 6.9 and 6.10 that delay reduction is highest for test case 2. It can be said that in absolute numbers, the model performs best for test case 2 in terms of delay reduction.

Test case 2 is the test case where the input delays are also high (second highest), the initially delayed trains are only sprinters and reordering is applied as one the rescheduling measures. It can be observed that absolute delay reduction is also high for test case 1, the case most similar to case 2. Overall, one main aspect can be concluded from the sensitivity analysis of absolute reduction of propagated delay:

- Absolute delay reduction is higher for test cases where reordering and retiming measures are applied than for test cases where only retiming measures are applied.

### 6.4.3. Sensitivity analysis of percentage reduction of propagated delay to different test cases

This subsection investigates the sensitivity of the percentage of propagated delay reduction. This analysis is a valuable addition to the previous subsection, since absolute numbers can not portray everything. The comparison of different test cases as well as the presentation of results is the same as in the previous subsection.

#### Test case 1 and test case 4

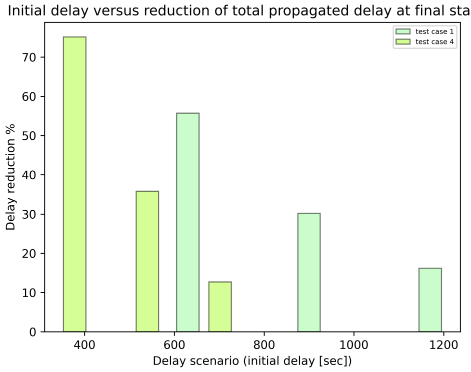


Figure 6.11: Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in seconds

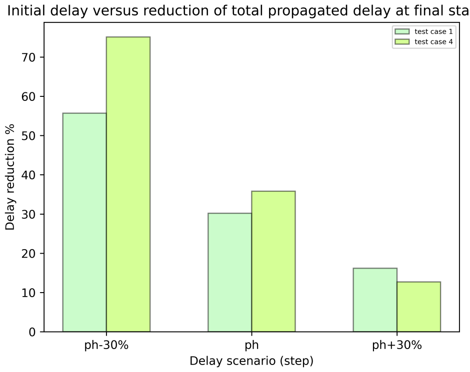


Figure 6.12: Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 4, with delay scenarios visualized in step relative to planned headway

Figures 6.11 and 6.12 show the delay reduction in percentage of both test cases 1 and 4. It can be noted that for both cases, the percentage delay reduction decreases as the initial delay increases. It can thus be concluded that, at least for test cases with only one initially delayed train, the model performs worse in terms of delay reduction in networks with a higher severity of delay. However, it does seem like a rather logical effect that a more gravely disturbed network is harder to recover. It could also be that the absolute delay reduction reaches a limit at some point, meaning the percentage delay reduction will eventually and inevitably decrease for higher input delays. Besides approaching a limit, the relation between initial delay and reduction of propagated delay seems non linear. However, definite conclusions can not be drawn about this in this research. In order to do so, additional analysis in more complex situations is necessary.

#### Test case 2 and test case 3

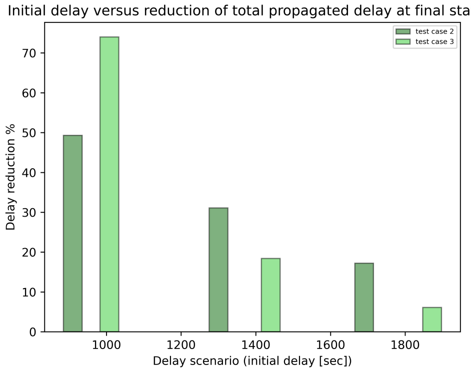


Figure 6.13: Sensitivity of percentage delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in seconds

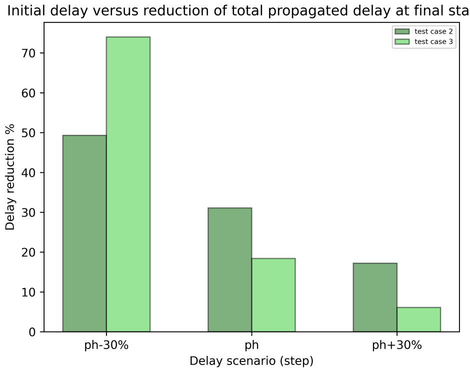


Figure 6.14: Sensitivity of percentage delay reduction to initial delay, comparing test cases 2 and 3, with delay scenarios visualized in step relative to planned headway

Looking at Figures 6.13 and 6.14, that show the percentage delay reduction for test cases 2 and 3, the same trends can also be seen. Namely, the fact that also in percentage, delay reduction drops as the initial conflict is more severe. For the first delay scenario of test case 3, ph-30%, the percentage delay reduction is very high. This could be explained by the fact that the initial delay in this scenario is not very high, especially for the intercity. Since intercities have less stops it could be that their initial delays have less consequences, and this conflict can be mitigated strongly with just a retiming measure.

### Test case 1 and test case 2

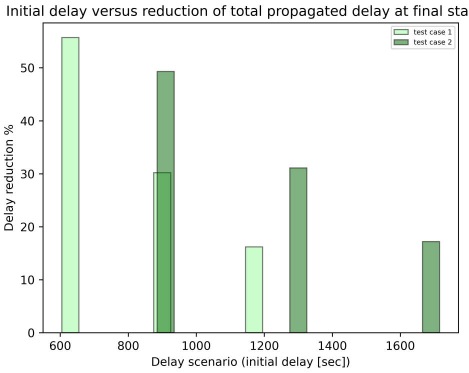


Figure 6.15: Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in seconds

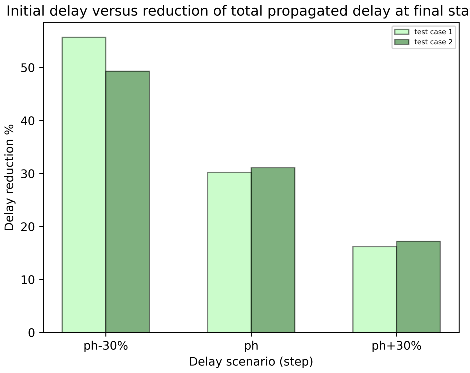


Figure 6.16: Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 2, with delay scenarios visualized in step relative to planned headway

Figures 6.15 and 6.16, show the same trend. The delay reduction decreases with initial delay increase. What is also interesting to note is that yest case 2 yields more delay reduction in absolute numbers, but looking at percentages this seems to drop heavily after the first delay scenario.

### Test cases 1-4

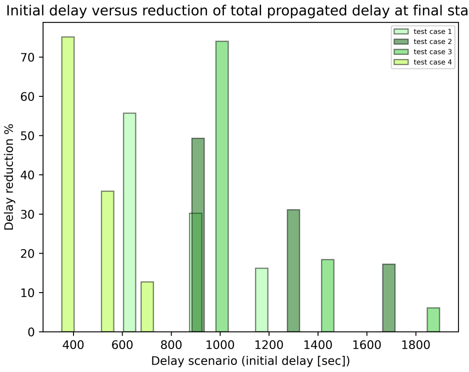


Figure 6.17: Sensitivity of percentage delay reduction to initial delay, comparing test cases 1-4, with delay scenarios visualized in seconds

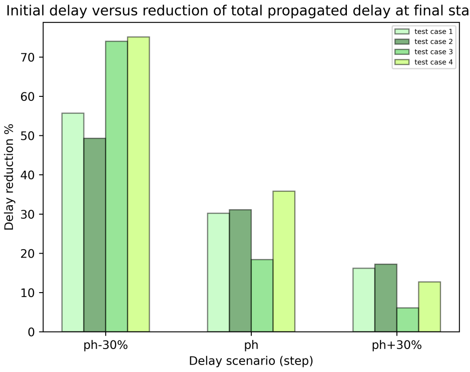


Figure 6.18: Sensitivity of percentage delay reduction to input delay, comparing test cases 1-4, with delay scenarios visualized in step relative to planned headway

Looking at Figures 6.17 and 6.18, it can be seen that for test cases 3 and 4 the first delay scenario corresponds to a very high delay reduction, but there is a large drop after that. For test case 1 and 2 this large drop is absent and the performance is similar, like already observed when comparing those two test cases. However overall, also in terms of percentual delay reduction, the model seems to perform best for test case 2.

When referring to the trends observed in the previous comparisons, the display of all four test cases confirms this. First, percentage delay reduction decreases for higher input delays. Second, the delay reduction seems to approach some limit. This latter statement can not be made definite within this research. However, it is in line with expectations, since generally when initial delays exceed 20 minutes, rerouting measures are to be included to solve the conflict. This has been described in Section 3.4.3, where state-of-the-art approaches where evaluateed. This provides an explanation for why higher initial delays approaching 20 minutes are harder to solve with only retiming and reordering measures. The main conslusions to be drawn from this analysis are:

- Percentage delay reduction decreases as initial delay increases.
- Percentage delay reduction seems to approach a limit as initial delay increases.

### 6.4.4. Sensitivity analysis of computation time to different test cases

This subsection will investigate the sensitivity of the computation time needed to solve the conflict resolution problem to the different input delays.

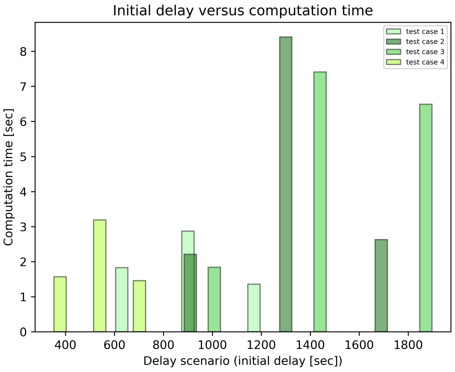


Figure 6.19: Sensitivity of computation time to initial delay, comparing test cases 1-4, with delay scenarios visualized in seconds

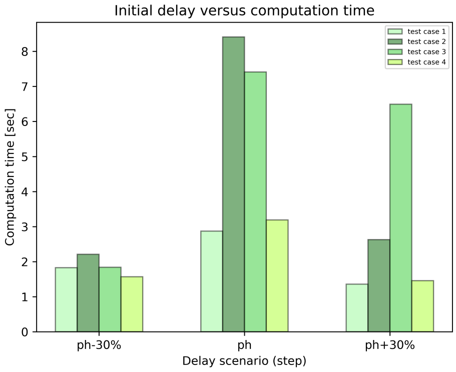


Figure 6.20: Sensitivity of computation time to initial delay, comparing test cases 1-4, with delay scenarios visualized in step relative to planned headway

Figures 6.19 and 6.20 show the test cases and their corresponding computation times for every delay scenario. First of all, it should be noted that all computation times are less than 10 seconds, which is considerably fast and suitable for real time implementations. However, in addition to that it should also be observed that the computation times tend to increase for higher input delays. This can be caused by increased problem complexity and more information to process. It can be observed that for test cases 2 and 3, which contain two initially delayed trains instead of one, the computation time is generally higher. This shows that computational complexity can thus vary, even within the same network size.

This case study has been performed on a relatively small corridor with only one hourly cycle of trains. More research needs to be performed in order to draw more clear conclusions on the computational performance. This could confirm the influence of increased number of initially delayed trains on the computation time. Additionally, further research could show how a larger network, which means more timeable points and thus an alternative graph with more nodes, influences the computation time. The trend from Figure 6.20 suggests that computation time might rise to unacceptable high levels, which can be problematic for real time optimization. Within this case study this conclusion can not yet be drawn, however this is a common trend observed in current literature as well, as has been described in Chapter 2 and Section 3.4.3. In the case of unacceptably high computation times, non-centralized options can be valuable to explore, as already mentioned in the introduction of this work. From this section it can concluded that:

- Computation times increase as initial delays increase.
- Computation times are higher for test cases with two delayed trains, i.e. higher situational complexity, than for test cases with only one delayed train.

## 6.5. Conclusion

After the execution of the case study, answers to research questions 4a-4d can be found. It can be said that the case study has been executed following a suitable approach and evaluated on relevant key performance indicators. This has been done by imposing various test cases on the model in order to see the variation in model performance behaviour. Thereby subquestion 4a has been answered. Regarding model performance, it can be concluded that the model performs well in reducing propagated delays. Retiming and reordering measures are taken for 10 out of 12 input cases, and in all cases, propagated delay is reduced with at least 6% and in some cases even up to 70%. All in all, for most cases the propagated delay reduction lies between 10% and 50%. The computation time of conflict resolution for the Rtd-Gvc corridor is under 10 seconds for all inputs. This answers subquestion 4b. Additionally, it can be said that the model yields to most significant results in cases where reordering measures are applied. Also it can be stated that higher input delays lead to less percentage delay reduction and higher computation times. However, since this model is designed for disturbances, meaning small input delays, the model is still considered suitable. An overview of the main conclusions found in this chapter is as follows:

- Absolute delay reduction is higher for test cases where reordering and retiming measures are applied than for test cases where only retiming measures are applied.
- Percentage delay reduction decreases as initial delay increases.
- Percentage delay reduction seems to approach a limit as initial delay increases.
- Computation times increase as initial delays increase.
- Computation times are higher for test cases with two delayed trains, i.e. higher situational complexity, than for test cases with only one delayed train.

These trends as observed in the case study are in line with phenomena observed in conflict resolution approaches in current literature, as has been described in Section 3.4.3, which answers subquestion 4c. All in all, it can be concluded that an appropriate model has been designed for conflict resolution under moving block signalling. In order to be able to compare this new model more accurately to existing methods in numerical terms, further research will first be necessary. This which will be elaborated on in the conclusion of this thesis.

# 7

# Conclusions and recommendations

Throughout this thesis, the goal has been to answer the research question ”How can centralized methods be designed to perform conflict resolution under moving block signalling?”. In order to provide an answer to this question, research has been performed on both current conflict resolution methods for fixed block systems and the working concepts, properties and characteristics of moving block signalling systems. This has led to a new mathematical approach based on alternative graph modelling, that has been extended from fixed block to moving block suitability. The novel mathematical formulation has been tested and verified using illustrative examples. Consequently, the performance of the model has been tested during a case study in the Rotterdam - The Hague corridor in the Netherlands, using data from a simulation environment. Conclusions can now be drawn by addressing and answering all research questions. A literature paper version of this thesis is attached as final appendix.

## 7.1. Addressing research subquestions

### 7.1.1. Subquestion 1: What is the current state of the art regarding centralized and non-centralized conflict resolution methods under moving block signalling?

Within the current state of the art, moving block approaches are underrepresented. This research has provided a very elaborate review on current literature regarding conflict resolution methods, moving block applications and non-centralized approaches within the railway sector. From this literature research it can first be concluded that conflict resolution approaches for moving block signalling need to be designed, to which this work has contributed. Second, it can be said that there are reasons to also investigate non centralized methods for moving block conflict resolution, which has not been elaborately researched in this work but will be further discussed in the recommendations.

### 7.1.2. Subquestion 2: What are the modelling characteristics of conflict resolution under moving block signalling?

Moving block signalling systems distinguish itself from fixed block signalling systems by several aspects. The most prominent changed properties of moving block compared to fixed block are the absence of trackside signals, the safety being maintained by constant monitoring of speed and position, and the potential for increased railway capacity. This means railway tracks are no longer designed for trains with the worst braking performance, and trains can have a tailored Movement Authority to their braking properties. With the knowledge of these properties, important assumptions and considerations have been made, in the form of taking a microscopic approach, taking into account disturbances in the network and focusing in reducing secondary delay. When designing a conflict resolution model, conform to these considerations, this can best be evaluated on three important KPI’s. First of all the total propagated delay, second of all the number of affected trains and third of all the computation time. This has led to the choice of developing a novel modelling approach based on the state-of-the-art Alternative Graph Method. This is based on a job shop scheduling problem, where the railway system is viewed as multiple machines interacting with each other. When modelling the railway network as an alternative graph, the corresponding objective is to minimize the maximum propagated delay in the network. This research proposes to extend this graph formulation with virtual nodes, that are related to trains rather than infrastructure. Additionally, conditional arcs are added to the model, that ensure a safe distance between trains at all times and obey to the train order established by alternative arcs. These additional aspects have made the modelling approach suitable for moving block signalling systems.

### 7.1.3. Subquestion 3: How can centralized conflict resolution under moving block signaling be performed?

Centralized conflict resolution can be performed by translating the current situation on the railway tracks to the mathematical formulation as described above. This has been done by implementing the modelling approach in an object oriented Python code. For this research, a suitable solution method has been chosen in the form of a commercial solver, after which a proof of concept has been designed. From this proof of concept, several conclusions can be drawn. First, this proof of concept has shown that the mathematical approach can distinguish static and virtual sections, by taking into account different arc weights and different node types. Second, it has been shown that the model can safely reschedule trains using retiming and reordering measures, by always respecting the formulated constraints. This includes the working principle of the conditional arcs, that always obey the train order and ensure a safe distance between two trains.

### 7.1.4. Subquestion 4: What are the impacts on conflict resolution performance for a centralized approach?

The performance of the mathematical approach has been investigated in a case study on the Rotterdam-The Hague corridor. A model has been created based on simulation data, to which a variety of inputs has been given. From the case study it can be concluded that for the majority of input situations, retiming and reordering measures can yield delay reductions of 10-50% within 10 seconds of computation time. The measures affect 2-4 out of 6 trains, mostly with retiming measures. The effects are strongest when reordering measures are also applied, but this is of course only done when this is beneficial in terms of delay. Subsequently, it can also be concluded that percentage delay reduction decreases and computation time increases as the input delay or situational complexity to the model increase. This could eventually cause the delay reduction to reach a limit and become unrewarding with respect to the amount of computation necessary the complete the rescheduling. However, since this model is designed for disturbances (maximum initial delay of 20 minutes), the model is considered suitable for its cause. The model developed in this research is not designed to resolve disruptions requiring replanning of crew and rolling stock resources.

## 7.2. Addressing main research question: How can centralized methods be designed to perform real time conflict resolution under moving block signaling?”

A new approach has been designed to perform conflict resolution under moving block signalling, that has been based on a valued state-of-the-art model, has been verified in a proof of concept and has been evaluated on performance during a case study. Overall, it can be concluded that a novel, verified and appropriate model has been designed to perform conflict resolution under moving block signalling, that is subjected to relevant assumptions and considerations. The designed extended alternative graph approach has been proven of great potential within the proof of concept. The case study has shown that the model can reduce propagated delays with 10-50% for the majority of input situations, within 10 seconds of computation time.

## 7.3. Recommendations for further research

Besides these current conclusions, this research contains several assumptions and there are aspects this work did not look into. Because of the high potential and suitability of this newly designed model, further research is both desired and highly advised. Therefore, a number of recommendations for future research will be drawn from this thesis. These recommendations can be divided into roughly two categories. First, recommendations following from the assumptions and scope boundaries that were set in this research. Second, recommendations related to the formulation and implementation of the mathematical model. Additionally, some extra recommendations will also be discussed.

### 7.3.1. Recommendations following from scope boundaries and assumptions

Regarding this first category, this research only evaluated retiming and reordering options for conflict resolution. It can be valuable to also investigate rerouting, cancelling and short turning, especially since there is already research regarding these measures for fixed block systems. Additionally, it can be beneficial to also include freight trains in the network and investigating their influence on delay propagation and rescheduling measures. The difference between freight and passenger trains is more severe than that between sprinters and intercities, so therefore the influence of varying speeds and number of stops might be more clear and more distinctive than they are in this research.

