

A MULTI-AGENT-BASED APPROACH FOR RESOLVING REAL-TIME TRAIN RESCHEDULING PROBLEMS OF LARGE-SCALE RAILWAY NETWORKS

by

JIN LIU

A thesis submitted to

The University of Birmingham

for the degree of

DOCTOR OF PHILIOSOPHY

School of Engineering

College of Engineering and Physical Science

The University of Birmingham, the UK

24 April 2020

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

ABSTRACT

Train delays occur often in daily railway operations due to a variety of incidents in railway circumstance. As a result, train operations are usually affected by unforeseen disturbances in railway circumstance which lead to unpredicted minor train delays in the railway network. With the connections and conflicts in the journey of the delayed trains, minor delays are likely to be transferred into a series of knock-on delays which may influence all the trains across the whole network. In this thesis, a multi-agent-based system is proposed to tackle train rescheduling problems in large-scale railway networks which considers strong coupling of train timetables among junction areas. With the evaluation approaches in the UK railway network, the proposed multi-agent-based approach can generate optimised train dispatching solutions within reasonable computational time compared to the First Come First Served dispatching rule, which provides a step forward in tackling real-time train rescheduling in a distributed way.

The thesis starts with illustration of the current railway traffic management system and recent research on decision support systems which generate real-time actions to considered rail traffic, and discusses the main challenges of the state of the art, which are bottlenecks in the development of decision support systems for large-scale railway networks. A systematic hierarchical multi-agent system for generating real-time solutions for train rescheduling problems in large-scale railway networks is proposed, which covers a mathematic formulation for the train rescheduling problem, an approach for decomposing a large-scale railway network, a Genetic Algorithm for local searching of train dispatching solutions and collaborative strategies to achieve final solutions with global feasibility. Furthermore, train rescheduling problems in railway bottleneck sections are tackled by a two-agent system (P2P system) with a series of negotiation approaches

between the two agents. To test the proposed approaches, UK rail infrastructures are applied to test the P2P and multi-agent systems, respectively. In a railway bottleneck, the computational results are compared with a traditional centralised algorithm and First Come First Served dispatching approach based on designed delay scenarios, and the pros and cons of the P2P system are studied. For a large-scale railway network, the solutions generated by the multi-agent system for different delay conditions are compared against a First Come First Served approach. The proposed multi-agent approach can generate train dispatching solutions better than First Come First Served in terms of proposed objective function.

The thesis also discusses the limitations of the proposed approaches and outlines the author's future work plan on railway traffic management systems. With the proposed decentralised system for large-scale railway networks, more traffic actions such as the reallocation of crew members and rolling stock can be considered. And to satisfy the time horizon of traffic control in the industry, the author also discusses his ideas for boosting the searching algorithm.

ACKNOWLEDGEMENTS

I have had such a wonderful journey since I started my Ph.D. research in 2015. Because Doctor Mietek Brdys died in early summer 2015, my Ph.D. study was switched from the integrated monitoring, control and security of Critical Infrastructure Systems to solving traffic management problems in railway networks. I appreciate Professor Clive Roberts and Doctor Lei Chen, my supervisors in my Ph.D. research, giving me another opportunity to reveal the beauty of research and explore the charm of practical engineering problems. With their help, my research was achieved in a series of remarkable milestones, a circle of pursuing novel questions, solving each engineering problem, and appreciating the discoveries. When I look back at the precious memories of the past few years, I really appreciate I have greatly benefited from my supervisors' inspiration, knowledge and experience in work and life. I feel very grateful to have all of you walking along with me through my research.

I would also like to extend my gratitude to the following people: Dr. David Kirkwood for sharing his expertise on coding and railway simulation, Dr. Ning Zhao for his suggestions and inspirations on my thesis and journal paper writing, and Dr. Gemma Nicolson for her helpful review and sparkling inspirations on my publications.

Besides that, I would like to thank all the members of the Birmingham Centre for Railway Research and Education for their kind help and support.

Finally, I would like to say thank you to my parents and fiancée, for their love and support, which is the most precious thing in my life.

Table of Contents

ABSTRACT I

ACKNOWLEDGEMENTS III

Table of Contents IV

List of Figures VII

List of Tables IX

List of Abbreviations X

Chapter 1 1

RESEARCH BACKGROUND AND OBJECTIVES 1

1.1 Background 1

1.2 Research objectives and contributions 3

1.3 Thesis structure 7

Chapter 2 9

OVERVIEW OF RAILWAY TRAFFIC MANAGEMENT SYSTEMS AND THE STATE OF THE ART OF REAL-TIME TRAIN RESCHEDULING RESOLUTION 9

2.1 Railway operation and timetabling 9

2.2 Railway traffic management system 11

2.3 The state of the art of railway traffic management systems 16

2.4 Main challenges 38

2.5 Summary 41

Chapter 3 43

MILP FORMULATION FOR TRAIN RESCHEDULING PROBLEMS AND NETWORK DECOMPOSITION APPROACH FOR LARGE-SCALE RAILWAY NETWORKS 43

3.1 Problem statement 43

3.2 MILP formulation for train rescheduling problems 50

3.3 Overview of MAS application to train rescheduling problems 58

3.4 Decomposition approach for a large-scale railway network 63

3.5 Summary 65

Chapter 4 68

TWO-AGENT SYSTEM FOR TRAIN RESCHEDULING PROBLEMS IN A RAILWAY BOTTLENECK SECTION 68

4.1 Overview of P2P system 68

4.2 System structure 71

4.3 Algorithm development 72

4.4 Negotiation strategies 82

4.5 Centralised system 92

4.6 Case study 93

4.7 Summary 102

Chapter 5 104

MULTI-AGENT SYSTEM FOR TRAIN RESCHEDULING PROBLEMS IN A LARGE-SCALE RAILWAY NETWORK 104

5.1 Overview of MAS applications 104

5.2 Structure of MAS for large-scale railway traffic management 109

5.3 Algorithm development 111

5.4 Collaborative strategy: Condorcet voting method 114

5.5 Case study 119

5.6 Summary 134

Chapter 6 136

CONCLUSIONS AND FUTURE WORK 136

6.1 Conclusions 136

6.2 Future work 138

Appendix A: Train timetables around Derby station 141

Appendix B: Single train simulator 150

Appendix C: Computational experiments of the case study in Chapter 5 156

Appendix D: Author's publications 163

References 164

List of Figures

Figure 1-1 Year-on-year PPM and CaSL (Gower, 2019) 2

Figure 1-2 Data flow within a decision support system in a railway traffic management system 4

Figure 2-1 Example of a railway dispatcher's workstation 10

Figure 2-2 Hierarchical structure of a traffic management system 12

Figure 2-3 Closed loop of real-time railway traffic management 14

Figure 2-4 Example of Alternative Graph modelling (D'Ariano et al., 2008) 18

Figure 2-5 Example of stage-to-stage transformation in a train rescheduling problem 22