Another focus of further research could be to optimize the communication between the decision support system and the dispatcher, and investigating the trade-off between ambiguity and timely implementation of decisions, taking into account dispatchers preferences. Also, ProRail could investigate the rescheduling behaviour of this model to more KPI's that are tailored to their specific needs, like punctuality or energy use.

### 7.3.2. Recommendations regarding the mathematical approach

Regarding the second category, there are some recommendations to be made related to the mathematical model. First, in the alternative graph approach as applied in this research, the solution is not unique. There can be multiple rescheduling options that lead to the same value for maximum propagated delay. When improving this model, research can look into secondary objectives or additional constraints to give more guidance to possible rescheduling options. For example, prioritizing the option that leads the least use of energy. Second, due programming challenges the choice was made to assume the same length for all trains during the case study. This should be changed into various lengths, which creates options to investigate the length of trains on rescheduling measures.

Another recommendation follows from the assumptions made in this research when generating virtual nodes. This was based on train behaviour from simulations in current infrastructure. Future research could focus on accurate and dynamic node generation as well as prediction of arrival times at upcoming nodes. This can be done by frequently updating the current train speed and using this to subsequently calculate or generate new virtual nodes in a dynamic way and predict corresponding arrival times at those nodes. This would include making predictions over a time horizon, possibly in the form of Model Predictive Control (MPC), also called rolling horizon theory, where future states of a system are predicted over a time period and current states are then measured and given back to the system.

Related to this, the current optimization approach inspects a snapshot of the conflicted situation. However, for a more properly functioning traffic management system, this should be developed into a dynamic process that constantly updates the situation based on speed and position measurements. Additionally, the headway distances in this research were calculated using several guidelines and a prediction of virtual blocking times. For new researches, virtual blocking times could become more accurate and headway calculations can be improved and optimized.

Furthermore, in this thesis the optimization has been performed using a commercial solver. In the current ProRail traffic management system equipped with fixed block, tailored heuristics are used to reschedule trains. In future research, these heuristics (or algorithms) could be adjusted to fit the mathematical moving block approach as proposed in this work. In addition to that, the model implementation can be modified in such a way that an completely new alternative graph doesn't have to be constructed every time a conflict is detected, but the graph is remembered by the decision support system and only updated with new information.

Many of the preceding recommendations can be applied in additional case studies. New case studies can have larger networks or differently chosen corridors with more crossings, stations or a variety other situational characteristics. This could aid in confirming and extending the conclusions drawn from this research.

### 7.3.3. Recommendations for non-centralized approaches

As mentioned in the introduction of this research and as elaborately described within the state of the art review, investigating non-centralized options can be valuable. In the case study using a centralized method, computation times seem to go up as the situational complexity increases, which could cause problems for larger networks. However, this will first need more investigation to confirm this trend, although it can already be observed in current literature that this phenomenon generally occurs in many real time applications for large scale networks. Therefore, it is wise to investigate non-centralized approaches.

For non-centralized approaches, the system needs to be decomposed into subsystems. This can be done using mathematical tricks that create subproblems from a large scale problem. That would mean the entire graph is split into several, smaller graphs that can either be solved serial or in parallel. Another option is to divide the system geographically. In current practice, this is actually already being used, since the Dutch railway network is split in several local control areas. This is not fully automated or intelligent, but the structure can serve as a natural inspiration. This would mean all of these local areas have there own TMS that solves a graph separately.

A pitfall of making the system non-centralized is the risk of local solutions not being optimal on a global level, or even unfeasible. Therefore it can be wise to implement a multi-level non-centralized approach, with an overarching system above the subsystems. This creates hierarchy levels, where the overarching level has the highest rank. A first step can be to make this overarching system do feasibility checks on the local solutions. This can later be extended to also interfering with the optimality of local solutions.

For all different non-centralized approaches, it can be interesting to take something extra into account. Section 7.2.2 has described how MPC can be used to optimize the generation and predicted arrival times of virtual nodes. This principle can be combined with non-centralized approaches, meaning these predictions over a time horizon can be executed in a non-centralized manner. This creates options for a DMPC (distributed MPC) approach for conflict resolution.

### 7.3.4. Other recommendations

Finally, the job shop scheduling problem, which the Alternative Graph is based on, is not unique for railway problems. Job shop scheduling and planning theories are also applied in other fields. This thesis has provided a way to make the virtual beginning and end points of jobs, whilst still keeping a safe time interval between two jobs. This idea could be further investigated outside of the railway field and may provide useful insights for other applications.