Figure 2-6 Example of decision tree-based formulation for a train rescheduling problem 24

Figure 2-7 Example of discrete event railway system 31

Figure 2-8 Example of time event graph (initial making) (Goverde, 2010) 32

Figure 2-9 Example of time event graph for event 3 (Goverde, 2010) 33

Figure 2-10 Mainline area covered by ICONIS (Alstom, 2015) 37

Figure 2-11 Online product plan and data gathered by ICONIS (Alstom, 2015) 37

Figure 3-1 Example of single flat junction 44

Figure 3-2 Network examples including strong coupling 48

Figure 3-3 Example of decomposition approach to railway bottleneck 64

Figure 3-4 Example of decomposition approach to railway station area 65

Figure 4-1 P2P structure of a bottleneck in a railway network 72

Figure 4-2 Data exchange of the two agents in a bottleneck section in a control step 73

Figure 4-3 One-point and two-point crossover operators 77

Figure 4-4 Data structure of a generation after operators of classification and sorting 81

Figure 4-5 Step 1 of negotiation in condition 2; calculate the gap value 84

Figure 4-6 Step 2 of negotiation in condition 2 85

Figure 4-7 Step 3 of negotiation in condition 2 87

Figure 4-8 First step of negotiation in condition 3 88

Figure 4-9 Second step of negotiation in condition 3 89

Figure 4-10 Third step of negotiation in condition 3 90

Figure 4-11 Flow chart of decision-making in P2P system 91

Figure 4-12 North Stafford Junction and Stenson Junction bottleneck section 94

Figure 4-13 Simulation results of scenarios 1 to 10 98

Figure 4-14 Simulation results of scenarios 11 to 20 99

Figure 5-1 Two-pass data flow in a hierarchical MAS 108

Figure 5-2 Structure of MAS for large-scale train rescheduling problems 110

Figure 5-3 University of Birmingham microscopic simulator (example) 120

Figure 5-4 Network of Derby Station and its surrounding area and the definition of each junction in the case study 121

Figure 5-5 Accumulated delay at each junction area and station under FCFS (upper) and MAS (lower) approaches 130

Figure 5-6 Computational time of each agent in the MAS system 132

List of Tables

Table 4-1 Different scenarios and their data settings 96

Table 4-2 Maximum, minimum and average results for computational time and better trial percentage.....99

Table 5-1 Computational complexity at most of the whole problem and each junction agent in MAS.....122

Table 5-2 The exact values for computational complexity analysis in the case study ..123

Table 5-3 Summary of single train initiated delay scenarios.....125

Table 5-4 Summary of Multiple train initiated delay scenarios 126

Table 5-5 Summary of minor perturbation delay scenarios 127

Table 5-6 Summary of scenarios M_S28_S4 128

Table 5-7 Objective function values in class III 132

List of Abbreviations

Terms	Explanation / Meaning / Definition
ACO	Ant Colony Optimisation
AG	Alternative Graph
ANN	Artificial Neural Network
BF	Brute Force
BLF	Busy Line First
CDR	Conflict Detection and Resolution
DP	Dynamic Programming
DSS	Decision Support System
FCFS	First Come First Served
FLF	Fast Line First
GA	Genetic Algorithm
GPS	Global Positioning System
ITPS	Integrated Timetable Planning System
LTP	Long-Term Planning
MAS	Multi-Agent System
MILP	Mixed Integer Linear Programming
MINT	Manager on Integrated Network of Train Traffic
MRT	Mass Rapid Transit System
MSP	Meta-Super Peer

P2P	Peer to Peer
RECIFE	REcherche sur la Capacité des Infrastructures FErroviaires
SST	Stage-to-Stage Transformations
STP	Short-Term Planning
TD	Total Delay
TEG	Time Event Graph
TTOE	Timetables Order Enforced
VSTP	Very Short-Term Planning
WAD	Weighted Average Delay
WTA	Waiting Time Agreement

Chapter 1
RESEARCH BACKGROUND AND OBJECTIVES

1.1 Background

Rail transport is an efficient and high-capacity kind of transportation which affects large numbers of goods and passengers every day. In the latest report on transportation, the UK railway carried 17.0 billion tonne kilometres of freight and 1.71 billion National Rail passenger journeys in 2018. Although the goods moved by rail have declined as a result of a drop in the coal industry, passenger lines have become more intensive and grown more and more rapidly in recent years. The number of passengers in the UK has grown by 50% in the last decade, and is expected to double over the next 30 years (Sheeran and Kay, 2015). However, railway infrastructure and rolling stock has not been upgraded as fast as the passenger volume in recent years. As a result, the mainline railways in the UK are facing more intensive use of their services, especially in the centres of the main cities across the whole country. Statistical data on passenger and freight rail performance for the third quarter of 2018-19 (Gower, 2019) show that, both the punctuality and reliability of UK rail services have worsened, and the KPIs may keep getting worse in the future. The year-on-year trend of PPM (Public Performance Measure) and Cancellations and Significant Lateness (CaSL) is shown in Figure 1-1.

PPM is an indicator of National Punctuality; it has kept decreasing in the past 6 years and dropped significantly from 2017. CaSL indicates the reliability of railway transportation, and has also worsened from 2017.

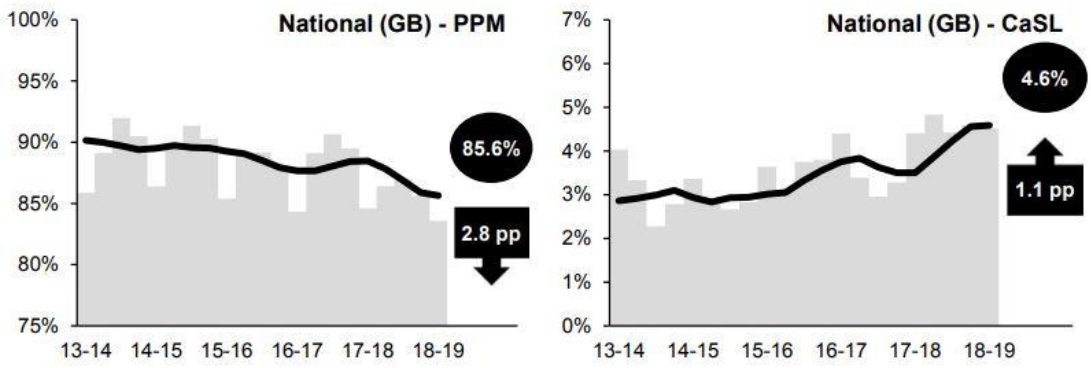


Figure 1-1 Year-on-year PPM and CaSL (Gower, 2019)

Therefore, the growing demand for railway transportation and its downgraded service quality have attracted more attention from both academia and industry. To fulfil the requirements of rail transport, more infrastructures have been planned and built, and the existing lines have been upgraded in recent years. As published by the UK government, to improve the rail infrastructure in the UK, Network Rail is spending £130 million every week in 2019, which includes the upgrade and maintenance of 20,000 miles of track, providing 6400 extra services and 5500 new train carriages and applying 170,000 more passenger trains (Open Access Government, 2019).