# Appendices

# A

# Enlarged plots

## A.1. Illustrative example 1

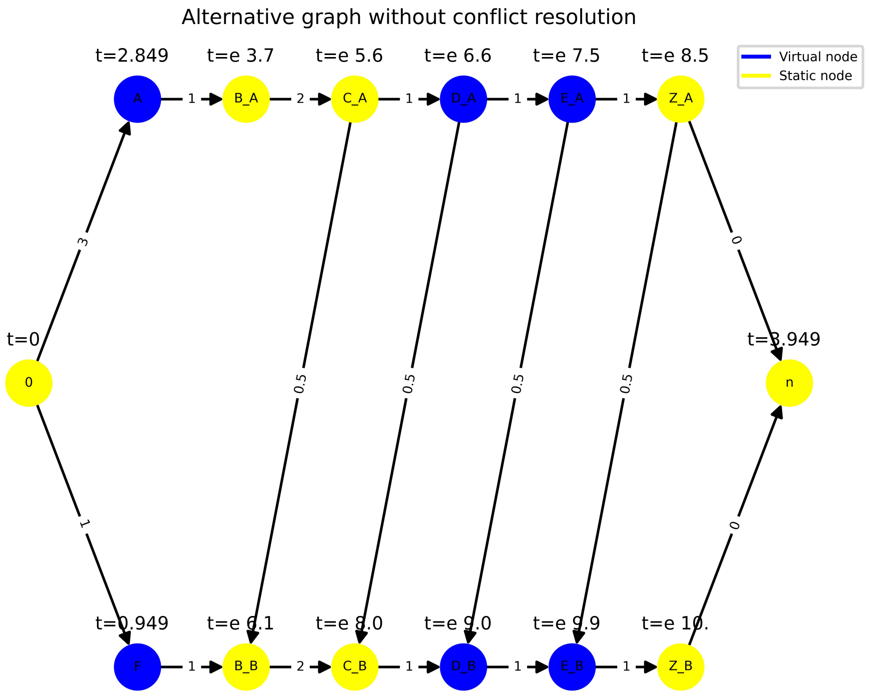


Figure A.1: Enlarged plot of illustrative example 1 without rescheduling

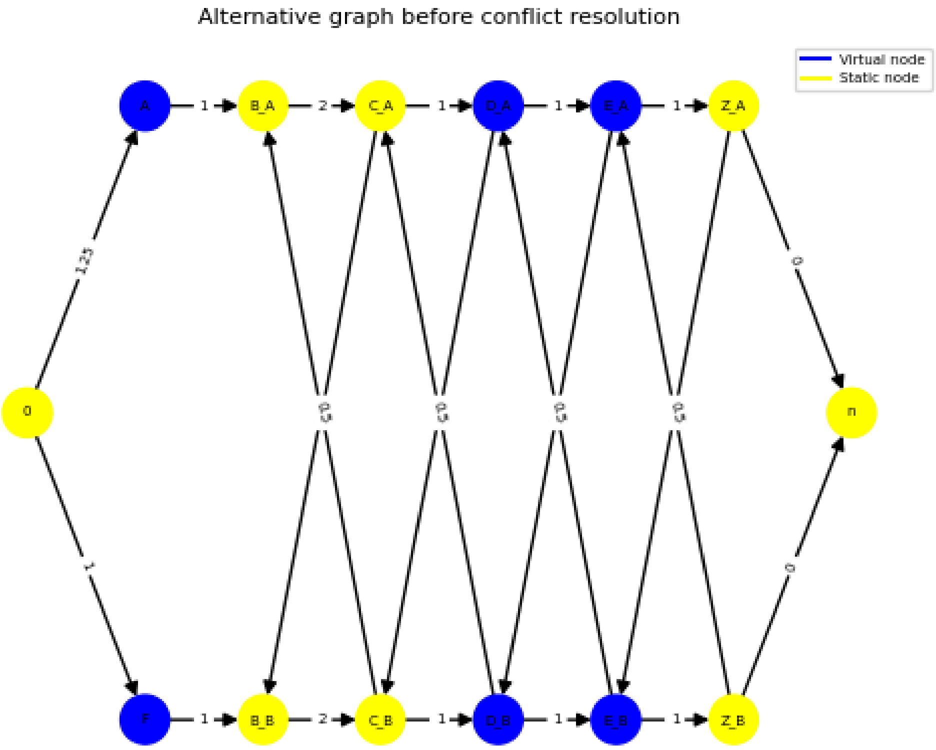


Figure A.2: Enlarged plot of illustrative example 1 before rescheduling

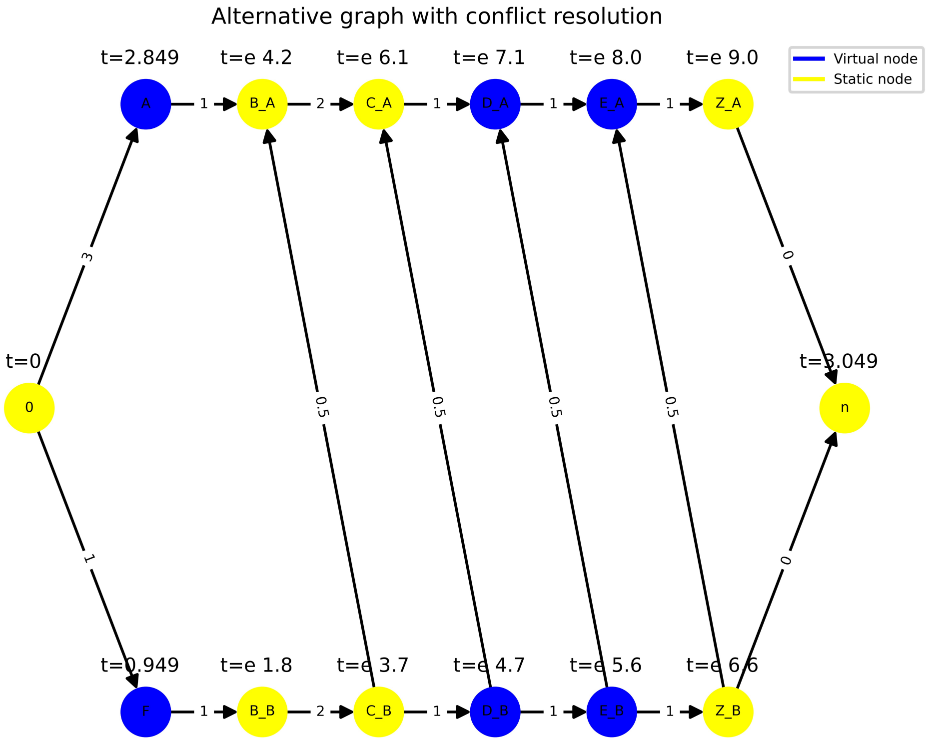


Figure A.3: Enlarged plot of illustrative example 1 with rescheduling

## A.2. Illustrative example 2

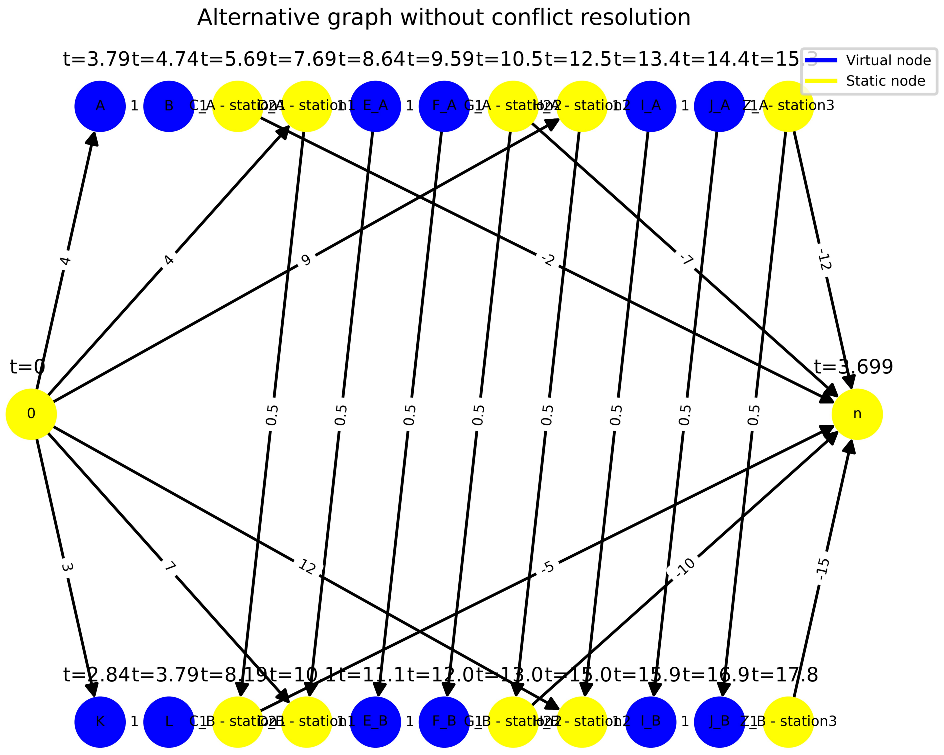


Figure A.4: Enlarged plot of illustrative example 2 without rescheduling

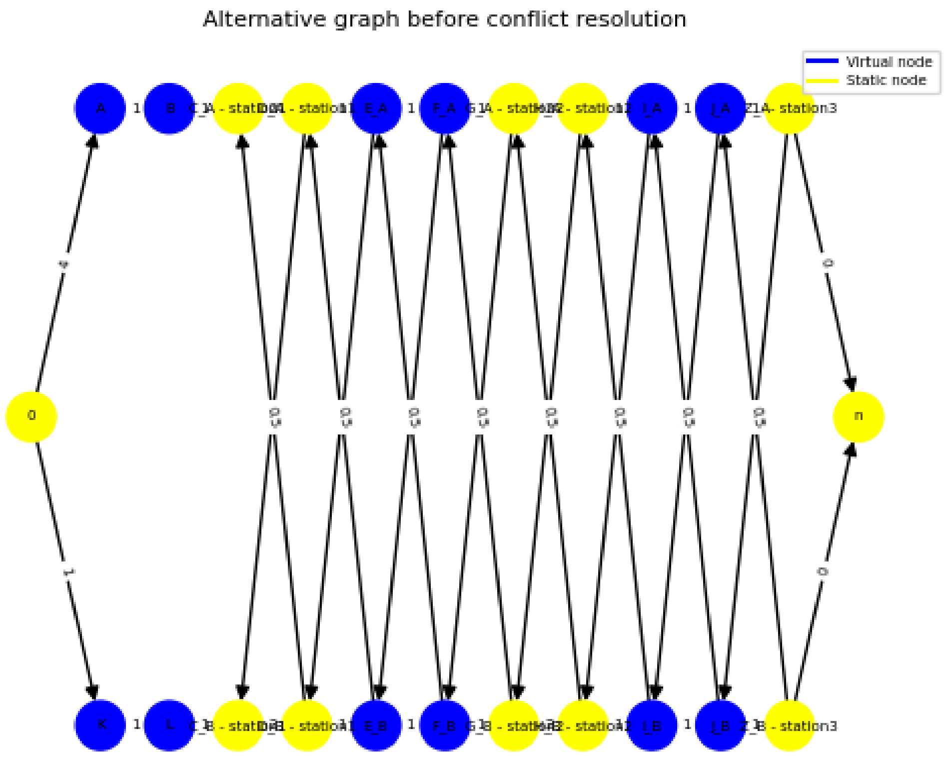


Figure A.5: Enlarged plot of illustrative example 2 before rescheduling

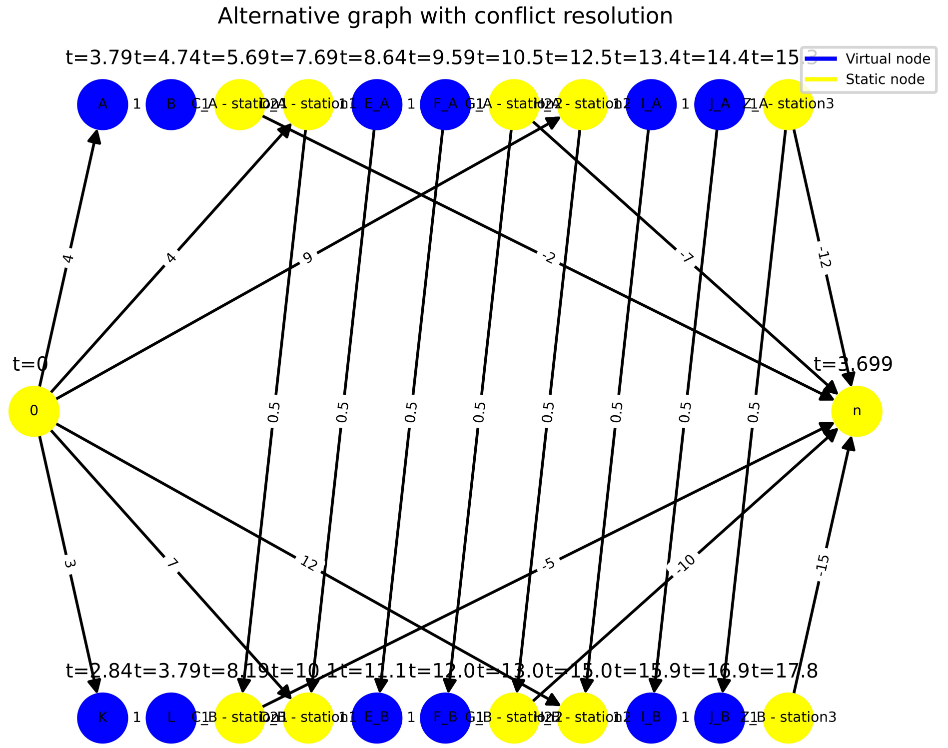


Figure A.6: Enlarged plot of illustrative example 2 with rescheduling

## A.3. Illustrative example 3

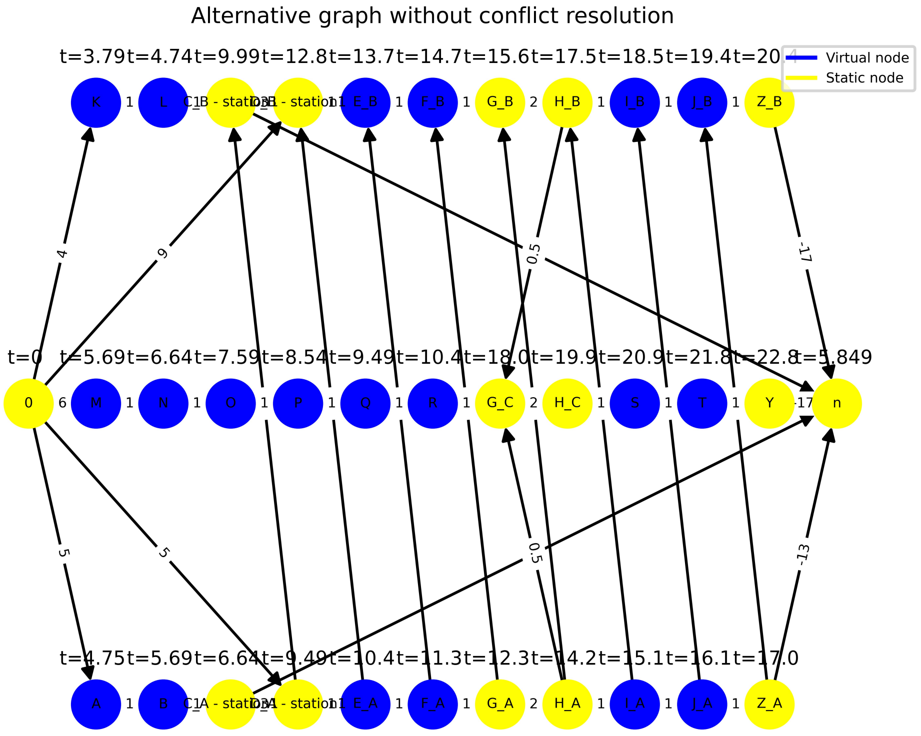


Figure A.7: Enlarged plot of illustrative example 3 without rescheduling

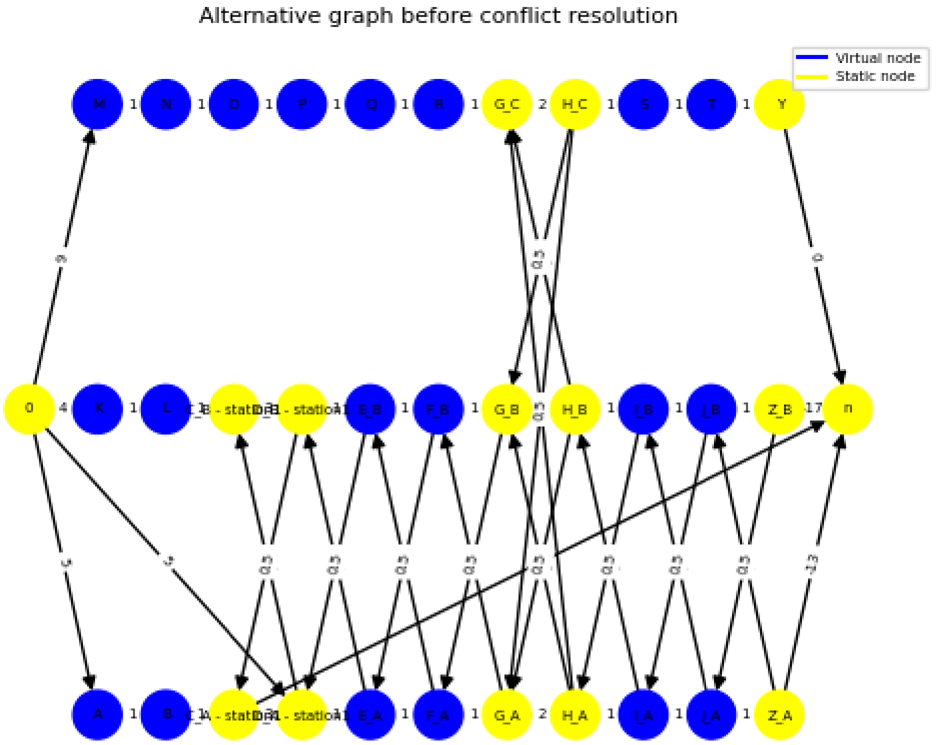


Figure A.8: Enlarged plot of illustrative example 3 before rescheduling

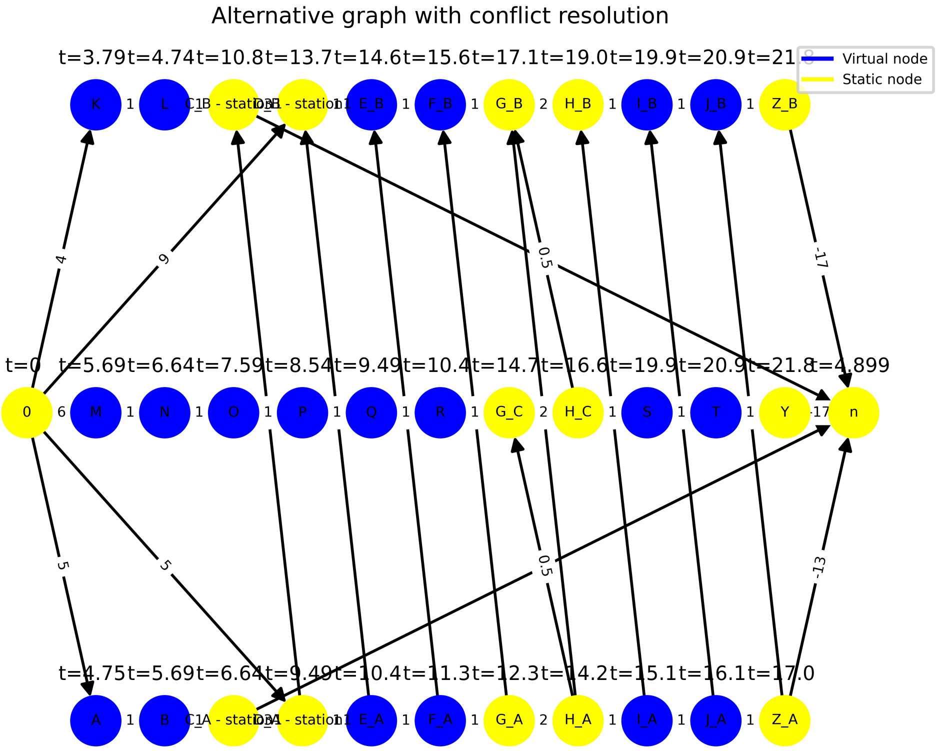


Figure A.9: Enlarged plot of illustrative example 3 with rescheduling

# B

# Additional information on trains and data in Rtd-Gvc corridor

## B.1. Additional train information

| Train | Train Type | Rolling stock | Length [m] | Route |
|---|---|---|---|---|
| A2400-H-1 | Intercity | VIRM10 | 270 | Rtd-Sdm-Dt-Gv-Laa-Ledn-Shl-Asdz-Dvd |
| A5000-H-1 | Sprinter | SGMm06 (3+3) | 158 | Rtd-Sdm-Dtcp-Dt-Rsw-Gvmw-Gv |
| A5100-H-1 | Sprinter | SGMm06 (3+3) | 158 | Rtd-Sdm-Dtcp-Dt-Rsw-Gvmw-Gv |
| C2400-H-1 | Intercity | VIRM10 | 270 | Rtd-Sdm-Dt-Gv-Laa-Ledn-Shl-Asdz-Dvd |
| C5000-H-1 | Sprinter | SGMm06 (3+3) | 158 | Rtd-Sdm-Dtcp-Dt-Rsw-Gvmw-Gv |
| C5100-H-1 | Sprinter | SGMm06 (3+3) | 158 | Rtd-Sdm-Dtcp-Dt-Rsw-Gvmw-Gv |

## B.2. Additional dataset information

**Types of information measured from FRISO simulation**

| Dutch | English |
|---|---|
| id | id |
| naam | name |
| materieelTypeId | rollingstock type id |
| beveiligingsType | security type |
| treinSerie | train series |
| treinType | train type |
| treinGebruik | train use |
| treinLengteM | train length (M) |
| vertrekprocedureTijd | time of departing procedure |
| bronDienstregeling | dispatching source |
| id2 | id 2 |
| naam3 | name 3 |
| type | type |
| lengteVanafVorigeBegrenzerM | length from previous limiter (M) |
| volgnummer | follow up number |
| richting | direction |
| kilometerLint | kilometer line |
| kilometrering | kilometrization |
| sectieNaam1 | section name 1 |
| dienstregelpuntCode | dispatching point code |

| | |
|---|---|
| voorkantPassageTijdstipS | front end passage time (S) |
| voorkantSnelheidMS | front end velocity (M/S) |
| voorkantVersnellingMS2 | front end acceleration (M/S^2) |
| voorkantIntervalS | front end interval (S) |
| remAfstandM | braking distance (M) |
| achterkantPassageTijdstipS | rear end passage time (S) |
| achterkantSnelheidMS | rear end velocity (M/S) |
| achterkantVersnellingMS2 | rear end acceleration (M/S^2) |
| achterkantIntervalS | rear end interval (S) |
| seinbeeld | signal perception |
| eindeRodeGolf | end Red Wave |
| isHoogSein | signal = high |
| sectieNaam2 | section name 2 |

**Types of timeable points logged from FRISO simulation**

| Dutch | English |
|---|---|
| SECTIE | Section |
| SPOOR | Platform/track |
| KAARTRAND | Map limit |
| DRGLPT_SPOOR | Dispatching point_track |
| DIENSTREGELPUNT_HART | Dispatching point_heart |
| SNELH_BD_M | Speed controller (?) |
| HOOGTE | Height point |
| LAS | Weld |
| WISSEL | Switch |
| LAS_DUB | Weld_double |
| GRENS | Limit |
| SNELH_BD_O | Speed controller |
| SNELH_BD_A | Speed controller |
| VERKENBORD | exploration board |
| HERHALING | recurrence |
| KNIK | buckle |
| BED_SEIN | control signal |
| AUT_SEIN_P | signal (?) |
| AHOB | ? |
| AUT_SEIN | signal (?) |
| VOORSEIN | front signal |
| BALISE | balise |
| BALISE_SCHAKEL | balise switch |
| LAS_ENK | weld_single |

# C

# Detailed case study results

## C.1. Situation 1

### C.1.1. Input step 1

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 630 | | | | | | | | | | |
| Sdm | A | 1172 | 568 | 1572 | 68 | 1842 | 0 | 2400 | -4 | 3292 | -12 | 3616 | -26 |
| | D | 1228 | 562 | 1628 | 62 | 1914 | 0 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1546 | 548 | 1946 | 48 | | 0 | 2816 | 18 | 3716 | 18 | | 0 |
| | D | 1609 | 541 | 2004 | 42 | | 0 | 2879 | 11 | 3774 | 12 | | 0 |
| Dt | A | 1728 | 535 | 2122 | 36 | 2312 | 6 | 2998 | 5 | 3892 | 6 | 4112 | 6 |
| | D | 1772 | 530 | 2172 | 30 | 2364 | 0 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1966 | 522 | 2385 | 42 | | 0 | 3260 | 17 | 4160 | 17 | | 0 |
| | D | 2011 | 517 | 2431 | 37 | | 0 | 3306 | 12 | 4206 | 12 | | 0 |
| Gvmw | A | 2131 | 511 | 2551 | 30 | | 0 | 3426 | 5 | 4326 | 5 | | 0 |
| | D | 2183 | 509 | 2601 | 27 | | 0 | 3476 | 2 | 4376 | 2 | | 0 |
| Gv | A | 2381 | 544 | 2780 | 43 | 2836 | 65 | 3678 | 41 | 4578 | 41 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 630 | | | | | | | | | | |
| Sdm | A | 1172 | 568 | 1572 | 68 | 1806 | -36 | 2373 | -31 | 3273 | -31 | 3611 | -31 |
| | D | 1228 | 562 | 1628 | 62 | 1914 | 0 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1546 | 548 | 1946 | 48 | | 0 | 2784 | -14 | 3704 | 6 | | 0 |
| | D | 1609 | 541 | 2004 | 42 | | 0 | 2868 | 0 | 3762 | 0 | | 0 |
| Dt | A | 1728 | 535 | 2122 | 36 | 2290 | -16 | 2986 | -6 | 3892 | 6 | 4112 | 6 |
| | D | 1772 | 530 | 2172 | 30 | 2364 | 0 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1966 | 522 | 2385 | 42 | | 0 | 3235 | -8 | 4148 | 5 | | 0 |
| | D | 2011 | 517 | 2431 | 37 | | 0 | 3294 | 0 | 4206 | 12 | | 0 |
| Gvmw | A | 2131 | 511 | 2551 | 30 | | 0 | 3414 | -6 | 4326 | 5 | | 0 |
| | D | 2180 | 506 | 2599 | 25 | | 0 | 3474 | 0 | 4374 | 0 | | 0 |
| Gv | A | 2343 | 505 | 2814 | 77 | 2758 | -12 | 3629 | -8 | 4568 | 31 | 4579 | 9 |

## C.1.2. Input step 2

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 900 | | | | | | | | | | |
| Sdm | A | 1429 | 825 | 1829 | 325 | 1975 | 132 | 2401 | -3 | 3294 | -10 | 3609 | -33 |
| | D | 1485 | 819 | 1885 | 319 | 2039 | 125 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1802 | 805 | 2202 | 305 | | 0 | 2816 | 18 | 3716 | 18 | | 0 |
| | D | 1866 | 798 | 2260 | 298 | | 0 | 2879 | 11 | 3774 | 12 | | 0 |
| Dt | A | 1984 | 792 | 2379 | 292 | 2519 | 213 | 2998 | 5 | 3892 | 6 | 4112 | 6 |
| | D | 2029 | 787 | 2429 | 287 | 2571 | 207 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2222 | 779 | 2641 | 298 | | 0 | 3260 | 17 | 4160 | 17 | | 0 |
| | D | 2268 | 774 | 2687 | 293 | | 0 | 3306 | 12 | 4206 | 12 | | 0 |
| Gvmw | A | 2388 | 768 | 2807 | 287 | | 0 | 3427 | 7 | 4327 | 7 | | 0 |
| | D | 2439 | 765 | 2857 | 283 | | 0 | 3477 | 3 | 4377 | 3 | | 0 |
| Gv | A | 2637 | 800 | 3036 | 299 | 3092 | 322 | 3680 | 42 | 4580 | 42 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 900 | | | | | | | | | | |
| Sdm | A | 1429 | 825 | 1829 | 325 | 1975 | 132 | 2373 | -31 | 3273 | -31 | 3611 | -31 |
| | D | 1485 | 819 | 1885 | 319 | 2039 | 125 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1802 | 805 | 2202 | 305 | | 0 | 2784 | -14 | 3704 | 6 | | 0 |
| | D | 1866 | 798 | 2260 | 298 | | 0 | 2868 | 0 | 3762 | 0 | | 0 |
| Dt | A | 1984 | 792 | 2379 | 292 | 2519 | 213 | 2986 | -6 | 3892 | 6 | 4112 | 6 |
| | D | 2029 | 787 | 2429 | 287 | 2571 | 207 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2222 | 779 | 2641 | 298 | | 0 | 3235 | -8 | 4148 | 5 | | 0 |
| | D | 2268 | 774 | 2687 | 293 | | 0 | 3294 | 0 | 4206 | 12 | | 0 |
| Gvmw | A | 2388 | 768 | 2807 | 287 | | 0 | 3414 | -6 | 4326 | 5 | | 0 |
| | D | 2436 | 762 | 2856 | 282 | | 0 | 3474 | 0 | 4374 | 0 | | 0 |
| Gv | A | 2599 | 762 | 3021 | 284 | 2965 | 195 | 3629 | -8 | 4570 | 33 | 4579 | 9 |

## C.1.3. Input step 3

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 1170 | | | | | | | | | | |
| Sdm | A | 1685 | 1081 | 2085 | 581 | 2231 | 389 | 2485 | 81 | 3294 | -10 | 3617 | -25 |
| | D | 1741 | 1075 | 2141 | 575 | 2296 | 382 | 2541 | 75 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 2059 | 1061 | 2459 | 561 | | 0 | 2859 | 61 | 3716 | 18 | | 0 |
| | D | 2122 | 1054 | 2517 | 555 | | 0 | 2922 | 54 | 3774 | 12 | | 0 |
| Dt | A | 2241 | 1048 | 2635 | 549 | 2775 | 469 | 3041 | 48 | 3892 | 6 | 4112 | 6 |
| | D | 2285 | 1043 | 2685 | 543 | 2828 | 464 | 3085 | 43 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2479 | 1035 | 2898 | 555 | | 0 | 3279 | 35 | 4160 | 17 | | 0 |
| | D | 2524 | 1030 | 2944 | 550 | | 0 | 3324 | 30 | 4206 | 12 | | 0 |
| Gvmw | A | 2647 | 1027 | 3064 | 543 | | 0 | 3444 | 24 | 4327 | 7 | | 0 |
| | D | 2699 | 1025 | 3114 | 540 | | 0 | 3494 | 20 | 4377 | 3 | | 0 |
| Gv | A | 2897 | 1060 | 3293 | 556 | 3349 | 578 | 3696 | 59 | 4580 | 42 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 1170 | | | | | | | | | | |
| Sdm | A | 1685 | 1081 | 2085 | 581 | 2231 | 389 | 2485 | 81 | 3273 | -31 | 3616 | -26 |
| | D | 1741 | 1075 | 2141 | 575 | 2296 | 382 | 2541 | 75 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 2059 | 1061 | 2459 | 561 | | 0 | 2859 | 61 | 3716 | 18 | | 0 |
| | D | 2122 | 1054 | 2517 | 555 | | 0 | 2922 | 54 | 3774 | 12 | | 0 |
| Dt | A | 2241 | 1048 | 2635 | 549 | 2775 | 469 | 3041 | 48 | 3892 | 6 | 4112 | 6 |
| | D | 2285 | 1043 | 2685 | 543 | 2828 | 464 | 3085 | 43 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2479 | 1035 | 2898 | 555 | | 0 | 3279 | 35 | 4160 | 17 | | 0 |
| | D | 2524 | 1030 | 2944 | 550 | | 0 | 3324 | 30 | 4206 | 12 | | 0 |
| Gvmw | A | 2644 | 1024 | 3064 | 543 | | 0 | 3444 | 24 | 4326 | 5 | | 0 |
| | D | 2693 | 1019 | 3112 | 538 | | 0 | 3493 | 19 | 4374 | 0 | | 0 |
| Gv | A | 2856 | 1018 | 3278 | 541 | 3222 | 451 | 3648 | 10 | 4576 | 39 | 4587 | 17 |

# C.2. Situation 2

## C.2.1. Input step 1

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 600 | | 100 | | | | | | | | |
| Sdm | A | 1172 | 568 | 1695 | 191 | 1849 | 7 | 2404 | 0 | 3298 | -6 | 3619 | -24 |
| | D | 1228 | 562 | 1751 | 185 | 1914 | 0 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1546 | 548 | 2068 | 171 | | 0 | 2816 | 18 | 3716 | 18 | | 0 |
| | D | 1609 | 541 | 2127 | 165 | | 0 | 2879 | 11 | 3774 | 12 | | 0 |
| Dt | A | 1728 | 535 | 2245 | 158 | 2385 | 79 | 2998 | 5 | 3892 | 6 | 4112 | 6 |
| | D | 1772 | 530 | 2295 | 153 | 2437 | 73 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1966 | 522 | 2507 | 164 | | 0 | 3260 | 17 | 4160 | 17 | | 0 |
| | D | 2011 | 517 | 2553 | 159 | | 0 | 3306 | 12 | 4206 | 12 | | 0 |
| Gvmw | A | 2137 | 517 | 2676 | 156 | | 0 | 3430 | 10 | 4330 | 10 | | 0 |
| | D | 2189 | 515 | 2726 | 152 | | 0 | 3480 | 6 | 4380 | 6 | | 0 |
| Gv | A | 2387 | 550 | 2906 | 169 | 2962 | 191 | 3683 | 46 | 4583 | 46 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 630 | | 280 | | | | | | | | |
| Sdm | A | 1172 | 568 | 1695 | 191 | 1841 | -2 | 2373 | -31 | 3273 | -31 | 3611 | -31 |
| | D | 1228 | 562 | 1751 | 185 | 1914 | 0 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1546 | 548 | 2068 | 171 | | 0 | 2784 | -14 | 3704 | 6 | | 0 |
| | D | 1609 | 541 | 2127 | 165 | | 0 | 2868 | 0 | 3762 | 0 | | 0 |
| Dt | A | 1728 | 535 | 2245 | 158 | 2385 | 79 | 2986 | -6 | 3892 | 6 | 4112 | 6 |
| | D | 1772 | 530 | 2295 | 153 | 2437 | 73 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1966 | 522 | 2507 | 164 | | 0 | 3235 | -8 | 4148 | 5 | | 0 |
| | D | 2011 | 517 | 2553 | 159 | | 0 | 3294 | 0 | 4206 | 12 | | 0 |
| Gvmw | A | 2131 | 511 | 2673 | 153 | | 0 | 3414 | -6 | 4326 | 5 | | 0 |
| | D | 2180 | 506 | 2722 | 148 | | 0 | 3474 | 0 | 4374 | 0 | | 0 |
| Gv | A | 2343 | 505 | 2887 | 150 | 2831 | 61 | 3629 | -8 | 4568 | 31 | 4579 | 9 |

### C.2.2. Input step 2

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 900 | | 400 | | | | | | | | |
| Sdm | A | 1429 | 825 | 1829 | 325 | 1975 | 132 | 2402 | -2 | 3296 | -8 | 3617 | -25 |
| | D | 1485 | 819 | 1885 | 319 | 2039 | 125 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1802 | 805 | 2202 | 305 | | 0 | 2816 | 18 | 3716 | 18 | | 0 |
| | D | 1866 | 798 | 2260 | 298 | | 0 | 2879 | 11 | 3774 | 12 | | 0 |
| Dt | A | 1984 | 792 | 2379 | 292 | 2519 | 213 | 2998 | 5 | 3892 | 6 | 4112 | 6 |
| | D | 2029 | 787 | 2429 | 287 | 2571 | 207 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2222 | 779 | 2641 | 298 | | 0 | 3260 | 17 | 4160 | 17 | | 0 |
| | D | 2268 | 774 | 2687 | 293 | | 0 | 3306 | 12 | 4206 | 12 | | 0 |
| Gvmw | A | 2391 | 771 | 2809 | 289 | | 0 | 3429 | 9 | 4329 | 9 | | 0 |
| | D | 2442 | 768 | 2859 | 285 | | 0 | 3479 | 5 | 4379 | 5 | | 0 |
| Gv | A | 2641 | 804 | 3038 | 301 | 3094 | 323 | 3681 | 44 | 4581 | 44 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 900 | | 400 | | | | | | | | |
| Sdm | A | 1429 | 825 | 1829 | 325 | 1975 | 132 | 2373 | -31 | 3273 | -31 | 3611 | -31 |
| | D | 1485 | 819 | 1885 | 319 | 2039 | 125 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1802 | 805 | 2202 | 305 | | 0 | 2784 | -14 | 3704 | 6 | | 0 |
| | D | 1866 | 798 | 2260 | 298 | | 0 | 2868 | 0 | 3762 | 0 | | 0 |
| Dt | A | 1984 | 792 | 2379 | 292 | 2519 | 213 | 2986 | -6 | 3892 | 6 | 4112 | 6 |
| | D | 2029 | 787 | 2429 | 287 | 2571 | 207 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2222 | 779 | 2641 | 298 | | 0 | 3235 | -8 | 4148 | 5 | | 0 |
| | D | 2268 | 774 | 2687 | 293 | | 0 | 3294 | 0 | 4206 | 12 | | 0 |
| Gvmw | A | 2388 | 768 | 2807 | 287 | | 0 | 3414 | -6 | 4326 | 5 | | 0 |
| | D | 2436 | 762 | 2856 | 282 | | 0 | 3474 | 0 | 4374 | 0 | | 0 |
| Gv | A | 2599 | 762 | 3021 | 284 | 2965 | 195 | 3629 | -8 | 4568 | 31 | 4579 | 9 |

### C.2.3. Input step 3

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 1170 | | 520 | | | | | | | | |
| Sdm | A | 1685 | 1081 | 2085 | 581 | 2231 | 389 | 2485 | 81 | 3294 | -10 | 3617 | -25 |
| | D | 1741 | 1075 | 2141 | 575 | 2296 | 382 | 2541 | 75 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 2059 | 1061 | 2459 | 561 | | 0 | 2859 | 61 | 3716 | 18 | | 0 |
| | D | 2122 | 1054 | 2517 | 555 | | 0 | 2922 | 54 | 3774 | 12 | | 0 |
| Dt | A | 2241 | 1048 | 2635 | 549 | 2775 | 469 | 3041 | 48 | 3892 | 6 | 4112 | 6 |
| | D | 2285 | 1043 | 2685 | 543 | 2828 | 464 | 3085 | 43 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2479 | 1035 | 2898 | 555 | | 0 | 3279 | 35 | 4160 | 17 | | 0 |
| | D | 2524 | 1030 | 2944 | 550 | | 0 | 3324 | 30 | 4206 | 12 | | 0 |
| Gvmw | A | 2647 | 1027 | 3065 | 545 | | 0 | 3444 | 24 | 4327 | 7 | | 0 |
| | D | 2699 | 1025 | 3115 | 541 | | 0 | 3494 | 20 | 4377 | 3 | | 0 |
| Gv | A | 2897 | 1060 | 3295 | 557 | 3350 | 580 | 3696 | 59 | 4580 | 42 | 4599 | 28 |

**Rescheduled**

| | | C5100 | *delay* | C5000 | *delay* | C2400 | *delay* | A5100 | *delay* | A5000 | *delay* | A2400 | *delay* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 1170 | | 520 | | | | | | | | |
| Sdm | A | 1685 | 1081 | 2085 | 581 | 2231 | 389 | 2485 | 81 | 3273 | -31 | 3613 | -29 |
| | D | 1741 | 1075 | 2141 | 575 | 2296 | 382 | 2541 | 75 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 2059 | 1061 | 2459 | 561 | | 0 | 2859 | 61 | 3704 | 6 | | 0 |
| | D | 2122 | 1054 | 2517 | 555 | | 0 | 2922 | 54 | 3772 | 10 | | 0 |
| Dt | A | 2241 | 1048 | 2635 | 549 | 2775 | 469 | 3041 | 48 | 3892 | 6 | 4112 | 6 |
| | D | 2285 | 1043 | 2685 | 543 | 2828 | 464 | 3085 | 43 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2479 | 1035 | 2898 | 555 | | 0 | 3279 | 35 | 4160 | 17 | | 0 |
| | D | 2524 | 1030 | 2944 | 550 | | 0 | 3324 | 30 | 4206 | 12 | | 0 |
| Gvmw | A | 2644 | 1024 | 3064 | 543 | | 0 | 3444 | 24 | 4326 | 5 | | 0 |
| | D | 2693 | 1019 | 3112 | 538 | | 0 | 3493 | 19 | 4374 | 0 | | 0 |
| Gv | A | 2856 | 1018 | 3278 | 541 | 3222 | 451 | 3648 | 10 | 4571 | 34 | 4583 | 12 |

# C.3. Situation 3

## C.3.1. Input step 1

**Unrescheduled**

| | | C5100 | *delay* | C5000 | *delay* | C2400 | *delay* | A5100 | *delay* | A5000 | *delay* | A2400 | *delay* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 600 | | | | 240 | | | | | | |
| Sdm | A | 1172 | 568 | 1572 | 68 | 2110 | 268 | 2410 | 6 | 3301 | -3 | 3620 | -23 |
| | D | 1228 | 562 | 1628 | 62 | 2175 | 261 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1546 | 548 | 1946 | 48 | | 0 | 2816 | 18 | 3716 | 18 | | 0 |
| | D | 1609 | 541 | 2004 | 42 | | 0 | 2879 | 11 | 3774 | 12 | | 0 |
| Dt | A | 1728 | 535 | 2122 | 36 | 2551 | 245 | 2998 | 5 | 3892 | 6 | 4112 | 6 |
| | D | 1772 | 530 | 2172 | 30 | 2603 | 239 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1966 | 522 | 2385 | 42 | | 0 | 3260 | 17 | 4160 | 17 | | 0 |
| | D | 2011 | 517 | 2431 | 37 | | 0 | 3306 | 12 | 4206 | 12 | | 0 |
| Gvmw | A | 2140 | 520 | 2559 | 39 | | 0 | 3432 | 12 | 4332 | 12 | | 0 |
| | D | 2192 | 518 | 2611 | 37 | | 0 | 3482 | 8 | 4382 | 8 | | 0 |
| Gv | A | 2391 | 553 | 2835 | 98 | 3046 | 275 | 3685 | 47 | 4585 | 47 | 4599 | 28 |

**Rescheduled**

| | | C5100 | *delay* | C5000 | *delay* | C2400 | *delay* | A5100 | *delay* | A5000 | *delay* | A2400 | *delay* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 600 | | | | 240 | | | | | | |
| Sdm | A | 1172 | 568 | 1572 | 68 | 2110 | 268 | 2373 | -31 | 3273 | -31 | 3611 | -31 |
| | D | 1228 | 562 | 1628 | 62 | 2175 | 261 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1546 | 548 | 1946 | 48 | | 0 | 2784 | -14 | 3704 | 6 | | 0 |
| | D | 1609 | 541 | 2004 | 42 | | 0 | 2868 | 0 | 3762 | 0 | | 0 |
| Dt | A | 1728 | 535 | 2122 | 36 | 2551 | 245 | 2986 | -6 | 3892 | 6 | 4112 | 6 |
| | D | 1772 | 530 | 2172 | 30 | 2603 | 239 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1966 | 522 | 2385 | 42 | | 0 | 3235 | -8 | 4148 | 5 | | 0 |
| | D | 2011 | 517 | 2431 | 37 | | 0 | 3294 | 0 | 4206 | 12 | | 0 |
| Gvmw | A | 2131 | 511 | 2551 | 30 | | 0 | 3414 | -6 | 4326 | 5 | | 0 |
| | D | 2180 | 506 | 2599 | 25 | | 0 | 3474 | 0 | 4374 | 0 | | 0 |
| Gv | A | 2343 | 505 | 2762 | 25 | 3023 | 252 | 3629 | -8 | 4568 | 31 | 4579 | 9 |

## C.3.2. Input step 2

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 900 | | | | 540 | | | | | | |
| Sdm | A | 1429 | 825 | 1829 | 325 | 2264 | 422 | 2525 | 121 | 3292 | -12 | 3616 | -26 |
| | D | 1485 | 819 | 1885 | 319 | 2328 | 414 | 2580 | 114 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1802 | 805 | 2202 | 305 | | 0 | 2898 | 101 | 3716 | 18 | | 0 |
| | D | 1866 | 798 | 2260 | 298 | | 0 | 2962 | 94 | 3774 | 12 | | 0 |
| Dt | A | 1984 | 792 | 2379 | 292 | 2705 | 399 | 3080 | 88 | 3892 | 6 | 4112 | 6 |
| | D | 2029 | 787 | 2429 | 287 | 2757 | 393 | 3125 | 83 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2222 | 779 | 2641 | 298 | | 0 | 3318 | 75 | 4160 | 17 | | 0 |
| | D | 2268 | 774 | 2687 | 293 | | 0 | 3364 | 70 | 4206 | 12 | | 0 |
| Gvmw | A | 2388 | 768 | 2807 | 287 | | 0 | 3484 | 63 | 4326 | 5 | | 0 |
| | D | 2436 | 762 | 2856 | 282 | | 0 | 3532 | 58 | 4376 | 2 | | 0 |
| Gv | A | 2634 | 797 | 3054 | 317 | 3193 | 422 | 3708 | 71 | 4578 | 41 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 1170 | | | | | | | | | | |
| Sdm | A | 1429 | 825 | 1589 | 85 | 2264 | 422 | 2418 | 14 | 3274 | -30 | 3611 | -31 |
| | D | 1485 | 819 | 1644 | 78 | 2328 | 414 | 2474 | 8 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1802 | 805 | 1962 | 65 | | 0 | 2805 | 7 | 3699 | 2 | | 0 |
| | D | 1866 | 798 | 2020 | 58 | | 0 | 2868 | 0 | 3774 | 12 | | 0 |
| Dt | A | 1984 | 792 | 2139 | 52 | 2705 | 399 | 2998 | 5 | 3892 | 6 | 4112 | 6 |
| | D | 2029 | 787 | 2189 | 47 | 2757 | 393 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2222 | 779 | 2401 | 58 | | 0 | 3248 | 5 | 4152 | 9 | | 0 |
| | D | 2268 | 774 | 2447 | 53 | | 0 | 3294 | 0 | 4206 | 12 | | 0 |
| Gvmw | A | 2388 | 768 | 2569 | 49 | | 0 | 3414 | -6 | 4326 | 5 | | 0 |
| | D | 2436 | 762 | 2619 | 45 | | 0 | 3474 | 0 | 4374 | 0 | | 0 |
| Gv | A | 2607 | 770 | 2795 | 58 | 3156 | 385 | 3629 | -8 | 4549 | 12 | 4577 | 6 |

## C.3.3. Input step 3

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 1170 | | | | 702 | | | | | | |
| Sdm | A | 1685 | 1081 | 2085 | 581 | 2418 | 576 | 2679 | 275 | 3292 | -12 | 3616 | -26 |
| | D | 1741 | 1075 | 2141 | 575 | 2482 | 568 | 2734 | 268 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 2059 | 1061 | 2459 | 561 | | 0 | 3052 | 255 | 3716 | 18 | | 0 |
| | D | 2122 | 1054 | 2517 | 555 | | 0 | 3116 | 248 | 3774 | 12 | | 0 |
| Dt | A | 2241 | 1048 | 2635 | 549 | 2858 | 553 | 3234 | 241 | 3892 | 6 | 4112 | 6 |
| | D | 2285 | 1043 | 2685 | 543 | 2911 | 547 | 3278 | 236 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2479 | 1035 | 2898 | 555 | | 0 | 3472 | 229 | 4160 | 17 | | 0 |
| | D | 2524 | 1030 | 2944 | 550 | | 0 | 3518 | 224 | 4206 | 12 | | 0 |
| Gvmw | A | 2644 | 1024 | 3064 | 543 | | 0 | 3637 | 217 | 4326 | 5 | | 0 |
| | D | 2696 | 1022 | 3114 | 540 | | 0 | 3686 | 212 | 4376 | 2 | | 0 |
| Gv | A | 2894 | 1057 | 3293 | 556 | 3349 | 578 | 3862 | 225 | 4578 | 41 | 4599 | 28 |

**Rescheduled**

| | | C5100 | delay | C5000 | delay | C2400 | delay | A5100 | delay | A5000 | delay | A2400 | delay |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | 1170 | | | | 702 | | | | | | |
| Sdm | A | 1685 | 1081 | 2085 | 581 | 2418 | 576 | 2679 | 275 | 3293 | -11 | 3616 | -26 |
| | D | 1741 | 1075 | 2141 | 575 | 2482 | 568 | 2734 | 268 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 2059 | 1061 | 2459 | 561 | | 0 | 3052 | 255 | 3716 | 18 | | 0 |
| | D | 2122 | 1054 | 2517 | 555 | | 0 | 3116 | 248 | 3774 | 12 | | 0 |
| Dt | A | 2241 | 1048 | 2635 | 549 | 2858 | 553 | 3234 | 241 | 3892 | 6 | 4112 | 6 |
| | D | 2285 | 1043 | 2685 | 543 | 2911 | 547 | 3278 | 236 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 2479 | 1035 | 2898 | 555 | | 0 | 3472 | 229 | 4160 | 17 | | 0 |
| | D | 2524 | 1030 | 2944 | 550 | | 0 | 3518 | 224 | 4206 | 12 | | 0 |
| Gvmw | A | 2644 | 1024 | 3064 | 543 | | 0 | 3637 | 217 | 4326 | 5 | | 0 |
| | D | 2693 | 1019 | 3112 | 538 | | 0 | 3686 | 212 | 4374 | 0 | | 0 |
| Gv | A | 2856 | 1018 | 3275 | 538 | 3331 | 560 | 3841 | 204 | 4576 | 39 | 4587 | 17 |

# C.4. Situation 4

## C.4.1. Input step 1

**Unrescheduled**

| | | C5100 | delay | C5000 | delay | C2400 | delay | A5100 | delay | A5000 | delay | A2400 | delay |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | | | | | 240 | | | | | | |
| Sdm | A | 610 | 6 | 1510 | 6 | 2110 | 268 | 2410 | 6 | 3301 | -3 | 3620 | -23 |
| | D | 666 | 0 | 1566 | 0 | 2175 | 261 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1016 | 18 | 1916 | 18 | | 0 | 2816 | 18 | 3716 | 18 | | 0 |
| | D | 1079 | 11 | 1974 | 12 | | 0 | 2879 | 11 | 3774 | 12 | | 0 |
| Dt | A | 1198 | 5 | 2092 | 6 | 2551 | 245 | 2998 | 5 | 3892 | 6 | 4112 | 6 |
| | D | 1242 | 0 | 2142 | 0 | 2603 | 239 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1460 | 17 | 2360 | 17 | | 0 | 3260 | 17 | 4160 | 17 | | 0 |
| | D | 1506 | 12 | 2406 | 12 | | 0 | 3306 | 12 | 4206 | 12 | | 0 |
| Gvmw | A | 1638 | 18 | 2536 | 16 | | 0 | 3432 | 12 | 4332 | 12 | | 0 |
| | D | 1689 | 15 | 2588 | 14 | | 0 | 3482 | 8 | 4382 | 8 | | 0 |
| Gv | A | 1914 | 77 | 2813 | 75 | 3038 | 268 | 3685 | 47 | 4585 | 47 | 4599 | 28 |

**Rescheduled**

| | | C5100 | delay | C5000 | delay | C2400 | delay | A5100 | delay | A5000 | delay | A2400 | delay |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | | | | | 240 | | | | | | |
| Sdm | A | 610 | 6 | 1473 | -31 | 2110 | 268 | 2373 | -31 | 3273 | -31 | 3611 | -32 |
| | D | 666 | 0 | 1566 | 0 | 2175 | 261 | 2466 | 0 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1016 | 18 | 1904 | 6 | | 0 | 2805 | 7 | 3704 | 6 | | 0 |
| | D | 1079 | 11 | 1962 | 0 | | 0 | 2868 | 0 | 3762 | 0 | | 0 |
| Dt | A | 1198 | 5 | 2080 | -6 | 2551 | 245 | 2986 | -6 | 3892 | 6 | 4112 | 6 |
| | D | 1242 | 0 | 2142 | 0 | 2603 | 239 | 3042 | 0 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1460 | 17 | 2355 | 12 | | 0 | 3235 | -8 | 4148 | 5 | | 0 |
| | D | 1506 | 12 | 2400 | 6 | | 0 | 3294 | 0 | 4194 | 0 | | 0 |
| Gvmw | A | 1626 | 5 | 2526 | 5 | | 0 | 3414 | -6 | 4326 | 5 | | 0 |
| | D | 1674 | 0 | 2574 | 0 | | 0 | 3474 | 0 | 4374 | 0 | | 0 |
| Gv | A | 1837 | 0 | 2766 | 29 | 2997 | 227 | 3629 | -8 | 4568 | 31 | 4578 | 7 |