However, the upgrade of rail infrastructure and train services usually involves huge expense and a long time span, though they are the direct way to improve service quality. As a result, train operators are keen to find an approach to minimise disturbance to the railway network and keep services operating according to their planned timetable. In general, a train timetable works on four timescales (Capacity for Rail, 2015):

- Long-Term Planning (LTP): two timetables are produced for a year (starting from December and May, respectively) which cover 26 weeks each;
- Short-Term Planning (STP): done 18 weeks ahead of the planned timetable being taken, which usually results in missed services;

- Very Short-Term Planning (VSTP): changes made the day before the service is due to run, and which can be applied directly to the Integrated Timetable Planning System (ITPS);
- Ad-hoc planning: used when timetables cannot be carried out on that day due to the disturbances in railway circumstance.

Ad-hoc planning is usually called real-time train rescheduling and it has become a critical topic in recent years in both academia and industry. The process for generating a real-time solution for trains in the network is named decision-making. As our current railway is almost always operated under its best capability, real-time decision-making for train timetables and sequencing comes extremely vital for daily railway operation. Due to the interconnection of train timetables, generating an optimal solution for the whole network is still a great challenge. The real-time decision is usually made by a local human dispatcher who is trained and able to make a decision within his scope on the further journey of the train. However, sometimes, the dispatcher is not able to find the optimal solution and it is not possible to know the whole impact of his decisions across the whole network. As a result, real-time decision-making for railway networks, especially large-scale networks is still an open question, and in recent years lots of researchers have shown their innovations on this topic.

1.2 Research objectives and contributions

According to recent research, the most common way to tackle real-time train rescheduling problems is by applying different advanced algorithms. In general, an algorithm is developed with respect to the layout of the rail network and applied to the traffic management centre of the local area. The algorithm keeps updating the movement status of trains in its local area and makes an optimised solution based on the conflicts in its

network. Figure 1-2 shows the data flow of a decision support system in a railway traffic management system. A decision support system is usually attached with traffic management control centre. In general, network data (including distance, gradient and speed limits) and train parameters (including traction parameters and penalty rate) are saved in its local database; initial timetables which cover all train schedules are provided at the day. The decision support system will monitor all trains in its local control area with a designed time horizon and predict their passing time against their initial timetables at conflict areas in a local network. The train rescheduling algorithm with be triggered if train delay (disturbance) is detected and will lead to knock-on delays in conflict areas; the decision support system will generate a solution to minimise knock-on delays in conflict areas and transmitted into future signalling and interlocking actions.

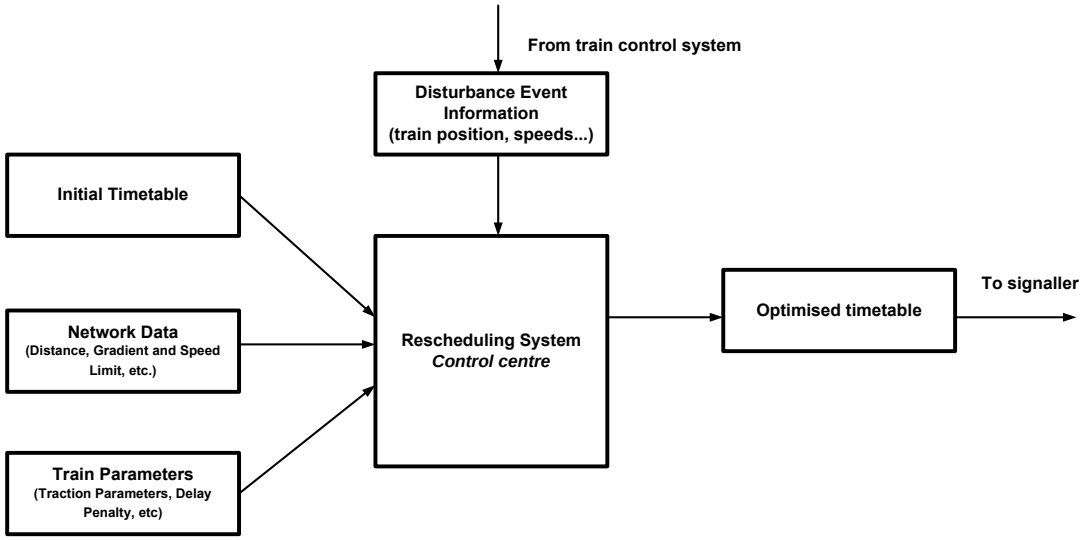


Figure 1-2 Data flow within a decision support system in a railway traffic management system

However, the developed system still has the following bottlenecks.

1) The algorithm can be time-consuming, and the computational time may exceed the time horizon of real-time control when the considered train volume and railway

network come larger. In other words, enlarging the train volume and network considered will lead to a huge searching space, and finding an optimised solution in a larger searching space usually costs more computational time.

2) With an increased searching space, it is hard for the algorithm to find the optimal solution. Because the algorithm usually evaluates a number of solutions within its time horizon, a larger searching space can also lead to poor quality of the final result.
3) The algorithm should be developed specifically with respect to the layout of the network because trains are operated with different constraints, and the layout of the local network can also affect the decision of its local operation. However, the algorithm is hard to fit in all different layouts of the entire network, which means the proposed algorithm has poor scalability.

In this thesis, the author proposes a Multi-Agent System (MAS) which can solve train rescheduling problems for a large-scale railway network. With regards to the proposed approach for a large-scale railway network, retiming and re-sequence actions will be considered, and the MAS can provide a new method for decomposing the large railway network and an appropriate approach for decoupling the interconnection among train timetables. Moreover, with regard to the nature of an MAS, better scalability will be applied into the railway traffic management system, and due to the decrease in the search space, a better-quality final solution and less computational time can be promised. In summary, the following contributions will be illustrated in the following sections in this thesis.

1) A Mixed Integer Linear Programming (MILP)-based model is formulated to state the train rescheduling problems for a large-scale rail network. Within the proposed model, strong coupling of train timetables is emphasised, and retiming and reordering of trains is taken into consideration for further optimisation.