## C.4.2. Input step 2

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | | | | | 540 | | | | | | |
| Sdm | A | 610 | 6 | 1505 | 1 | 2264 | 422 | 2525 | 121 | 3292 | -12 | 3616 | -26 |
| | D | 666 | 0 | 1566 | 0 | 2328 | 414 | 2580 | 114 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1016 | 18 | 1916 | 18 | | 0 | 2898 | 101 | 3716 | 18 | | 0 |
| | D | 1079 | 11 | 1974 | 12 | | 0 | 2962 | 94 | 3774 | 12 | | 0 |
| Dt | A | 1198 | 5 | 2092 | 6 | 2705 | 399 | 3080 | 88 | 3892 | 6 | 4112 | 6 |
| | D | 1242 | 0 | 2142 | 0 | 2757 | 393 | 3125 | 83 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1460 | 17 | 2360 | 17 | | 0 | 3318 | 75 | 4160 | 17 | | 0 |
| | D | 1506 | 12 | 2406 | 12 | | 0 | 3364 | 70 | 4206 | 12 | | 0 |
| Gvmw | A | 1626 | 5 | 2526 | 5 | | 0 | 3484 | 63 | 4326 | 5 | | 0 |
| | D | 1677 | 3 | 2576 | 2 | | 0 | 3534 | 60 | 4376 | 2 | | 0 |
| Gv | A | 1901 | 64 | 2800 | 63 | 3192 | 422 | 3710 | 73 | 4578 | 41 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | | | | | 540 | | | | | | |
| Sdm | A | 610 | 6 | 1473 | -31 | 2264 | 422 | 2525 | 121 | 3273 | -31 | 3612 | -30 |
| | D | 666 | 0 | 1566 | 0 | 2328 | 414 | 2580 | 114 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1014 | 17 | 1904 | 6 | | 0 | 2898 | 101 | 3704 | 6 | | 0 |
| | D | 1079 | 11 | 1962 | 0 | | 0 | 2962 | 94 | 3762 | 0 | | 0 |
| Dt | A | 1198 | 5 | 2080 | -6 | 2705 | 399 | 3080 | 88 | 3892 | 6 | 4112 | 6 |
| | D | 1242 | 0 | 2142 | 0 | 2757 | 393 | 3125 | 83 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1460 | 17 | 2355 | 12 | | 0 | 3318 | 75 | 4148 | 5 | | 0 |
| | D | 1506 | 12 | 2400 | 6 | | 0 | 3364 | 70 | 4206 | 12 | | 0 |
| Gvmw | A | 1626 | 5 | 2526 | 5 | | 0 | 3484 | 63 | 4326 | 5 | | 0 |
| | D | 1674 | 0 | 2574 | 0 | | 0 | 3532 | 58 | 4374 | 0 | | 0 |
| Gv | A | 1837 | 0 | 2766 | 29 | 3151 | 381 | 3687 | 50 | 4570 | 33 | 4579 | 9 |

## C.4.3. Input step 3

**Unrescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | | | | | 702 | | | | | | |
| Sdm | A | 610 | 6 | 1505 | 1 | 2418 | 576 | 2679 | 275 | 3292 | -12 | 3616 | -26 |
| | D | 666 | 0 | 1566 | 0 | 2482 | 568 | 2734 | 268 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1016 | 18 | 1916 | 18 | | 0 | 3052 | 255 | 3716 | 18 | | 0 |
| | D | 1079 | 11 | 1974 | 12 | | 0 | 3116 | 248 | 3774 | 12 | | 0 |
| Dt | A | 1198 | 5 | 2092 | 6 | 2858 | 553 | 3234 | 241 | 3892 | 6 | 4112 | 6 |
| | D | 1242 | 0 | 2142 | 0 | 2911 | 547 | 3278 | 236 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1460 | 17 | 2360 | 17 | | 0 | 3472 | 229 | 4160 | 17 | | 0 |
| | D | 1506 | 12 | 2406 | 12 | | 0 | 3518 | 224 | 4206 | 12 | | 0 |
| Gvmw | A | 1626 | 5 | 2526 | 5 | | 0 | 3637 | 217 | 4326 | 5 | | 0 |
| | D | 1677 | 3 | 2576 | 2 | | 0 | 3688 | 214 | 4376 | 2 | | 0 |
| Gv | A | 1901 | 64 | 2800 | 63 | 3346 | 576 | 3864 | 227 | 4578 | 41 | 4599 | 28 |

**Rescheduled**

| | | **C5100** | ***delay*** | **C5000** | ***delay*** | **C2400** | ***delay*** | **A5100** | ***delay*** | **A5000** | ***delay*** | **A2400** | ***delay*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rtd | D | | | | | | 702 | | | | | | |
| Sdm | A | 610 | 6 | 1489 | -15 | 2418 | 576 | 2679 | 275 | 3291 | -13 | 3615 | -27 |
| | D | 666 | 0 | 1566 | 0 | 2482 | 568 | 2734 | 268 | 3366 | 0 | 3714 | 0 |
| Dtcp | A | 1016 | 18 | 1904 | 6 | | 0 | 3052 | 255 | 3716 | 18 | | 0 |
| | D | 1079 | 11 | 1962 | 0 | | 0 | 3116 | 248 | 3774 | 12 | | 0 |
| Dt | A | 1198 | 5 | 2080 | -6 | 2858 | 553 | 3234 | 241 | 3892 | 6 | 4112 | 6 |
| | D | 1242 | 0 | 2142 | 0 | 2911 | 547 | 3278 | 236 | 3942 | 0 | 4164 | 0 |
| Rsw | A | 1460 | 17 | 2355 | 12 | | 0 | 3472 | 229 | 4160 | 17 | | 0 |
| | D | 1506 | 12 | 2400 | 6 | | 0 | 3518 | 224 | 4206 | 12 | | 0 |
| Gvmw | A | 1626 | 5 | 2526 | 5 | | 0 | 3637 | 217 | 4326 | 5 | | 0 |
| | D | 1674 | 0 | 2574 | 0 | | 0 | 3686 | 212 | 4374 | 0 | | 0 |
| Gv | A | 1837 | 0 | 2737 | 0 | 3331 | 560 | 3841 | 204 | 4576 | 39 | 4586 | 15 |

# D

# Python functions

## D.1. Function to initialize node values

```
def get_node_value(self, node):

nodes = self.nodes
edges = self.edges
edges = list(itertools.chain(*edges))
node = node
ind = self.nodes.index(node)

for i in range(0, len(self.nodes)):
    setattr(self.nodes[i], 'value', 0)
for i in range(0, ind-1):
    nodes[i+1].value = nodes[i].value + edges[i].get_weight()
    node.value = nodes[i+1].value
    return node.value
```

## D.2. Function to determine minimum headways

**Determine train speed at current node**

```
 def v(train, node):
    train_length = train.length
    train = train
    node = node

    t_cl_1 = 0

    for edge_list in train.edges:
        for edge in edge_list:
            if edge.src == node:
                if node == train.nodes[-1]:
                    t_cl_1 = 30
                else:                               t_cl_1 = edge.weight
                    v = train_length/t_cl_1
                    return v
```

**Minimum headway between static nodes**

```
def headway_static(source, destination):
    src = source
```

```
    dest = destination
    train_1 = None
    train_2 = None
    edge_1 = None
    t_rel = 3
    t_set = 7
    t_sight = 6
    dec = 0.5

    for train in trains:
        if src in train.nodes:
            train_1 = train
        if dest in train.nodes:
            train_2 = train
            for edgelist in train_1.edges:
                for edge in edgelist:
                    if src == edge.src:
                        if edge.dest == train_1.nodes[-1]:
                            continue
                        else:
                            edge_1 = edge

    t_run = 100/v(train_1, src)
    t_cl = edge_1.weight
    t_app = v(train_1, src)/dec + 50/v(train_1, src)

    hdwy = t_run + t_cl + t_app + t_rel + t_set + t_sight

    return hdwy
```

**Minimum headway between virtual nodes**

```
def headway_virtual(source, destination):
    src = source
    dest = destination
    train_1 = None
    train_2 = None
    edge_1 = None
    t_rel = 3
    t_sight = 6
    dec = 0.5

    for train in trains:
        if src in train.nodes:
            train_1 = train
        if dest in train.nodes:
            train_2 = train
            for edgelist in train_1.edges:
                for edge in edgelist:
                    if src == edge.src:
                        edge_1 = edge

    t_cl = edge_1.weight
    t_app = v(train_1, src)/dec

    hdwy = t_cl + t_app + t_rel + t_sight
```

```
    return hdwy
```

## D.3. Function to create alternative arcs

```
def create_alt_arcs(nodes_1, nodes_2):

nodes_1 = nodes_1
nodes_2 = nodes_2
Alt_arcs = []
Alt_arc_pairs_nr = len(nodes_1) - 1

for i in range(0, Alt_arc_pairs_nr):
    Alt_arcs.append([])
    Alt_arc_pair = []

for i in range(0, len(nodes_1) - 1):
    Arc_1 = Edge(nodes_2[i + 1], nodes_1[i], headway_weight, 'Conditional', 'Headway')
    Alt_arc_pair.append(Arc_1)
    Alt_arcs[i].append(Arc_1)
    Arc_2 = Edge(nodes_1[i + 1], nodes_2[i], headway_weight, 'Conditional', 'Headway')
    Alt_arc_pair.append(Arc_2)
    Alt_arcs[i].append(Arc_2)

    for pair in Alt_arcs:
        if pair[0].src.type == 'static' and pair[0].dest.type == 'static':
        pair[0].__setattr__('type', 'Alternative')
        pair[1].__setattr__('type', 'Alternative')

return Alt_arcs
```

## D.4. Function to establish conditional arc obedience

```
def z_ind(a_r_c):

a_r_c = a_r_c

corresponding_train = None
for i in range(0, len(trains)):
    if a_r_c[0].src in trains[i].nodes:
        corresponding_train = trains[i]

for j in range(0, len(Alt_arcs_alt) - 1):
    if Alt_arcs_alt[j][0].src not in corresponding_train.nodes:
        continue
    else:
        if corresponding_train.nodes.index(Alt_arcs_alt[j][0].src) < corresponding_train.
        < corresponding_train.nodes.index(Alt_arcs_alt[j + 1][0].src):
                z_index = j
                return z_index
            else:
                z_index = j + 1
                return z_index
```

# A Multi Machine Approach for Conflict Resolution under Moving Block Signalling Systems

Mathilde Janssens, Vasso Reppa, Egidio Quaglietta, Dick Middelkoop, and Rudy Negenborn

***Abstract*—Railway networks are to play an increasingly large role in European transportation. This has boosted the urgency of railway innovations, of which the development of decision support systems for conflict resolution is an important aspect. This paper contributes to this development by formulating a suitable mathematical approach for railway networks equipped with moving block signalling systems. Two dispatching actions to reschedule trains are applied, namely retiming and reordering. The designed approach is an extension to an existing method, based on graph theory, that is able to reschedule trains in case of conflict. The novel method uses additional node- and arc types in order to ensure moving block suitability. The new node type enables the possibility to create nodes that are related to trains, rather than infrastructure. The new arc type ensures a continuously safe time interval between two trains in the absence of trackside signals. An optimization problem, with the objective of minimizing the maximum propagated delay, is formulated. Hereafter, the performance is evaluated by a case study in the Rotterdam-The Hague corridor. According to the experimental results, the designed model is able to reduce delay propagation up to 50% for the majority of input situations within 10 seconds of computation time. Overall, the designed method shows promising results, but further research will be necessary to make it applicable in practice.**

***Index Terms*—Railway traffic management, moving block signalling, alternative graph theory, railway conflict resolution, retiming, reordering, ERTMS**

## I. INTRODUCTION

THE development of railway transportation has been boosted to a higher urgency during the last few years. The International Energy Agency (IEA) has expressed the need to increase the role of railway transportation for both freight and passengers, due to two prominent reasons [29]. First, because railway transport is recognized as one of the most sustainable ways of travel to this day, being accountable for less than 0.5% of European Green House Gas emissions [28]. In comparison, road traffic is responsible for over 70%. Second, because of the general increase in transport demand that followed from population growth and increasing globalization and urbanisation [29]. In 2021, the European Union declared that year to be ”The Year of Rail”, throughout which the benefits of sustainable, safe, and smart transportation would be continuously highlighted [28].

The increased use of railway transport can be supported by building new infrastructure. However, since this is a very time intensive task, innovations are also directed towards the increase of railway capacity and efficiency with the current infrastructural resources. Part of these innovations concern smart solutions in railway traffic management.

A key part of railway traffic management is conflict resolution. Since railway networks are large and complex systems, subjected to disturbances, conflicts are an inevitable phenomenon in railway operations. Conflicts come in different types, one of which is a *track occupation conflict*. This is the type of conflict considered in this paper and occurs when two trains claim the same infrastructural resource simultaneously [23], mostly due to an *initial delay* (or: primary delay). This form of delay is caused by the fact that a railway process, such as driving from one station to another or dwelling at a station, lasts longer than planned [20]. This conflict leads to undesired effects, mostly in the form of a *propagated delay* (or: secondary delay). This is delay caused by primary delays being passed on to another train. Resolving conflicts is necessary and desired to regain schedule feasibility and restore operational efficiency. Conflict resolution is currently handled by railway dispatchers, who use predefined plans and human expertise. However, this may not always result in the best solution. Therefore, dispatchers can be aided by *decision support systems*. Decision support systems are, as defined by the authors of [30], computer-based interactive systems that support decision makers rather than replace them, utilising data and models with varying degrees of structure. This paper aims at designing a decision support system for conflict resolution under moving block signalling.

Railway rescheduling is also known as the Real Time Railway Traffic Management problem (RTRTM problem) [13]. Since conflicts are detected during the execution of operations, solutions need to be generated real time. This means that the time elapsed between the moment of conflict detection and the moment of action implementation influences the performance of the decision support system [11].

An important property to be taken into account when designing a decision support system for railway applications is the safety and signalling system of the railway network. The European Union is working on an overarching traffic management standard for all European countries: the *European Railway Traffic Management System (ERTMS)*. Three levels of ERTMS have currently been designed, where level 1 and 2 are equipped with a traditional fixed block signalling system and level 3 is equipped with a moving block signalling system [8]. The most fundamental differences between fixed- and moving block signalling are the following:

- In a fixed block signalling system, the railway network is divided into sections of fixed length (blocks), which can

by occupied by one train at a time. In a moving block signalling system, no such sections are present.
- In a fixed block signalling system, the fixed sections are guarded by signals to ensure safety throughout the network. A train can only enter a section following a green signal, implying no other train is present in that section. In a moving block signalling system, these trackside signals are replaced by on board monitoring devices, continuously measuring the trains' speed and position. This is used to calculate the safe distance between two trains and its corresponding speed profiles. These are communicated to the trains, ensuring safety throughout the network.
- In a fixed block signalling system, the maximum capacity is thus defined by the number of block sections. In a moving block signalling system, the maximum capacity is defined by the minimum headway between two trains, equal to the absolute braking distance plus a safety margin

These characteristics make that moving block signalling systems come with several significant advantages. First, operational costs can be reduced due to savings in track side maintenance [8]. Second, traction energy can be reduced because there are less unplanned stops and speed profiles are more flattened [26]. Additionally, there is potential for increased traffic fluidity, driving at higher speeds, and more competitiveness with other transport modes [31], [14], [8]. For more extensive explanations on moving block signalling systems, its properties, and its potential, refer to the COMBINE projects (COntrol center for a Moving Block signallINg systEm) [8].

This paper describes the development of a new approach for conflict resolution under a moving block signalling system. This has not been done before in current literature. The objective of the conflict resolution approach will be equal to minimizing the maximum secondary delay in the network. This objective has already been deployed in a state-of-the-art method as proposed by Corman and D'Ariano [4], namely alternative graph theory. The newly designed approach is an extension to this existing method, from fixed block to moving block suitability.

This paper is organized as follows. Section II describes scientific works related to this topic. Section III provides the theoretical fundamentals of the alternative graph used to construct the mathematical formulation. Section IV presents the optimization problem used to perform the rescheduling. Section V introduces a case study in the Rotterdam-The Hague corridor. Section VI displays the results from this case study and evaluates the model performance. Section VII ends this paper with a conclusion and recommendations.

## II. RELATED WORKS

Decision support systems for conflict resolution are recognized as a promising possibility to enhance railway operations. For fixed block signalling systems, this concept has been quite widely researched. In the work of Pellegrini et al. (2015) [13], the authors implement a rescheduling approach that implements retiming and rerouting measures. They introduce a method called RECIFE-MILP, a heuristic algorithm based on a mixed integer linear problem. They minimize delay propagation within short computation times, which makes it suitable for real time applications. Luan et al. (2018) [9] also propose a solution approach for the real time traffic management problem using MILP and MILNP formulations. They compare different methods that provide both a dispatching solution and a train control solution (speed profiles). Their dispatching solution includes reordering, retiming and rerouting. A MILP approach implementing train speed profile options yields the best results within the required computation time of 3 minutes. In several works written by Corman et al (2012, 2014) [23], [3] and D'Ariano et al (2007, 2008) [4], [5], a different method to formulate a railway rescheduling problem is used. They implement an alternative graph approach, where train operations can be modelled using graph theory. This approach will be used in this paper as well, and will be elaborately discussed in the next section. In the works of Corman and D'Ariano, multiple approaches towards train rescheduling have been taken, whilst using this alternative graph method. In Corman et al (2012) [23], trains are rescheduled with different classes of priority, which leads to interesting insights in relation to delay propagation. The work of Corman et al (2014) [3] divides the network into subsystems in order to solve problems both locally and globally. The paper by D'Ariano et al. (2007) [4] adopts the alternative graph method with a variable speed model, where speed coordination is implemented in the iterative solution procedure. In D'Ariano et al. (2008) [5] the authors implement an iterative approach for reordering and rerouting trains using a local search algorithm.

All scientific contributions mentioned above assume a fixed block signalling system. In the work of Mazzarello and Ottaviani (2007) [11], the alternative graph approach is once again implemented. The authors design an elaborate traffic management system for real time railway traffic optimization, that consists of both speed profile generation and conflict detection and resolution. In their work, the authors briefly touch upon the moving block principle and propose an approximation to moving block train driving in the alternative graph formulation. However, in this moving block approximation the block sections still have a fixed length, as they do in fixed block. Mathematical approaches for real time rescheduling under a moving block signalling system are highly underrepresented in current literature. There are some papers that cover moving block signalling for metro- or subway systems. In most of these works, for example Pochet et al. (2016) [32], Carvajal-Carreño et al. (2016) [26], and Mazzanti et al. (2018) [27], the moving block signalling system is referred to as Communication-Based-Train-Control (CBTC). Although these papers can be used as valuable inspiration for moving block railway design, rated speed and operation environment are quite different from railway systems. The contribution of this paper is thus the design of a solution approach to the railway rescheduling problem, that implements retiming and reordering measures, suitable for moving block signallings systems and real time applications.

## III. THEORY

In this section, the theoretical fundamentals of the modelling approach will be explained. First, the state-of-the-art alternative graph formulation is explained. Second, the model extension for moving block suitability is elaborated. Some important notes need to be made before continuation. First, this research employs a microscopic approach on unidirectional tracks and only considers passenger trains. Second, conflict resolution is viewed as the process of rescheduling trains in such a way that a conflict free schedule is obtained, whilst minimizing delays. Note that this means minimizing propagated delays, since initial delays cannot be recovered by rescheduling. Third, this paper only considers initial delays in the form of disturbances, meaning a maximum initial delay of 1200 seconds per train.

### *A. The job shop scheduling problem*

Train scheduling problems, and therefore also rescheduling problems, can be represented as a job shop scheduling problem [4]. In the classical job shop problem, the objective is to schedule a set of jobs to be performed on a set of machines within a minimized amount of time. Additionally, any job with a precedence relation to another job can only start after its predecessor is finished. The processing of a job by a machine is denoted an operation.

In railway problems, trains are defined as jobs and track sections as machines. The route of every train consists of a series of operations, composing the trains journey from its starting point to its destination. When applying the job shop scheduling objective to railway application, it is thus the goal to get all trains from start to end in the shortest possible time frame. The objective can be subjected to several constraints, which will be elaborated on later.

In a fixed block signalling system, this means sections of predefined length processing all trains that need to cross this section within their route. In a moving block signalling system, sections of fixed length are absent, which is why the view on the job shop scheduling problem slightly changes. The jobs are still trains, but a machine can be represented by an infinitesimally small track section, that can process one train at a time. An illustration on this new view on the job shop scheduling problem can be observed in Figure 1. This principle is used throughout the model extension in the next subsection, but before extending an existing model, this existing model must first be clarified.

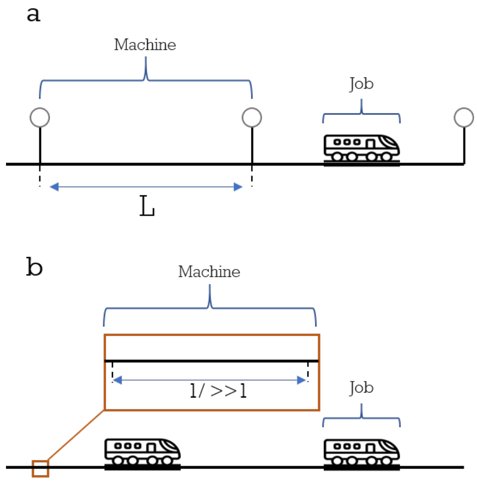


Fig. 1. **a** a visualization of the railway job shop scheduling problem for fixed block signalling systems. **b** a visualization of the railway job shop scheduling problem for moving block signalling systems. Train icon from [33]

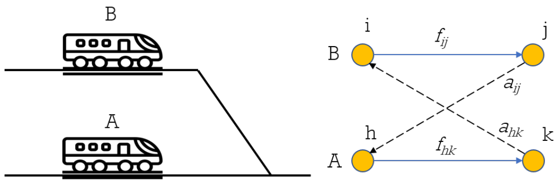


Fig. 2. A conflict between train A and B, where the choice of alternative arc (indicated with dotted lines) will decide the precedence relation between the two trains

### *B. The alternative graph approach*

A job shop scheduling problem can be modelled as an alternative graph, as has been done by D'Ariano and Corman in many of their works [3], [23], [4], [5]. In their approach, the entire network is represented as nodes and two types of arcs: fixed arcs and alternative arcs. The graph is thus formulated as $G = (N, F, A)$. Each node in the graph represents the starting time of an operation at node $i$, for example entering a track section, station or junction. Each fixed arc connecting two nodes in the graph represents the precedence relation between two operations, where the weight of the arc $f_{ij}$ implies the processing time of the operation. An alternative arc can be added between two different train routes, and represents scheduling alternatives in the network. If two trains, say train A and train B, require the same resource simultaneously, a conflict is detected. Alternative arcs $a_{ij}$ and $a_{hk}$ are placed between the end time of this operation for train A and the starting time of this operation for train B and vice versa. Only one of these arcs is chosen, meaning that only one of these trains can start the operation first. In other words, there is now a precedence relation between the ending of the operation for one train and the start of the operation for another. The weight of these arcs is the minimum headway distance between the two trains. This is visualized in Figure 2. The entire railway network, or the part considered, can be modelled within this graph structure. The choice of $a_{ij}$ or $a_{hk}$ composes and influences the solution of the graph. As mentioned earlier, the objective corresponding to this formulation is the minimization of the maximum propagated delay. This is found as follows. For modelling purposes, node $0$ and node $n$ are introduced. The first operation of every train starts at node $0$ and the final node for is equal to sink node $n$, see also Figure 3. It should be noted that node $n$ is a node added for modelling purposes and does not represent an actual operation executed in the network. The time $t_n$ at node $n$ is equal to the maximum propagated delay in the network. This is computed by finding a subset $S$ of $A$, the optimal combination of alternative arc choices, that yields the longest shortest path from 0 to $n$.

### C. Extended model formulation for moving block operations

It can be noted that in the traditional alternative graph approach, the processing time $f_{ij}$ of a fixed arc is the time it takes a train to cross a section of predefined length (running time). The entering of this section is denoted as operation $i$, whilst the entering of the following section is denoted as operation $j$. In a moving block situation, sections of fixed length are absent (and machines thus have an infinitesimal length). To formulate an alternative graph that represents this concept, four aspects are introduced into the theory, which are explained below:

*1) Virtual nodes:* Fixed arc weights are equal to processing times of an operation performed by a machine. If this machine is a track section of infinitesimal length, the processing time becomes equal to the time it takes a train to cross its own length. In other words, the clearing time of the train. This could be viewed as trains continuously passing their own length in order to get from A to B, which makes nodes $i$ and $j$ of fixed arc $f_{ij}$ representative of the front and rear end of a train (+ a safety margin). Therefore, nodes can be described as to be moving along with a train across its route. These nodes are called virtual nodes.

*2) Conditional arcs:* As described, the arc weight $a_{ij}$ of an alternative arc represents the safe headway distance between two trains. In a moving block system, this safe headway distance is to be maintained all throughout the network without trackside signals. Therefore, an arc representing headway between two trains should be placed between two preceding trains throughout the entire network. This arc type is similar to the alternative arc, but has one main difference. Alternative arcs represent a choice in train order. However, the option to change train order is not always a possibility whilst the need to maintain safety is always present. Therefore, conditional arcs are introduced. These arcs have the same weight principle as alternative arcs, only they obey to the order of trains that has already been established. Figure 4 visualizes the principle of these added concepts, where the conditional arcs are denoted $c_{ij}$.

*3) Static and virtual sections:* Besides the representation of train operations as virtual nodes, there are still infrastructural elements on the railway tracks that are static. Most importantly, stations and switches. If all operations are represented only as virtual nodes, this can cause difficulties in passing either of these static components. Therefore, fixed infrastructure is still represented as a fixed section, in this theory called static section, in the same manner as is done in existing alternative graph theory. This means the alternative graph will consist of both static and virtual nodes and arc weights can be either running times or clearing times. The two distinguished node types have the following definition:

**Static node**: Nodes related to physical points from the infrastructure with dimensions, like switches. They represent operations performed by machines of fixed length L. A static section is a block that starts and ends with a static node and the weight of the connecting edge equal to running time.

**Virtual node** Nodes related to driving trains, that represent operations performed by machines with infinitesimal length. A virtual section is a block that starts and ends with a virtual node and the weight of the connecting edge equal to clearing time.

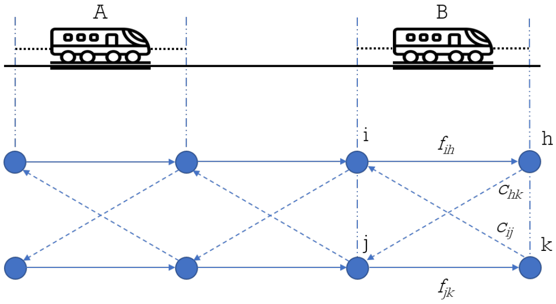


Fig. 4. Trains A and B driving on a track in a moving block signalling system. The graph representing the network consists of virtual nodes indicating the front and rear end of the trains, fixed arcs with weight equal to the clearing time of the train and conditional arcs maintaining a safe headway distance as well as the already established train order

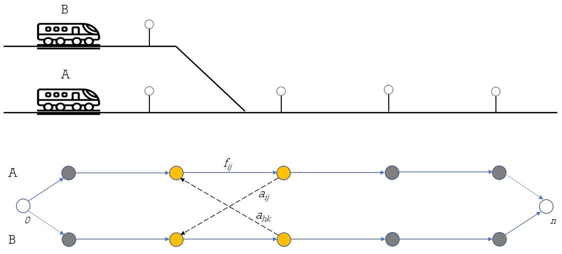


Fig. 3. Alternative graph for two trains on a converging track, explicitly showing node 0 and sink node $n$. The optimal solution consists of the choice of alternative arc ($a_{ij}$ or $a_{hk}$, the yields the s

*4) Generation of virtual nodes and prediction of arrival times at future virtual nodes:* Virtual nodes are related to trains. However, to properly use alternative graph theory, it is necessary to construct a complete graph from beginning to end point of the considered situation. Therefore, the on board speed and position measurements are not only used to create virtual nodes of the current train state, but also to predict the future train state. These states can be updated accordingly. An example alternative graph showing all four aspects of the new modelling method is presented in Figure 5. Three trains in a network are visualized, where train A and B converge to the same track when arriving at a station and train C then crosses their route. There are two static sections, one that represents the station and one that represents the switch (indicated with yellow nodes). The rest of the operations are represented by virtual sections (indicated with blue nodes), related to the trains + a safety margin and the corresponding future node predictions. It should be noted that for this paper, no extensive research has yet been performed on optimizing prediction of future nodes and this principle remains conceptual.

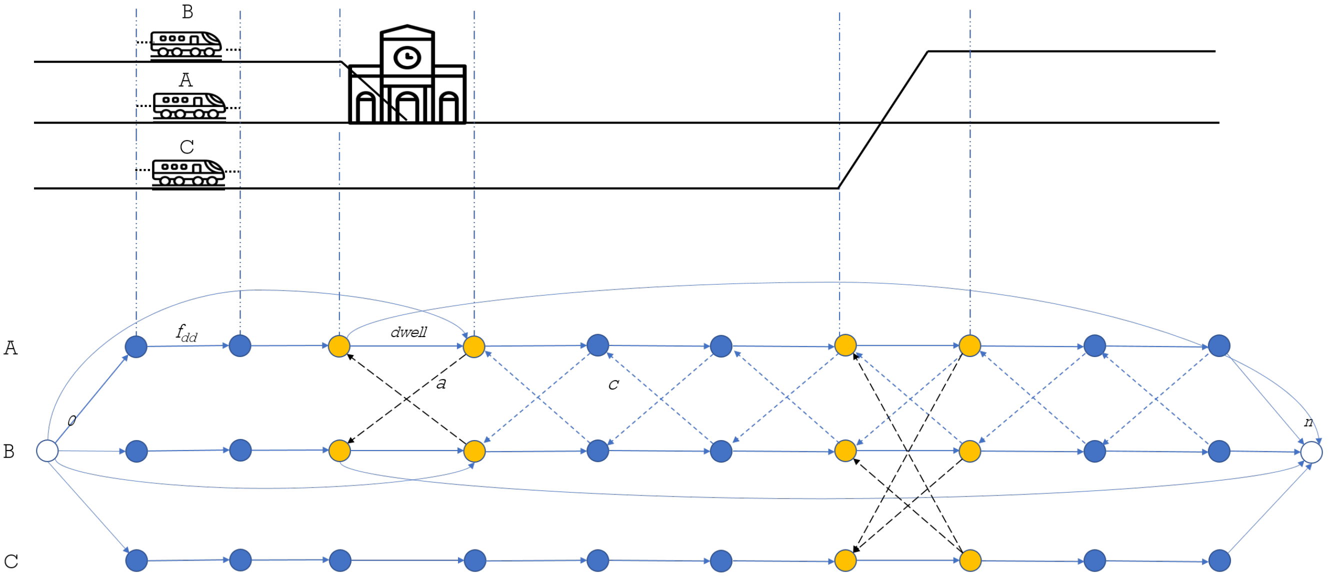


Fig. 5. An example situation of three trains in a network, modelled as an alternative graph extended to moving block suitability. Blue nodes indicate virtual nodes, yellow nodes indicate static nodes. Black dotted arcs indicate alternative arcs, blue dotted arcs indicated conditional arcs

## IV. AN ALTERNATIVE GRAPH-BASED RESCHEDULING MODEL FOR MOVING BLOCK RAIL OPERATIONS

The graph as described in the previous section is now used to formulate an optimization problem. The graph consists of the following parameters:

$G(N, F, A)$ = graph

$N$ = set of nodes
$F$ = set of fixed arcs
$A$ = set of alternative arcs

$i, j$ = indices for nodes
$s$ = index for static node
$v$ = index for virtual node
$\theta$ = index for trains

$N_s \subset N$ = set of static nodes
$N_v \subset N$ = set of virtual nodes
$F_{ss} \subset F$ = set of fixed arcs between two static nodes
$F_{sv} \subset F$ = set of fixed arcs between a static and a virtual node
$F_{vv} \subset F$ = set of fixed arcs between two virtual nodes

$D \subset A$ = set of decisional alternative arcs
$C \subset A$ = set of conditional alternative arcs

$f_{ij}$ = weight of fixed arcs (running time or clearing time)

where:
$f_{i_s j_s}$ = running time
$f_{i_s j_v}$ = running time (to nearest predicting timing point)
$f_{i_v j_v}$ = clearing time

$a_{ij}$ = weight of alternative arcs

where:
$a_{ij}, c_{ij}$ = headway time

$x_\theta$ = scheduled arrival time of train $\theta$
$y_\theta$ = scheduled departure time of train $\theta$
$d_{ij,\theta}$ = scheduled dwell time of train $\theta$

$\tau_i$ = earliest possible start of operation at node $i$
$t_i$ = actual start of operation at node $i$, decision variable

The optimization problem describes the rescheduling process in case of conflict. The decision variable of the optimization problem is $t$, the actual starting time of an operation. This represents the retiming aspect. The alternative graph formulation aims to find the subset of alternative (and therefore conditional) arcs that minimize $t$ at the final station arrival. This represents the reordering aspect. The objective is to reschedule as to minimize the maximum secondary delay, which is equal to the arrival time at sink node $n$, $t_n$, minus the starting time at node 0, $t_0$:

Objective function:

$$min \quad t_n - t_0 \tag{1}$$

Subject to:

$$t_j - t_i \geq f_{ij} \quad \forall \quad (i, j) \in F \tag{2}$$

$$t_j - t_i \geq a_{ij} \lor t_k - t_h \geq a_{hk} \quad \forall \quad ((i, j), (h, k)) \in D \tag{3a}$$

$$\text{if} \quad a_{ij} \quad \text{is selected, then} \qquad t_w - t_v \geq c_{vw} \quad \forall \quad (v, w) \in C_{a_{ij}} \tag{3b}$$

$$\text{if} \quad a_{hk} \quad \text{is selected, then} \qquad t_z - t_u \geq c_{uz} \quad \forall \quad (u, z) \in C_{a_{hk}} \tag{3c}$$

$$t_i \geq q_{i,\theta} \tag{4}$$

$$t_j - t_i \geq d_{ij,\theta} \tag{5}$$

Where the indices $h, i, j, k, u, v, w, z$ are in the set of nodes

$N$ and $\theta$ is in the set of trains $\Theta$.

Equations 1, 2 and 3a in this formulation are based on the mathematical description as by D'Ariano and Corman in many of their works [4], [5], [23]. Equations 3b and 3c show the contribution of this research, explicitly describing the conditional arcs. The contribution of this work is also represented in the different subsets in the set of nodes $N$ and fixed arcs $F$, as described under "indices and sets".

Equation 1 represents the minimization of the maximum arrival time at node $n$, which equals the maximum propagated delay in the network.

Equation 2 represents that the starting time of the operation at the next node (node $j$) is at least the starting time of the previous operation (node $i$) plus the processing time of that operation ($f_{i_s j_s}$, $f_{i_s j_v}$ or $f_{i_v j_v}$)

Equation 3a represents that only one decisional alternative arc can be chosen. This implies that only one of two trains can start an operation first, as was visually represented in Figure 2. For the selected alternative arc, it is stated that the starting time of the operation at node $j$ is at least the starting time of the operation at node $i$ plus the safe time interval (headway) before a next operation can start. The headway is denoted with $a_{ij}$ and $a_{hk}$.

Equations 3b and 3c show the dependence of the conditional alternative arcs. If one decisional arc is selected, the conditional arc that has the same direction must be selected for all conditional arcs dependent on that decisional arc. This implies the same train order must be maintained after this has been established by the decisional alternative arcs. Additionally, these constraints show that the starting time of the operation at node $w$ is at least the starting time of the operation at node $v$ plus the safe time interval (headway) before a next operation can start, similar to constraint 3a. The headway is denoted with $c_{vw}$ and $c_{uz}$. Recall that conditional arcs are placed between trains driving on the same track without any trackside signals and these arc weights thus represent an important safety constraint.

Equation 4 represents that the starting time of the operation at node $i$, when the operation is a station departure, is at least the scheduled departure time at node $i$ for train $\theta$. This constraint could be viewed as already included in constraint 2, but is written separately to show station departure times as an important component of the model.

Equation 5 represents the starting time of the operation at node $j$ (station departure) is at least the starting time of the operation at node $i$ (station arrival) plus the scheduled dwell time of train $\theta$ between nodes $i$ and $j$. This constraint could be viewed as already included in constraint 2, but is written separately to show station dwell times as an important component of the model.

## V. CASE STUDY: THE ROTTERDAM-THE HAGUE DUTCH CORRIDOR

This novel approach is tested and evaluated by the execution of a case study in the Rotterdam-The Hague (Den Haag) Dutch corridor. Since this research only considers one direction of traffic, Rotterdam is viewed as the starting point and The Hague as the final station. An overview of the corridor is presented in Figure 6. The corridor has been split in two for readability purposes, the dashed lines indicates that the bottom and top view are connected in the real railway network. The intermediate stops on the route are:

Rotterdam (Rtd) - Schiedam (Sdm) - Delft Campus (Dtcp) - Delft (Dt) - Den Haag Rijswijk (Gvrw) - Den Haag Moerwijk (Gvmw) - Den Haag (Gv)

There are two types of passenger trains driving from Rotterdam to The Hague; sprinters and intercities. sprinters stop at all intermediate stations, intercities only stop in Rtd, Sdm, Dt and Gv. For the case study, one hourly cycle of Rotterdam train departures is considered. This yields a total of six trains, four sprinters (C5100, C5000, A5100, A5000) and two intercities (C2400, A2400). The scheduled hourly departure times from Rotterdam Central for these trains are given in table I.

The case study has been performed using data from one of ProRail's simulators, FRISO (Flexible Railway Infrastructure Simulation Environment). In the FRISO environment, all six trains have been simulated to be driving conflict free between Rtd and Gvc according to their regular timetables. During the simulation, information is being gathered on the trains' behaviour, logged at different timeable points. This information is used as a basis to construct to alternative graph. The optimization problem is then solved using the commercial solver GUROBI Optimizer with Python.