2) The structure of an MAS to tackle train rescheduling problems for a complex railway traffic management problem is developed, including a Peer to Peer (P2P) two-agent system specifically developed for railway bottleneck sections and a hierarchical MAS for a large-scale railway network. Furthermore, to plug the MAS into a current railway traffic management system, an appropriate approach for decomposing the railway network is also discussed. Due to the decomposition of a railway network, the searching space of a train rescheduling problem of a large-scale rail network is degraded into several sub-problems, which enables better scalability of the decision support system.
3) To solve the MILP problem, a designed Genetic Algorithm (GA) is proposed in this thesis; to fulfil the operational constraints and collaborations among agents in P2P and MAS, different operators to ease their further collaborative strategies are designed. Furthermore, the key innovations in the designed GA are its extra operators (classification and modification); the classification enables individual diversity of the algorithm and provides a group of individuals to future collaboration, and the modification operator corrects the unfeasible solutions in a generation with some greedy rules which improves solution quality and execution time of the algorithm.
4) An innovative collaborative strategy for MAS is proposed, including a series of negotiation protocols and a Condorcet voting mechanism to ensure the final solutions generated by MAS have the scope of the entire network and choose the most majority of each sub-network.
5) Two British railway case studies are applied to test the performance of the P2P system and MAS, respectively. Train delay scenarios which are designed as disturbances in railway operation, are applied in each case study. The computational experience is also illustrated in each case study. The results show that P2P system can provide better

solutions against FCFS and centralised approach in designed scenarios and the proposed MAS is able to improve dispatching solution with up to 34.11%, which identifies the MAS is a promising approach to solve train rescheduling problems in large scale networks. Furthermore, the pros and cons of applying MAS are also discussed.

1.3 Thesis structure

The thesis is organised as follows:

Chapter 1: The background of railway traffic management, research objectives, contributions of the thesis and thesis structure are illustrated.

Chapter 2: This chapter summarises the principle of railway operations, the architecture of the current railway traffic management system, the state of the art of train rescheduling approaches and the main challenges.

Chapter 3: A train rescheduling problem is formulated using MILP, including detailed operational constraints and strong coupling among train timetables. The approach to decoupling the train timetables and dividing the railway network is also discussed in this chapter, which degrades a large-scale railway network into simple junction level.

Chapter 4: The systematic P2P system for railway bottlenecks is shown. The chapter starts from the recent application of P2P systems in different research areas, and then illustrates the structure of a P2P system for a railway bottleneck section, the local searching algorithm (GA) and its negotiation strategies between the two agents in this system. Finally, the advantages and disadvantages of the P2P system with the computational experience of the case study are discussed.

Chapter 5: This chapter mainly introduces a hierarchical structured MAS for a large-scale train rescheduling problem. The recent applications of MAS is reviewed. Then,

the MAS structure for a large-scale railway network is shown. The MAS keeps using the same search algorithm as the P2P system, and its further Condorcet voting approach is illustrated. The test case is executed in a microscopic railway simulator with practical train timetables and infrastructure in Derby station and its surrounding area. The computational results are shown finally and discussed with the benefits of MAS.

Chapter 6: This chapter summarise the conclusions and findings of this thesis, and the future work is also presented.

Chapter 2

OVERVIEW OF RAILWAY TRAFFIC MANAGEMENT SYSTEMS AND THE STATE OF THE ART OF REAL-TIME TRAIN RESCHEDULING RESOLUTION

This chapter introduces the detailed functions of a railway traffic management system and its actions to operate daily railway services. Then, the state of the art of railway traffic management systems is shown in both academia and industry. Furthermore, to link up with the innovations in this thesis, the main challenges in current systems and the author's opinions on solving the research bottlenecks are stated.

2.1 Railway operation and timetabling

Within a railway system, trains are operated based on their timetables which are called working timetables. These working timetables are planned at least half a year ahead and consist of train arrival and departure times, minimum dwelling time in the station area, and passing time at specific nodes in the train's journey, such as junctions, sidings, looping, etc. The main objective for designing and updating the working timetables is to maximise the capacity and service frequency of passenger lines, minimise the journey time and connection time for passengers and maintain the high-quality services (Barber et al., 2009; Demitz et al., 2004). To ensure the robustness of the timetables, extra time is inserted into timetable which can absorb minor delays caused by disturbances in railway circumstance. Besides that, based on the working timetable, railway operators will generate their rolling stock and crew scheduling to satisfy the plan, and further rostering is also desired on every day of the schedule. So, railway operation is a chain of actions which relate to the allocation of large resources. However, the extra time inserted into the timetable is not sufficient to absorb all the delays in the network. Due to the speed limits

of a train and its connection with other services, recovery from significant delays is hard to achieve and inevitable for daily railway operation.

As a result, real-time rescheduling is usually done on demand by a local human dispatcher when an unabsorbed delay is in sight and likely to be transmitted to other trains. The dispatcher can monitor the computerised workstation with all the information in its local area which includes rolling stock, infrastructure, crew information and the delays to each train. An example of a dispatcher's workstation is shown in Figure 2-1.

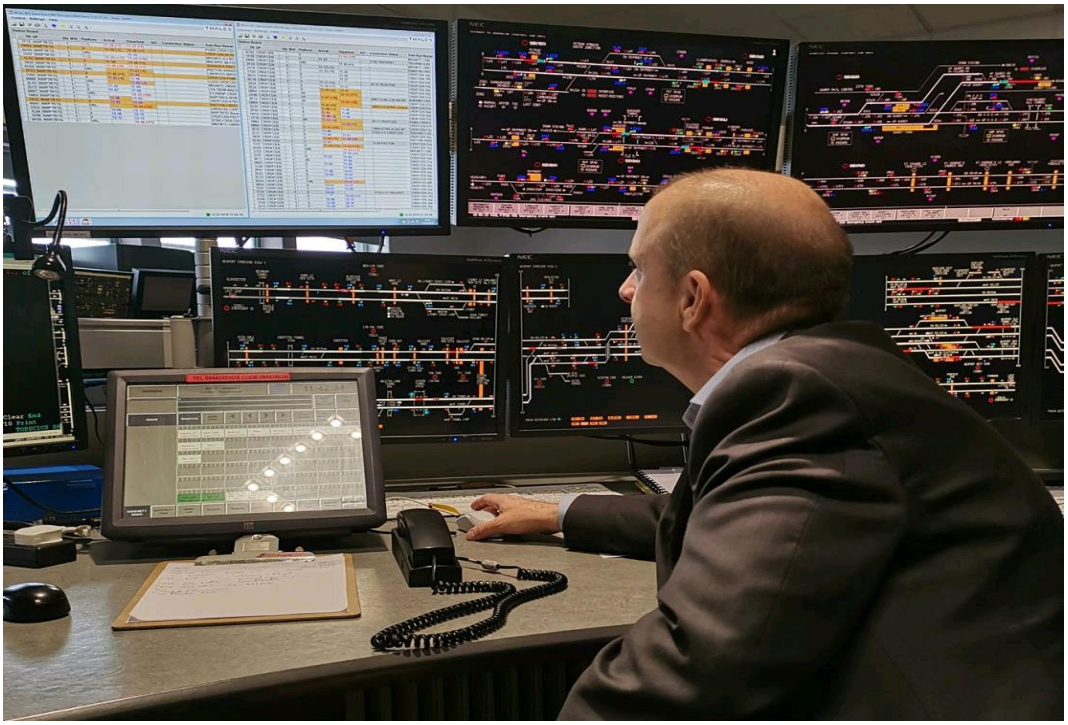


Figure 2-1 Example of a railway dispatcher's workstation

Because the dispatching decision may lead to a series of effects within the entire network, the rescheduling actions are classified into three types (Veelenturf et al., 2016): timetable rescheduling, rolling stock rescheduling and crew member rescheduling. Timetable rescheduling, also called train rescheduling, is the most common way to adjust traffic flow in a railway network, and mainly includes train retiming, reordering and rerouting. Rolling stock rescheduling is used when planned rolling stock service cannot be ensured and

usually needs a new allocation of rolling stock in its local area, and crew rescheduling is the generation of a new allocation or rostering to satisfy the demand of train operation (Nielsen, 2011). Because timetable rescheduling only generates ad-hoc timetables for trains, with no further influence on rolling stock and crew planning, it is much easier to be achieved in daily operation (Dai, 2016). In this thesis, only timetable rescheduling including retiming and reordering is considered. To simplify the terminology used in this thesis, the train rescheduling problem or train dispatching problem is defined as generating retiming and reordering actions for the trains in the considered network, to optimise rail traffic flow.

2.2 Railway traffic management system

The railway traffic management system is a key element which provides harmonised control of the entire network and makes commands to ensure daily services. The actions achieved by a railway traffic management system are covered by several subsystems in railway circumstance. Figure 2-2 shows the hierarchical structure of a traffic management system and interaction among the systems on traffic management problems.

In general, a railway network is divided into several subareas to ease the achievement of rail services. The traffic management centre is the top layer in the traffic management system, providing the long-term timetables for daily service and real-time management command to optimise train movements. The centralised traffic control centre generates movement authorities, line information and interfaces to interlocking (Ferravante, 2006). To avoid train conflict during movement, the route of a train's journey is ensured by line-side signalling (i.e., the occupation of a future track circuit should be checked and turn the traffic signalling to corresponding colour) and track switches (i.e., switches are

movable track elements powered by point machines, enabling trains to move from one track to another) whose actions are guaranteed by local interlocking and signalling blocks.

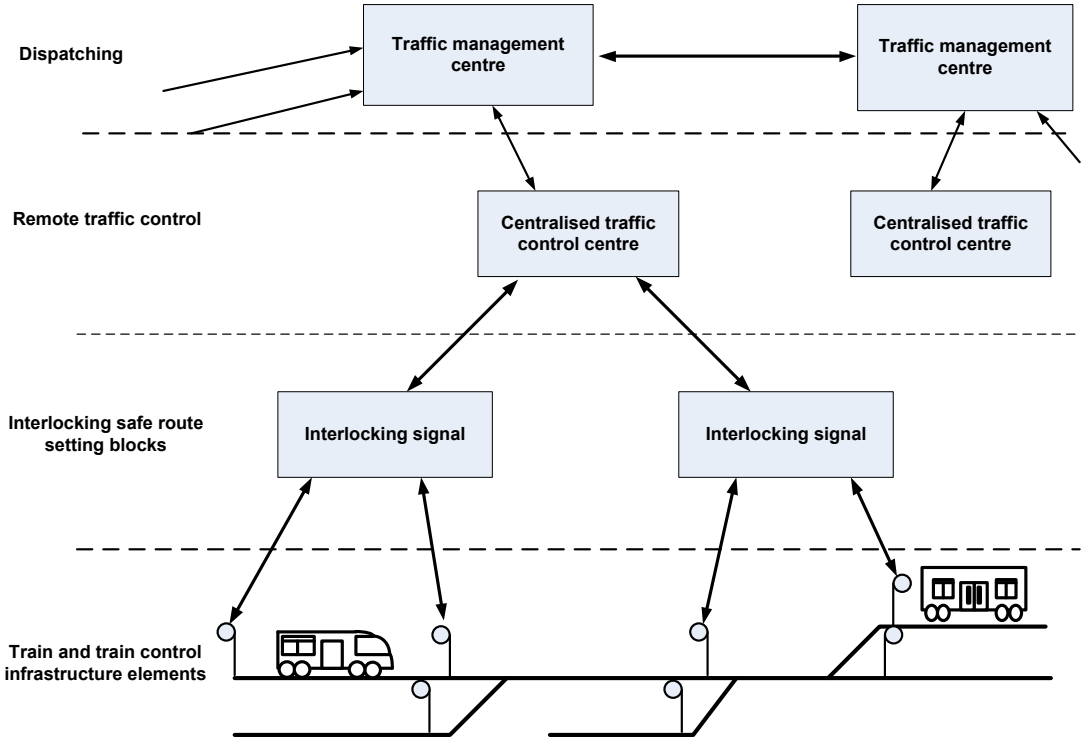


Figure 2-2 Hierarchical structure of a traffic management system

From the definitions given by Pacciarelli, (2013), the objective of a rescheduling problem in a railway system is set as searching for a new timetable without conflict for future rail services. According to different control objectives, the railway traffic management problems can be roughly categorised into three classes:

1) Real-time rescheduling

   Real-time rescheduling, also called Conflict Detection and Resolution (CDR), focuses on rail operations up to 45 min in the future; its objective is to minimise train primary and knock-on delays with no rerouting and replatforming actions (Törnquist, 2007).

2) Delay management

   Delay management optimises train timetables using the same strategies as real-time rescheduling plus rerouting of passengers and replatforming of trains (Dollevoet and Huisman, 2014). The main purpose of delay management is minimising passenger waiting time at station areas and enhancing the connections between services, which covers a time horizon roughly between 10~15 min and 2~3 hours (Schöbel, 2001). Furthermore, a Waiting Time Agreement (WTA) is applied to define the maximal waiting time for a train in a station to ensure its attached connections.

3) Disruption management

   Disruption management is triggered when significant delays are suffered in the network. To satisfy the planning services as much as possible, a new timetable will be generated with new allocation of rolling stock and crew members (Fang et al., 2015; Schachtebeck and Schöbel, 2010). Disruption management covers at least 2 hours, and railway operators usually have emergency timetables defined in advance which can be selected and applied fast (Ghaemi et al., 2017).

Real-time rescheduling in a railway system consists of a closed loop of traffic control among subsystems. Figure 2-3 illustrates the closed loop of railway traffic management. Classic real-time rescheduling can be summarised into the following steps (Albrecht et al., 2008):

1) Starting from planned timetable;
2) Monitoring and transmission of real-time data to operation system;
3) Detection of train conflicts;
4) Resolution of train conflicts;
5) Generation of ad-hoc planning and applying to future control command.