Conflicts are now added to the Rtd-Gvc network. Different test cases are formulated in the form input delays for one or two of the six trains. The scheduled gaps, or planned headway, between all trains are not identical. Therefore, delays are given as a percentage of the planned headway (ph) in order to stimulate delay propagation (if no measures

TABLE I
DEPARTURE TIMES (SECONDS & MINUTES) FOR ALL SIX TRAINS LEAVING ROTTERDAM CENTRAL STATION IN THE DIRECTION OF THE HAGUE, WITHIN ONE HOURLY CYCLE

| Train Nr | Rtd departure [sec] | Rtd departure [min] |
|---|---|---|
| C5100 | 360 | 6 |
| C5000 | 1260 | 21 |
| C2400 | 1620 | 27 |
| A5100 | 2160 | 36 |
| A5000 | 3060 | 51 |
| A2400 | 3420 | 57 |

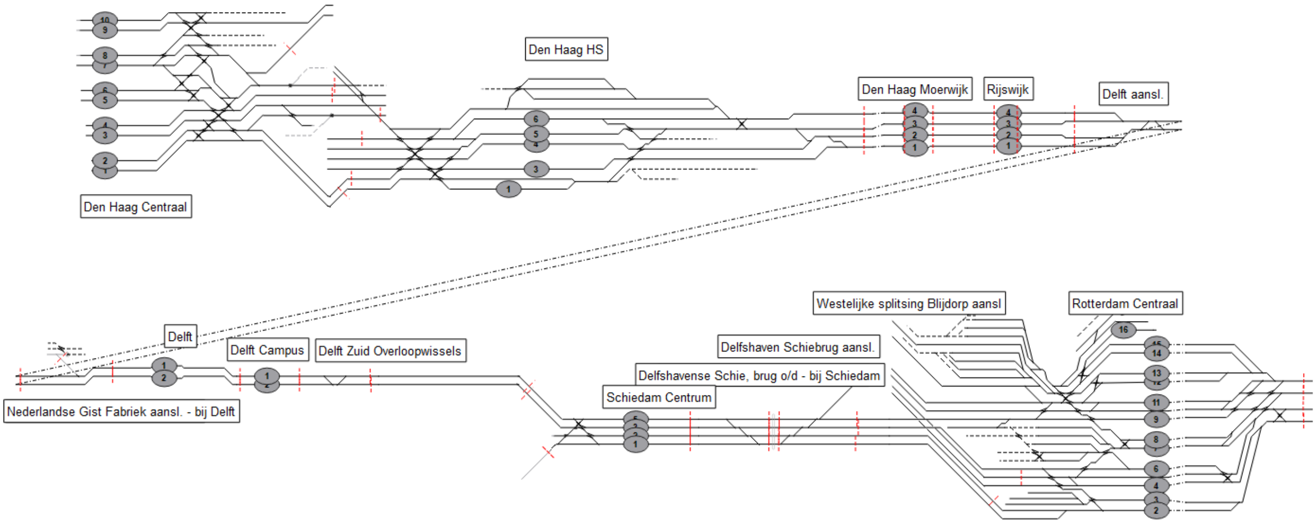


Fig. 6. Microscopic overview of the Rotterdam-The Hague corridor

are taken) and thus trigger rescheduling measures. Four input situations are defined, all having three delay scenarios, see table II. As can be observed from the table, situation 1 and 4 only have one train with initial delay, a sprinter and an intercity respectively. Situations 2 and 3 have two initially delayed trains. Namely two sprinters in the former, one sprinter and one intercity in the latter.

TABLE II
INITIAL DELAYS (IN SECONDS) TO BE IMPOSED ON THE RTD-GVC CORRIDOR AS A PERCENTAGE OF THE PLANNED HEADWAY (PH)

| Test case | Delay scenario | Input delay value (sec) | | | |
|---|---|---|---|---|---|
| | | C5000 | C5100 | C2400 | Total |
| 1. | ph -30% | 630 | | | 630 |
| | ph | 900 | | | 900 |
| | ph + 30% | 1170 | | | 1170 |
| 2. | ph -30% | 630 | 280 | | 910 |
| | ph | 900 | 400 | | 1300 |
| | ph +30% | 1170 | 520 | | 1690 |
| 3. | ph -30% | 630 | | 378 | 1008 |
| | ph | 900 | | 540 | 1440 |
| | ph +30% | 1170 | | 702 | 1872 |
| 4. | ph -30% | | | 378 | 378 |
| | ph | | | 540 | 540 |
| | ph +30% | | | 702 | 702 |

## VI. RESULTS

The performance of the model within the case study is evaluated on three KPI's:

1) Number of trains affected
2) Reduction of propagated delay (compared to not applying a rescheduling algorithm, in %)
3) Computation time (in seconds)

The first KPI is formulated in relation to passenger satisfaction. Trains always drive according to a fixed schedule, and the less trains affected the more trains still drive regularly, which is pleasant for train passengers. It can be interesting to research the trade off between regularity and maximum delay reduction, but this has not been done in this paper. The second KPI is a very valuable KPI in conflict resolution system development. After all, delay reduction is the objective of the mathematical model and from a dispatching point of view, minimizing the delay is the primary goal. The final KPI is an important KPI when designing solution approaches for real time applications. In real time traffic management, feasibility could be favoured over optimality and solutions need to be generated fast. The quicker the algorithm, the sooner the train driver can receive directions and can take action.

Regarding the number of affected trains, table III shows the number of retimed and reorderd trains for every test case and delay scenario. It should be noted that trains cannot be reorderd without also being retimed. Nonetheless, the reorderings and retimings are counted seperately (so if the solution contains one reordered train and one retimed train, this has to be the same train). In table III it can indeed be observed that the number of retimed trains is in all cases at least equal to the number of reordered trains. It is in line with expectations that the number of retimings is often larger, since retiming a train is a less severe measure and can already (partially) reduce propagated delay without reordering being necessary. Additionally, it is evident that a reordering measure is always imposed on at least two trains, which is the minimum number of reordered trains if larger than zero. However, in this case study reordering is never performed for more than two trains. It could be that this might be reformed in more complex situations that involve more crossings and track layout variations, or if initial timetable conditions are different.

In Figures 7-10, results in relation to delay reduction are displayed. The x-axis indicates the different delays

TABLE III
INITIAL DELAYS (IN SECONDS) TO BE IMPOSED ON THE RTD-GVC CORRIDOR

| Test case | Delay scenario | Retimed | Reordered |
|---|---|---|---|
| 1. | ph-30% | 4 | 2 |
| | ph | 3 | 2 |
| | ph+30% | 3 | 2 |
| 2. | ph-30% | 4 | 2 |
| | ph | 3 | 2 |
| | ph+30% | 3 | 2 |
| 3. | ph-30% | 3 | 0 |
| | ph | 2 | 0 |
| | ph+30% | 0* | 0 |
| 4. | ph-30% | 2 | 0 |
| | ph | 1 | 0 |
| | ph+30% | 0* | 0 |

scenarios imposed on the model. The y-axis indicates the percent reduction of propagated delay. This reduction is measured in comparison to not taking any rescheduling measures and only relying on timetable buffer times. The figures thus show the sensitivity of delay reduction to the different input delay scenarios. Figure 7 compares test case 1 and 2, Figure 8 compares test case 1 and 4, Figure 9 compares test case 2 and 3, and Figure 10 compares all test cases.

Overall it can be observed that delay is indeed reduced for in all test cases, varying from 5% to up to 70% of delay reduction. The two situations with very high reduction rates include delayed intercities. The large reductions could be explained by the fact that the initial delay is not very high, especially for the intercity. Since intercities have less stops it could be that their initial delays have less consequences, and this conflict can be mitigated strongly with just a retiming measure. For the majority of the situations, delay propagation is reduced 10-50%.

It can be found that for test cases 1 and 2, the delay reduction is generally higher than for test cases 3 and 4. Looking at table III, it can be observed that test cases 1 and 2 are the situations where reordering measures are applied, which is not the case for the other two. This could explain a larger delay reduction, since reordering is a more severe measure than retiming, so effects can be more significant once applied.

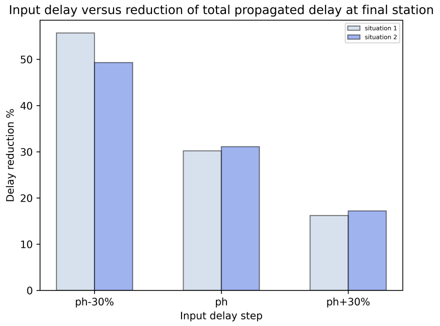


Fig. 7. Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 2, displayed per delay scenario

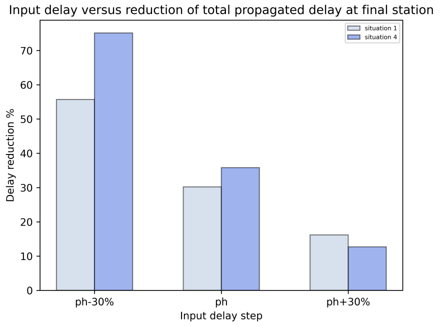


Fig. 8. Sensitivity of percentage delay reduction to initial delay, comparing test cases 1 and 4, displayed per delay scenario

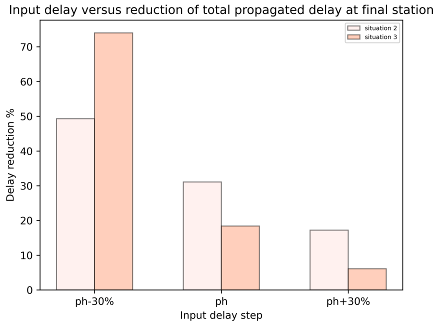


Fig. 9. Sensitivity of percentage delay reduction to initial delay, comparing test cases 2 and 3, displayed per delay scenario

Besides the fact that rescheduling measures are applied and delays are being reduced accordingly, there is another trend to be noticed from these results. In all four Figures 7-10 a decreasing trend is showing, indicating that delay reduction is less for delay scenarios with higher initial delays. This could imply that the delay reduction reaches a limit at some point and that the model within the current scope is not sufficient anymore to reduce propagated delays. It should be researched whether this is the case and if so, if there are solutions. One approach could be to also implement rerouting into the model, which enables more feasible solutions and therefore possibly better results.

The sensitivity of computation time to different test cases is shown in Figure 11. First of all, it should be noted that all computation times are less then 10 seconds, which is considerably fast and suitable for real time implementations. However, in addition it should also be observed that the computation times tend to increase for delay scenarios with higher initial delays. This can be caused by increased problem complexity and more information to process. It can be observed that for test cases 2 and 3, which contain two initially delayed trains instead of one, the computation time is generally higher. This shows that computational complexity can thus vary, even within the same network size. Computation times becoming too high due to complex situations and large amounts of data is not an

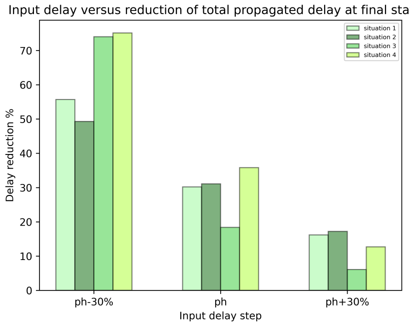


Fig. 10. Sensitivity of percentage delay reduction to initial delay, comparing all test cases, displayed per delay scenario

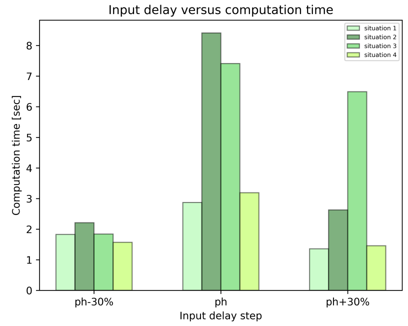


Fig. 11. Sensitivity of computation time to initial delay, comparing all test cases, displayed per delay scenario

uncommon problem in real time applications. The increasing trend shown in Figure 11 might indicate that this could cause problems for applications in larger and complex networks. In that case, it can be valuable to investigate tailored heuristics or non-centralized options for optimization.

## VII. CONCLUSION

A new approach has been designed to perform conflict resolution under moving block signalling, that has been based on a valued state-of-the-art model and has been evaluated on performance during a case study. From the case study it can be concluded that for the majority of test cases, retiming and reordering measures can yield delay reductions of 10-50% within 10 seconds of computation time. The measures affect 2-4 out of 6 trains, mostly with retiming measures. The effects are strongest when reordering measures are also applied, but this is of course only done when this is beneficial in terms of delay. Subsequently, it can also be concluded that delay reduction decreases and computation time increases as the initial delay or situational complexity to the model increase. This could eventually cause the delay reduction to reach a limit and become unrewarding with respect to the amount of computation time necessary the complete the rescheduling. However, since this model is designed for disturbances, a maximum initial delay of 20 minutes, the model is considered suitable for its cause.

Further research is necessary in order to make this new model useful in practice, in order to be applicable in an actual ERTMS level 3 system. Three important recommendations for additional research are given. First, rerouting should be implemented into the model, to extend the solution space and possibly generate more optimal results. Second, research should be performed on the generation of virtual nodes, predicting virtual arrival times, and how to implement this into a dynamically functioning decision support system. Third, this paper only considered passenger trains, but also adding freight trains to the network could show new insights in terms of model performance.

## ACKNOWLEDGMENT

I would like to thank Maura Mazzarello for her valuable input to this research.

## REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] Corman, F., D'Ariano, A., Hansen, I. & Pacciarelli, D. Optimal multi-class rescheduling of railway traffic. *Journal Of Rail Transport Planning  Management*. **1**, 14-24 (2011,11), https://doi.org/10.1016/j.jrtpm.2011.06.001

[3] Corman, F., D'Ariano, A., Pacciarelli, D. & Pranzo, M. Dispatching and coordination in multi-area railway traffic management. *Computers  Operations Research*. **44** pp. 146-160 (2014,4), https://doi.org/10.1016/j.cor.2013.11.011

[4] D'Ariano, A., Pranzo, M. & Hansen, I. Conflict Resolution and Train Speed Coordination for Solving Real-Time Timetable Perturbations. *IEEE Transactions On Intelligent Transportation Systems*. **8**, 208-222 (2007,6), https://doi.org/10.1109/tits.2006.888605

[5] D'Ariano, A., Corman, F., Pacciarelli, D. & Pranzo, M. Reordering and Local Rerouting Strategies to Manage Train Traffic in Real Time. *Transportation Science*. **42**, 405-419 (2008,11), https://doi.org/10.1287/trsc.1080.0247

[6] D'Ariano, A., Samà, M., D'Ariano, P. & Pacciarelli, D. Evaluating the Applicability of Advanced Techniques for Practical Real-time Train Scheduling. *Transportation Research Procedia*. **3** pp. 279-288 (2014), https://doi.org/10.1016/j.trpro.2014.10.007

[7] Giannettoni, M. & Savio, S. The European project COMBINE 2 to improve knowledge on future rail traffic management systems. *WIT Transactions On The Built Environment*. **74** (2004)

[8] Giuliari, M., Pellegrini, F. & Savio, S. Moving block and traffic management in railway applications: The EU project COMBINE. *WIT Transactions On The Built Environment*. **50** (2000)

[9] Luan, X., Wang, Y., Schutter, B., Meng, L., Lodewijks, G. & Corman, F. Integration of real-time traffic management and train control for rail networks - Part 1: Optimization problems and solution approaches. *Transportation Research Part B: Methodological*. **115** pp. 41-71 (2018,9), https://doi.org/10.1016/j.trb.2018.06.006

[10] Luan, X., Schutter, B., Meng, L. & Corman, F. Decomposition and distributed optimization of real-time traffic management for large-scale railway networks. *Transportation Research Part B: Methodological*. **141** pp. 72-97 (2020,11), https://doi.org/10.1016/j.trb.2020.09.004

[11] Mazzarello, M. & Ottaviani, E. A traffic management system for real-time traffic optimisation in railways. *Transportation Research Part B: Methodological*. **41**, 246-274 (2007,2), https://doi.org/10.1016/j.trb.2006.02.005

[12] Narayanaswami, S. & Rangaraj, N. A MAS architecture for dynamic, realtime rescheduling and learning applied to railway transportation. *Expert Systems With Applications*. **42**, 2638-2656 (2015,4), https://doi.org/10.1016/j.eswa.2014.11.013

[13] Pellegrini, P., Marliere, G., Pesenti, R. & Rodriguez, J. RECIFE-MILP: An Effective MILP-Based Heuristic for the Real-Time Railway Traffic Management Problem. *IEEE Transactions On Intelligent Transportation Systems*. **16**, 2609-2619 (2015,10), https://doi.org/10.1109/tits.2015.2414294

[14] Xu, P., Corman, F., Peng, Q. & Luan, X. A train rescheduling model integrating speed management during disruptions of high-speed traffic under a quasi-moving block system. *Transportation Research Part B: Methodological*. **104** pp. 638-666 (2017,10), https://doi.org/10.1016/j.trb.2017.05.008

[15] Xu, P., Zhang, D., Guo, J., Liu, D. & Peng, H. Integrated Train Rescheduling and Rerouting during Multidisturbances under a Quasi-Moving Block System. *Journal Of Advanced Transportation*. **2021** pp. 1-15 (2021,4), https://doi.org/10.1155/2021/6652531

[16] Corman, F., D'Ariano, A., Pacciarelli, D. & Pranzo, M. Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport*. **2**, 219-247 (2010,8), https://doi.org/10.1007/s12469-010-0032-7

[17] Keita, K., Pellegrini, P. & Rodriguez, J. A three-step Benders decomposition for the real-time Railway Traffic Management Problem. *Journal Of Rail Transport Planning Management*. **13** pp. 100170 (2020,5), https://doi.org/10.1016/j.jrtpm.2019.100170

[18] Corman, F., D'Ariano, A., Marra, A., Pacciarelli, D. & Samà, M. Integrating train scheduling and delay management in real-time railway traffic control. *Transportation Research Part E: Logistics And Transportation Review*. **105** pp. 213-239 (2017,9), https://doi.org/10.1016/j.tre.2016.04.007

[19] Quaglietta, E., Pellegrini, P., Goverde, R., Albrecht, T., Jaekel, B., Marlière, G., Rodriguez, J., Dollevoet, T., Ambrogio, B., Carcasole, D., Giaroli, M. & Nicholson, G. The ON-TIME real-time railway traffic management framework: A proof-of-concept using a scalable standardised data communication architecture. *Transportation Research Part C: Emerging Technologies*. **63** pp. 23-50 (2016,2), https://doi.org/10.1016/j.trc.2015.11.014

[20] Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L. & Wagenaar, J. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*. **63** pp. 15-37 (2014,5), https://doi.org/10.1016/j.trb.2014.01.009

[21] Carlier, J. & Pinson, E. An Algorithm for Solving the Job-Shop Problem. *Management Science*. **35**, 164-176 (1989,2), https://doi.org/10.1287/mnsc.35.2.164

[22] Oliveira, E. & Smith, B. A job-shop scheduling model for the single-track railway scheduling problem. *RESEARCH REPORT SERIES-UNIVERSITY OF LEEDS SCHOOL OF COMPUTER STUDIES LU SCS RR*. (2000)

[23] Corman, F., D'Ariano, A., Pacciarelli, D. & Pranzo, M. Optimal inter-area coordination of train rescheduling decisions. *Transportation Research Part E: Logistics And Transportation Review*. **48**, 71-88 (2012,1), https://doi.org/10.1016/j.tre.2011.05.002

[24] Quaglietta, E., Wang, M. & Goverde, R. A multi-state train-following model for the analysis of virtual coupling railway operations. *Journal Of Rail Transport Planning Management*. **15** pp. 100195 (2020,9), https://doi.org/10.1016/j.jrtpm.2020.100195

[25] Corman, F., D'Ariano, A., Pacciarelli, D. & Pranzo, M. Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies*. **17**, 607-616 (2009,12), https://doi.org/10.1016/j.trc.2009.04.001

[26] Carvajal-Carreño, W., Cucala, A. & Fernández-Cardador, A. Fuzzy train tracking algorithm for the energy efficient operation of CBTC equipped metro lines. *Engineering Applications Of Artificial Intelligence*. **53** pp. 19-31 (2016,8), https://doi.org/10.1016/j.engappai.2016.03.011

[27] Mazzanti, F. & Ferrari, A. Ten Diverse Formal Models for a CBTC Automatic Train Supervision System. *Electronic Proceedings In Theoretical Computer Science*. **268** pp. 104-149 (2018,3), https://doi.org/10.4204/eptcs.268.4

[28] Commission, E. The journey begins – 2021 is the European Year of Rail!. , https://ec.europa.eu/commission/presscorner/detail/en/IP_20_2528, retrieved at 30-06-2021

[29] Agency, I. The Future of Rail. , https://www.iea.org/reports/the-future-of-rail, retrieved at 06-07-2021

[30] Eom, S., Lee, S., Kim, E. & Somarajan, C. A survey of decision support system applications (1988–1994). *Journal Of The Operational Research Society*. **49**, 109-120 (1998,2), https://doi.org/10.1057/palgrave.jors.2600507

[31] Zafar, N., Khan, S. & Araki, K. Towards the safety properties of moving block railway interlocking system. *Int. J. Innovative Comput., Info Control*. **8**, 5677-5690 (2012)

[32] Pochet, J., Baro, S. & Sandou, G. Supervision and rescheduling of a mixed CBTC traffic on a suburban railway line. *2016 IEEE International Conference On Intelligent Rail Transportation (ICIRT)*. (2016,8), https://doi.org/10.1109/icirt.2016.7588547

[33] Flaticon.com undefined. ,

# Bibliography

[1] International Energy Agency. *The Future of Rail*. retrieved at 06-07-2021. URL: `https://www.iea.org/reports/the-future-of-rail`.