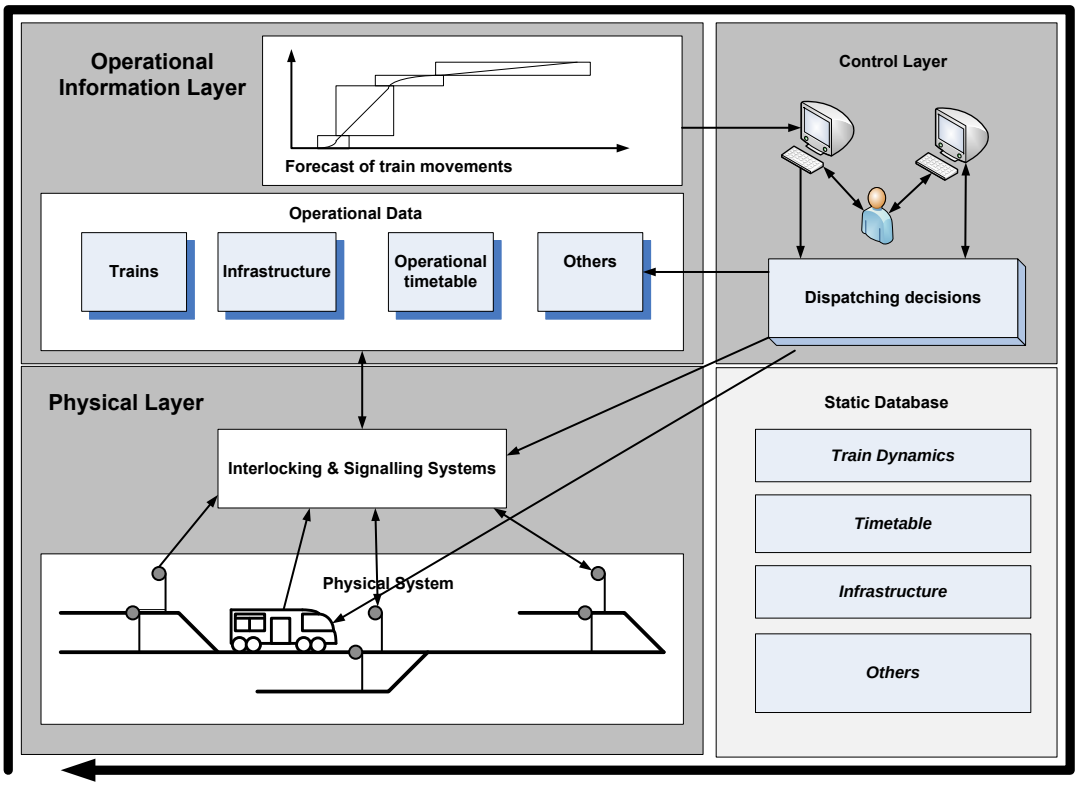


Figure 2-3 Closed loop of real-time railway traffic management

The detailed closed loop of real-time rescheduling is organised into four layers: physical layer, operational information layer, control layer and static database layer (Liu et al., 2018). Within the physical layer, real-time train position and movement information are gathered from the interlocking and signalling system, which is called track-based train positioning. A train can be pinpointed by the occupation of a track circuit and the actions of trackside signalling in its journey. Alternatively, the position of a train can also be detected by an embedded Global Positioning System (GPS) with updates at a certain frequency (Kecman, 2014). Then, the train's position and movement information are transmitted to the operational information layer and its future movements are predicted by that layer based on operational data which includes train information, infrastructure planning, operational timetables, etc. In this stage, future conflicts among trains in its area are identified and shown on the computerised workstation of human dispatcher in the

control layer. Human dispatchers usually make their dispatching decisions for the trains in its local network using their experience of real-time train dispatching. Because the local human dispatcher not only needs to solve the conflicts in the local area but also needs to consider future effects on the entire network, dispatching strategies differ for different types of disturbance in the network. To the author's knowledge, the most common rules applied in real-time train rescheduling problems are (Dai, 2016):

a) First Come First Served (FCFS): dispatch the train which arrives at the signal before the conflict area first.
b) Timetables Order Enforced (TTOE): dispatch the trains in the same order as their operational timetable.
c) Fast Line First (FLF): dispatch the trains from a fast line over the trains from a slow line when conflict happens.
d) Busy Line First (BLF): dispatch the trains from a busy line, that is one with larger number of trains following within a given distance.

As the dispatching decisions are made, the decisions are transmitted back to the physical layer and converted into future interlocking and signalling actions. At the same time, the static database layer will generate log files on the real-time information, which are saved in its database and will be used for future analysis of rail service quality.

In general, TTOE is the simplest approach for rescheduling problems because trains are dispatched with the same order in their schedule, and the original operational timetable is usually planned with the consideration of different operational requirements among different railway operators. As a result, train connections can be absolutely satisfied with TTOE. However, TTOE usually enlarges knock-on delays in junction areas and leads to unnecessary penalties for train operators. FCFS is the most widely used approach, this

approach ensures the knock-on delay in its local area is minimised. As a result, the FCFS is usually chosen as a baseline in past research articles to show the improvement achieved by proposed algorithm against human dispatcher. However, due to their different service requirements of local trains and long-distance trains; FCFS does not consider the priority of considered trains, which possibly leads to congestion in a mix-operated line. Furthermore, to overcome the drawbacks of FCFS, FLF or BLF is applied occasionally to achieve a specific rescheduling purpose.

2.3 The state of the art of railway traffic management systems

To solve a real-time train rescheduling problem, the train movement and conflicts should be formulated in an appropriate model, and solutions are supported by searching algorithms. In the recent research on train rescheduling problems, the proposed approaches differ with respect to their different formulation methods and algorithms. Based on recent review papers on railway traffic management problems (Cacchiani et al., 2014; Corman and Meng, 2015; Fang et al., 2015), the state of the art of train rescheduling problems is illustrated in this section.

Due to the nature of railway operation, train movements can be viewed as a series of occupations of track blocks in the network. As a result, a train rescheduling problem can be modelled as a job-shop problem with extra constraints (D'Ariano et al., 2007a). For more details on job-shop problems, readers can refer to the work of Błażewicz et al. (1996). The jobs and operations in a job-shop problem correspond to trains and the travelling of trains, respectively. The processing time in an operation in a job-shop problem states the travelling time of a train in a specific track block, and railway operation conflict is presented as two or more jobs asking for the same operation within a time slot.

The formulation of train rescheduling can be roughly classified into two levels with their modelled details in railway network, the macroscopic and microscopic modelling. The macroscopic level presents the railway as nodes and arcs which stand for stations and journeys between stations, respectively. And the microscopic level also considers every block section with its attached signals and detailed train kinematic model. Obviously, microscopic models can illustrate train movements and infrastructure behaviours more carefully. However, microscopic models usually need more computational time and larger memory space. So the trade-off between the two levels of formulation is the first step in solving a train rescheduling problem, and its effects on further solution searching are profound (Cacchiani et al., 2014). In this thesis, the contributions of recent researches with their modelling approaches are summarised, and the applied algorithms with their pros and cons are discussed.

2.3.1 Microscopic Models

2.3.1.1 Alternative Graph (AG)

AG is the most commonly used graph-based approach, and was first proposed by Mascis and Pacciarelli (2002); D'Ariano et al. (2008) applied AG to a real-time railway traffic management problem and proposed a sophisticated system, Railway traffic Optimisation by Means of Alternative graphs (ROMA). Publications with modelling on AG account for most research articles on railway traffic management problem. An example of an AG is shown in the figure below.

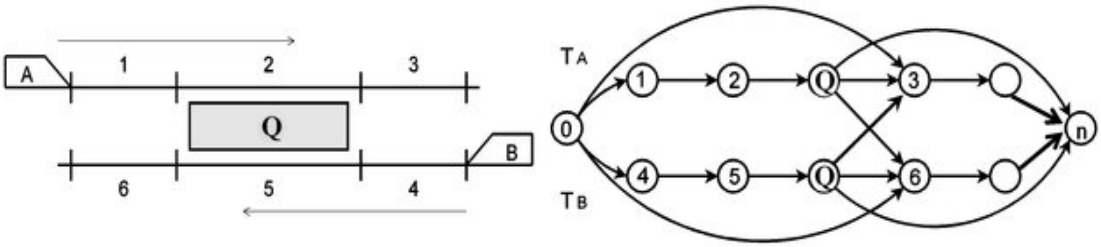


Figure 2-4 Example of Alternative Graph modelling (D'Ariano et al., 2008)

Within an AG, track blocks are numbered first and stated as a node in the AG. The movement of a train is modelled as a train passing a series of nodes in the network linked with arcs which consist of all the operational constraints for each train in the network. In the AG example above, train A and train B depart from the left side and right side of the network, respectively, and move to opposite directions, with dwelling in station Q. So, to organise the AG for train A and train B, both of them start from node 0 and end at node n. Train A passes nodes 1, 2, Q and 3, which stands for its journey from track section 1 to track section 3 with a dwelling in station Q. Similarly, the branch of train B in AG stands for its journey from track circuit 4 to track circuit 6 with a stop at station Q. Arcs between track section nodes illustrate the travelling time for each train in a specific track section which is measured with a train kinematic model. Moreover, the arcs between node 0 and node 3 or 5 state the planned departure time of train A or train B, and the arc between node Q and node n stands for the minimum travelling time of each train from station Q to the end of its journey. The arcs from node Q in one branch of AG to the node after node Q in another branch are the minimum connection times of train A and train B in station Q.

Branch-and-bound is the first type of algorithm applied in AG to solve a real-time train rescheduling problem (D'Ariano et al., 2007a). Due to the huge searching space of the AG, the authors proposed dynamic and statistic implication rules to speed up the branch-

and-bound algorithm which considers relationships among alternative arcs and off-line knowledge of network topology. Moreover, the algorithm starts by generating an initial partial solution and searches for a list of partial solutions. The algorithm chooses one of the partial solutions from the list and builds new solutions with alternative arcs not included in former solutions. The proposed approach was tested on a railway network in a Dutch network around Schiphol Amsterdam Airport with a timetable including 54 services hourly. The computational time limit was set as 120 s and the results were compared with three heuristic approaches; the computational instances show the algorithm could reduce secondary delay by up to 100 s. In a further study, D'Ariano et al. (2007b) extended their research, considering speed coordination of neighbouring trains to solve the train rescheduling problem. A variable-speed model was proposed to update the train speed profile and make a decision on decelerating a train or not. The computational experience shows that a dynamic speed model for trains can absorb more train delays across the railway network and provide more flexibility in rescheduling decisions. The paper also proposes an approach on optimisation of automatic cruise control which provides a new branch on a later traffic management topic. The same researchers then looked at relaxing timetabling constraints to provide more flexibility of train arrival and departure times at station areas (D'Ariano et al., 2008). The arrival and departure times were replaced with time windows organised by minimum and maximum arrival time and dwelling time. The branch-and-bound algorithm was also applied to optimised the delays of trains at stations and reduce the delays by more than 30% in the same network around Schiphol Amsterdam Airport.

Furthermore, a decomposition approach has been illustrated to cut the searching space into a trackable scale and solve the large problem with synchronisation at each time interval (D'Ariano and Pranzo, 2009). The proposed approach can generate better

solutions with a reasonable time horizon which drops the delay to around 20 s from 400 s; and this article addresses the analysis of advanced techniques of problem decomposition which provides a new scope for tackling a huge optimisation problem in railway traffic management. Corman et al. (2009) proposed a green wave policy to avoid train speed regeneration in railway corridors when conflicts are inevitable in railway bottleneck sections. The proposed approach was tested on two complex Dutch networks, and computational results showed that the green wave policy can reduce train delays and energy consumption without train speed profile optimisation. The green wave policy provides another approach outside searching algorithms which uses little computational time and provides good-quality dispatching solutions. Corman et al. (2010b) used a tabu searching algorithm for train conflict resolution; the AG model considers retiming and rerouting of real-time rail operation, and the algorithm was compared with branch-and-bound approach including heuristic strategies to show the pros and cons of the tabu searching. The algorithm was applied in a Dutch railway network and was able to find optimal solution in small instances and reduce the delays in the network by 15% in the large instances within a computational time limit of 180 s (Corman et al., 2010b). A distributed approach to decouple the train rescheduling problem in railway bottlenecks was proposed by Corman et al. (2010a), The proposed interaction policy restricts the searching space and ensures the decision made by each part of area is feasible globally. The simulation results show that the distributed approach can generate an optimised solution in less computational time than a centralised approach. However, due to the restriction on searching space, the distributed approach is not able to find better solutions than the centralised approach in terms of objective function in this article. Furthermore, a Petri Nets and heuristic searching algorithm was proposed by Wang et al. (2016) for solving a train rescheduling problem in a double track rail corridor; the algorithm

generates new markings for the algorithm based on railway operational and safety principles. The proposed algorithm can search for an optimal or near optimal solution within seconds and recover the timetables from small delays in designed scenarios. Wang and Goverde (2016) illustrated a multi-phase train trajectory optimisation with constraints on signalling and train operations. The objective in the proposed approach is to minimise energy consumption in non-delay situations, and minimise the delay in operational timetables in a delay situation (Wang and Goverde, 2016).

2.3.1.2 Stage-to-Stage Transformations (SST)

An SST was proposed to model the shortest path problem which is formulated into several operation stages, in each of which only one operation is considered. In each stage, costs between two neighbouring steps are evaluated based on their optimisation objective. An example of formulation on railway conflicts is shown in Figure 2-5. The example considers four trains approaching a single flat junction, and the whole dispatching process for four trains is divided into stages which dispatch one train in each stage. As a result, only one train will pass the junction in each stage, and a stage consists of all the possible conditions. The arcs between two neighbouring stages state the time interval between dispatching the two trains in each condition, and the optimal solution is the branch which has the lowest total time interval from stage 0 to stage 4. To find an optimised solution, an appropriate searching approach should be applied to the SST model.