[2] Jacek Błażewicz, Wolfgang Domschke, and Erwin Pesch. “The job shop scheduling problem: Conventional and new solution techniques”. In: *European Journal of Operational Research* 93.1 (Aug. 1996), pp. 1–33. DOI: `10.1016/0377-2217(95)00362-2`. URL: `https://doi.org/10.1016/0377-2217(95)00362-2`.

[3] Valentina Cacchiani et al. “An overview of recovery models and algorithms for real-time railway rescheduling”. In: *Transportation Research Part B: Methodological* 63 (May 2014), pp. 15–37. DOI: `10.1016/j.trb.2014.01.009`. URL: `https://doi.org/10.1016/j.trb.2014.01.009`.

[4] J. Carlier and E. Pinson. “An Algorithm for Solving the Job-Shop Problem”. In: *Management Science* 35.2 (Feb. 1989), pp. 164–176. DOI: `10.1287/mnsc.35.2.164`. URL: `https://doi.org/10.1287/mnsc.35.2.164`.

[5] William Carvajal-Carreño, Asunción P. Cucala, and Antonio Fernández-Cardador. “Fuzzy train tracking algorithm for the energy efficient operation of CBTC equipped metro lines”. In: *Engineering Applications of Artificial Intelligence* 53 (Aug. 2016), pp. 19–31. DOI: `10.1016/j.engappai.2016.03.011`. URL: `https://doi.org/10.1016/j.engappai.2016.03.011`.

[6] Graziana Cavone et al. “An MPC-Based Rescheduling Algorithm for Disruptions and Disturbances in Large-Scale Railway Networks”. In: *IEEE Transactions on Automation Science and Engineering* (2020), pp. 1–14. DOI: `10.1109/tase.2020.3040940`. URL: `https://doi.org/10.1109/tase.2020.3040940`.

[7] Magali Champion. “Contribution to the modeling and inference of gene regulatory networks”. In: (Dec. 2014).

[8] Chongshuang Chen, Twan Dollevoet, and Jun Zhao. “One-block train formation in large-scale railway networks: An exact model and a tree-based decomposition algorithm”. In: *Transportation Research Part B: Methodological* 118 (Dec. 2018), pp. 1–30. DOI: `10.1016/j.trb.2018.10.003`. URL: `https://doi.org/10.1016/j.trb.2018.10.003`.

[9] European Commission. *The journey begins – 2021 is the European Year of Rail!* retrieved at 30-06-2021. URL: `https://ec.europa.eu/commission/presscorner/detail/en/IP_20_2528`.

[10] F. Corman et al. “Dispatching and coordination in multi-area railway traffic management”. In: *Computers & Operations Research* 44 (Apr. 2014), pp. 146–160. DOI: `10.1016/j.cor.2013.11.011`. URL: `https://doi.org/10.1016/j.cor.2013.11.011`.

[11] F. Corman et al. “Optimal inter-area coordination of train rescheduling decisions”. In: *Transportation Research Part E: Logistics and Transportation Review* 48.1 (Jan. 2012), pp. 71–88. DOI: `10.1016/j.tre.2011.05.002`. URL: `https://doi.org/10.1016/j.tre.2011.05.002`.

[12] Francesco Corman et al. “Centralized versus distributed systems to reschedule trains in two dispatching areas”. In: *Public Transport* 2.3 (Aug. 2010), pp. 219–247. DOI: `10.1007/s12469-010-0032-7`. URL: `https://doi.org/10.1007/s12469-010-0032-7`.

[13] Francesco Corman et al. “Evaluation of green wave policy in real-time railway traffic management”. In: *Transportation Research Part C: Emerging Technologies* 17.6 (Dec. 2009), pp. 607–616. DOI: `10.1016/j.trc.2009.04.001`. URL: `https://doi.org/10.1016/j.trc.2009.04.001`.

[14] Francesco Corman et al. “Integrating train scheduling and delay management in real-time railway traffic control”. In: *Transportation Research Part E: Logistics and Transportation Review* 105 (Sept. 2017), pp. 213–239. DOI: `10.1016/j.tre.2016.04.007`. URL: `https://doi.org/10.1016/j.tre.2016.04.007`.

[15] Francesco Corman et al. “Optimal multi-class rescheduling of railway traffic”. In: *Journal of Rail Transport Planning & Management* 1.1 (Nov. 2011), pp. 14–24. DOI: `10.1016/j.jrtpm.2011.06.001`. URL: `https://doi.org/10.1016/j.jrtpm.2011.06.001`.

[16] Andrea D’Ariano, Marco Pranzo, and Ingo A. Hansen. “Conflict Resolution and Train Speed Coordination for Solving Real-Time Timetable Perturbations”. In: *IEEE Transactions on Intelligent Transportation Systems* 8.2 (June 2007), pp. 208–222. DOI: `10.1109/tits.2006.888605`. URL: `https://doi.org/10.1109/tits.2006.888605`.

[17] Andrea D’Ariano et al. “Evaluating the Applicability of Advanced Techniques for Practical Real-time Train Scheduling”. In: *Transportation Research Procedia* 3 (2014), pp. 279–288. DOI: `10.1016/j.trpro.2014.10.007`. URL: `https://doi.org/10.1016/j.trpro.2014.10.007`.

[18] Andrea D’Ariano et al. “Reordering and Local Rerouting Strategies to Manage Train Traffic in Real Time”. In: *Transportation Science* 42.4 (Nov. 2008), pp. 405–419. DOI: `10.1287/trsc.1080.0247`. URL: `https://doi.org/10.1287/trsc.1080.0247`.

[19] “Distributed energy management of integrated electricity-thermal systems for high-speed railway traction grids and stations”. In: *CSEE Journal of Power and Energy Systems* (2020). DOI: `10.17775/cseejpes.2020.02000`. URL: `https://doi.org/10.17775/cseejpes.2020.02000`.

[20] domotica. *SPOORSECTOR STAAT KLAAR OM MOGELIJKHEDEN LANDELIJK IOT-NETWERK TE BENUTTEN*. retrieved at 29-06-2021 dutch source. 2016. URL: `https://domotica.nl/2016/10/26/spoorsector-iot-netwerk-mogelijkheden/`.

[21] S B Eom et al. “A survey of decision support system applications (1988–1994)”. In: *Journal of the Operational Research Society* 49.2 (Feb. 1998), pp. 109–120. DOI: `10.1057/palgrave.jors.2600507`. URL: `https://doi.org/10.1057/palgrave.jors.2600507`.

[22] B. Erol et al. “TTPlib 2008 – a library for train timetabling problems”. In: *Computers in Railways XI*. WIT Press, Aug. 2008. DOI: `10.2495/cr080591`. URL: `https://doi.org/10.2495/cr080591`.

[23] Muhammad Faris et al. “Distributed Optimization for Railway Track Maintenance Operations Planning”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Nov. 2018. DOI: `10.1109/itsc.2018.8569335`. URL: `https://doi.org/10.1109/itsc.2018.8569335`.

[24] M□ Giannettoni and S Savio. “The European project COMBINE 2 to improve knowledge on future rail traffic management systems”. In: *WIT Transactions on The Built Environment* 74 (2004).

[25] M Giuliari, F Pellegrini, and S Savio. “Moving block and traffic management in railway applications: The EU project COMBINE”. In: *WIT Transactions on The Built Environment* 50 (2000).

[26] NSW government. *Why is rail travel a better choice for the environment?* retrieved at 30-06-2021. URL: `https://www.transport.nsw.gov.au/projects/environment-and-safety/sydney-trains-environment-and-sustainability/why-rail-travel-a`.

[27] Qing Gu et al. “Peak Power Demand and Energy Consumption Reduction Strategies for Trains under Moving Block Signalling System”. In: *Mathematical Problems in Engineering* 2013 (2013), pp. 1–11. DOI: `10.1155/2013/940936`. URL: `https://doi.org/10.1155/2013/940936`.

[28] Gurobi. *Gurobi optimizer*. retrieved at 27-11-2021. URL: `https://www.gurobi.com/products/gurobi-optimizer/`.

[29] Gurobi. *Method*. retrieved at 15-01-2022. URL: `https://www.gurobi.com/documentation/9.5/refman/method.html`.

[30] Hamid Hassanabadi, Bijan Moaveni, and Mohammad Karimi. “A comprehensive distributed architecture for railway traffic control using multi-agent systems”. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 229.2 (Sept. 2013), pp. 109–124. DOI: `10.1177/0954409713503458`. URL: `https://doi.org/10.1177/0954409713503458`.

[31] R.J. Hill and L.J. Bond. “Modelling moving-block railway signalling systems using discrete-event simulation”. In: *Proceedings of the 1995 IEEE/ASME Joint Railroad Conference*. IEEE. DOI: `10.1109/rrcon.1995.395162`. URL: `https://doi.org/10.1109/rrcon.1995.395162`.

[32] Kaba Keita, Paola Pellegrini, and Joaquin Rodriguez. “A three-step Benders decomposition for the real-time Railway Traffic Management Problem”. In: *Journal of Rail Transport Planning & Management* 13 (May 2020), p. 100170. DOI: `10.1016/j.jrtpm.2019.100170`. URL: `https://doi.org/10.1016/j.jrtpm.2019.100170`.

[33] Bart Kersbergen, Ton van den Boom, and Bart De Schutter. “Distributed model predictive control for railway traffic management”. In: *Transportation Research Part C: Emerging Technologies* 68 (July 2016), pp. 462–489. DOI: `10.1016/j.trc.2016.05.006`. URL: `https://doi.org/10.1016/j.trc.2016.05.006`.

[34] Bart Kersbergen, Ton van den Boom, and Bart De Schutter. “Distributed model predictive control for rescheduling of railway traffic”. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Oct. 2014. DOI: `10.1109/itsc.2014.6958127`. URL: `https://doi.org/10.1109/itsc.2014.6958127`.

[35] Sara Khayyam et al. “Railway Energy Management System: Centralized–Decentralized Automation Architecture”. In: *IEEE Transactions on Smart Grid* 7.2 (Mar. 2016), pp. 1164–1175. DOI: `10.1109/tsg.2015.2421644`. URL: `https://doi.org/10.1109/tsg.2015.2421644`.

[36] Masakazu Kojima, Shinji Mizuno, and Akiko Yoshise. “A little theorem of the big M in interior point algorithms”. In: *Mathematical Programming* 59.1-3 (Mar. 1993), pp. 361–375. DOI: `10.1007/bf01581253`. URL: `https://doi.org/10.1007/bf01581253`.

[37] Jan A Koning. “Comparing the performance of ERTMS level 2 fixed block and ERTMS level 3 moving block signalling systems using simulation techniques”. In: *WIT Transactions on The Built Environment* 61 (2002).

[38] Xiaojie Luan et al. “Decomposition and distributed optimization of real-time traffic management for large-scale railway networks”. In: *Transportation Research Part B: Methodological* 141 (Nov. 2020), pp. 72–97. DOI: `10.1016/j.trb.2020.09.004`. URL: `https://doi.org/10.1016/j.trb.2020.09.004`.

[39] Xiaojie Luan et al. “Integration of real-time traffic management and train control for rail networks - Part 1: Optimization problems and solution approaches”. In: *Transportation Research Part B: Methodological* 115 (Sept. 2018), pp. 41–71. DOI: `10.1016/j.trb.2018.06.006`. URL: `https://doi.org/10.1016/j.trb.2018.06.006`.

[40] Franco Mazzanti and Alessio Ferrari. “Ten Diverse Formal Models for a CBTC Automatic Train Supervision System”. In: *Electronic Proceedings in Theoretical Computer Science* 268 (Mar. 2018), pp. 104–149. DOI: `10.4204/eptcs.268.4`. URL: `https://doi.org/10.4204/eptcs.268.4`.

[41] Maura Mazzarello and Ennio Ottaviani. “A traffic management system for real-time traffic optimisation in railways”. In: *Transportation Research Part B: Methodological* 41.2 (Feb. 2007), pp. 246–274. DOI: `10.1016/j.trb.2006.02.005`. URL: `https://doi.org/10.1016/j.trb.2006.02.005`.

[42] MobiRail. *Wat is GSM-R?* MobiRail, dutch source. URL: `http://www.mobirail.nl/Pages/wat_is_gsm-r.html`.

[43] Sundaravalli Narayanaswami and Narayan Rangaraj. “A MAS architecture for dynamic, realtime rescheduling and learning applied to railway transportation”. In: *Expert Systems with Applications* 42.5 (Apr. 2015), pp. 2638–2656. DOI: `10.1016/j.eswa.2014.11.013`. URL: `https://doi.org/10.1016/j.eswa.2014.11.013`.

[44] Elias Oliveira and Barbara M Smith. “A job-shop scheduling model for the single-track railway scheduling problem”. In: *RESEARCH REPORT SERIES-UNIVERSITY OF LEEDS SCHOOL OF COMPUTER STUDIES LU SCS RR* 21 (2000).

[45] Paola Pellegrini et al. “RECIFE-MILP: An Effective MILP-Based Heuristic for the Real-Time Railway Traffic Management Problem”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.5 (Oct. 2015), pp. 2609–2619. DOI: `10.1109/tits.2015.2414294`. URL: `https://doi.org/10.1109/tits.2015.2414294`.

[46] Juliette Pochet, Sylvain Baro, and Guillaume Sandou. “Supervision and rescheduling of a mixed CBTC traffic on a suburban railway line”. In: *2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT)*. IEEE, Aug. 2016. DOI: `10.1109/icirt.2016.7588547`. URL: `https://doi.org/10.1109/icirt.2016.7588547`.

[47] ProRail. *Coördinatie van het treinverkeer*. retrieved at 29-06-2021 dutch source. URL: `https://www.prorail.nl/over-ons/wat-doet-prorail/coordinatie-treinverkeer`.

[48] Egidio Quaglietta. “Analysis of Platooning Train Operations under V2V communication-based signalling:: fundamental modelling and capacity impacts of Virtual Coupling”. In: *Proceedings of the 98th Transportation Research Board Annual Meeting*. Transportation Research Board (TRB). 2019.

[49] Egidio Quaglietta. *Lecture 8.1 Real-time rescheduling*. Delft University of Technology. 2020.

[50] Egidio Quaglietta, Meng Wang, and Rob M.P. Goverde. “A multi-state train-following model for the analysis of virtual coupling railway operations”. In: *Journal of Rail Transport Planning & Management* 15 (Sept. 2020), p. 100195. DOI: `10.1016/j.jrtpm.2020.100195`. URL: `https://doi.org/10.1016/j.jrtpm.2020.100195`.

[51] EU rail. *Why EUrail is your greenest choice*. retrieved at 30-06-2021. URL: `https://www.eurail.com/en/get-inspired/everything-you-need-know-about-eurail/advantages-train-travel/why-eurail-your-greenest-choice`.

[52] Rijnmond.nl. “Bijna tien procent van treinen op Rotterdam Centraal rijdt niet op tijd”. In: (Dec. 2019). Photo by Rick Huijzer, retrieved at 20-12-2021. URL: `https://www.rijnmond.nl/nieuws/189424/Bijna-tien-procent-van-treinen-op-Rotterdam-Centraal-rijdt-niet-op-tijd`.

[53] C. Roberts et al. “Distributed quantitative and qualitative fault diagnosis: railway junction case study”. In: *Control Engineering Practice* 10.4 (Apr. 2002), pp. 419–429. DOI: `10.1016/s0967-0661(01)00159-9`. URL: `https://doi.org/10.1016/s0967-0661(01)00159-9`.

[54] Ian Tiseo. *Transportation-related GHG emission shares in the EU-27 2019, by mode*. retrieved at 03-01-2022. Nov. 2021. URL: `https://www.statista.com/statistics/1237597/transport-greenhouse-gas-emissions-by-mode-eu/`.

[55] treinreiziger. *Spoorkaart 2021: deze vier kaarten zijn er*. retrieved at 29-06-2021 dutch source. Jan. 2021. URL: `https://www.treinreiziger.nl/spoorkaart-2019-deze-vier-versies-zijn-er/`.

[56] Yihui Wang et al. “Optimal trajectory planning for trains under fixed and moving signaling systems using mixed integer linear programming”. In: *Control Engineering Practice* 22 (Jan. 2014), pp. 44–56. DOI: `10.1016/j.conengprac.2013.09.011`. URL: `https://doi.org/10.1016/j.conengprac.2013.09.011`.

[57] Margaret Wright. “The interior-point revolution in optimization: History, recent developments, and lasting consequences”. In: *Bulletin of the American Mathematical Society* 42.1 (Sept. 2004), pp. 39–56. DOI: `10.1090/s0273-0979-04-01040-7`. URL: `https://doi.org/10.1090/s0273-0979-04-01040-7`.

[58] Peijuan Xu et al. “A train rescheduling model integrating speed management during disruptions of high-speed traffic under a quasi-moving block system”. In: *Transportation Research Part B: Methodological* 104 (Oct. 2017), pp. 638–666. DOI: `10.1016/j.trb.2017.05.008`. URL: `https://doi.org/10.1016/j.trb.2017.05.008`.

[59] Peijuan Xu et al. “Integrated Train Rescheduling and Rerouting during Multidisturbances under a Quasi-Moving Block System”. In: *Journal of Advanced Transportation* 2021 (Apr. 2021). Ed. by Keping Li, pp. 1–15. DOI: `10.1155/2021/6652531`. URL: `https://doi.org/10.1155/2021/6652531`.

[60] Nazir Ahmad Zafar, Sher Afzal Khan, and Keijiro Araki. “Towards the safety properties of moving block railway interlocking system”. In: *Int. J. Innovative Comput., Info & Control* 8.7 (2012), pp. 5677–5690.

[61] Huimin Zhang et al. “Collaborative real-time optimization strategy for train rescheduling and track emergency maintenance of high-speed railway: A Lagrangian relaxation-based decomposition algorithm”. In: *Omega* 102 (July 2021), p. 102371. DOI: `10.1016/j.omega.2020.102371`. URL: `https://doi.org/10.1016/j.omega.2020.102371`.

[62] Feiqin Zhu et al. “Decentralized Cooperative Control of Multiple Energy Storage Systems in Urban Railway Based on Multiagent Deep Reinforcement Learning”. In: *IEEE Transactions on Power Electronics* 35.9 (Sept. 2020), pp. 9368–9379. DOI: `10.1109/tpel.2020.2971637`. URL: `https://doi.org/10.1109/tpel.2020.2971637`.