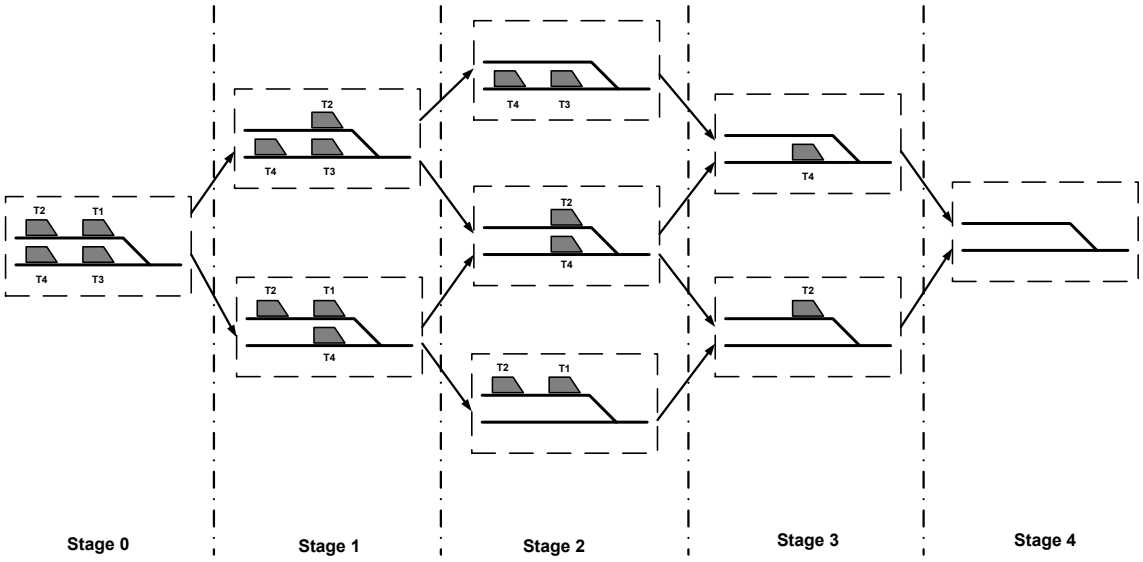


Figure 2-5 Example of stage-to-stage transformation in a train rescheduling problem

The most common searching algorithm for SST is Dynamic Programming (DP). Lew and Mauch (2007) proposed a control strategy to describe an efficient request-relocation neighbourhood evaluation procedure, and DP including heuristics was applied to search for an optimised solution. The computational results showed that DP can speed up the searching process against a straightforward approach and reduce the computational time by about 27%. Tabu searching and local searching have also been applied in some tested instances, and tabu searching performs better than local searching in all instances (Ho and Haugland, 2011). Based on SST formulation, a dynamic schedule synchronisation and energy saving approach for train control was proposed with the purpose of minimising the connection time between a Mass Rapid Transit System (MRT) and road-bound transport and ensuring efficient operation. The optimised timetable is generated in real time, and the computational results show the proposed approach can also reduce the energy consumption is some scenarios (Oettich, 2002). Fan (2012) applied DP to solve a real-time train dispatching problem in a benchmark railway network and compared the searching result with a Brute Force (BF) algorithm which evaluates all the possible

solutions in the train rescheduling problem considered. Four steps are proposed to be organised within the DP which starts from modelling the whole dispatching process into SST, states the time cost between two neighbouring stages, evaluates every stage and finally finds the optimal solution. The computational experience shows that DP can generate an optimal solution five times faster than BF due to the SST structures the whole searching space with different stages and avoids recalculate the cost in each stage every time when the same dispatching decision is made. However, SST and DP still have a bottleneck in their searching capability; the SST formulation inherently has to model the whole searching space, which is impossible for large and complex searching problems.

2.3.1.3 Decision tree

A decision tree is another frequently used graph-based modelling approach which structures the whole searching space with a tree-shaped graph whose branch states a possible solution for a considered problem. Figure 2-6 shows an example of a decision tree-based formulation for a train rescheduling problem. The example illustrates four trains approaching a flat junction. The root of the decision tree (node 0) states the initial stage in which the trains are waiting for the dispatching decision. The first layer of the decision tree represents all the possible decisions for dispatching the first train in the junction area, and subsequent levels state the possible decisions for successive trains. Each possible decision links a node in the upper layer with a branch in the decision tree graph, which indicates the cost of the decision in terms of the objective function in the optimisation. Similar to SST, the optimal solution can be found by evaluating all the branches from the initial node to the bottom of the decision tree.

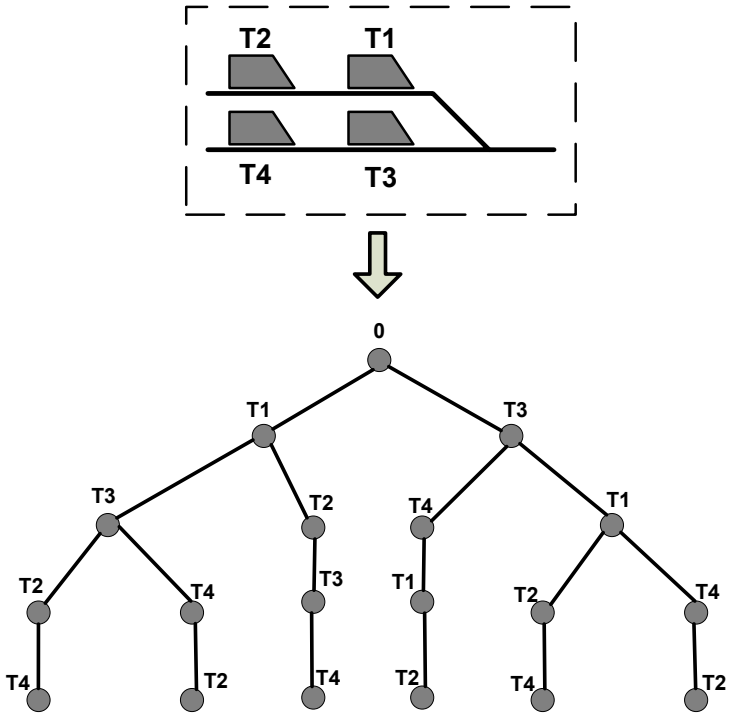


Figure 2-6 Example of decision tree-based formulation for a train rescheduling problem

Josyula et al. (2018) modelled a train rescheduling problem into a binary decision tree and proposed a parallel algorithm to reduce the computational time of optimisation when the searching space is huge. The algorithm can utilise a multi-core structure with a novel heuristic strategy which searches the tree for optimal decisions in most of the designed disturbance scenarios. However, the unstable algorithm may lead to significantly long computational time in a few scenarios. The heuristic strategy cuts the tree branch and minimises the searching space with multiple factors including solution costs, delayed trains and retimed trains. Furthermore, the parallel computing structure speeds up the algorithm so that it is 10.5 times faster than the sequential algorithm on a single processing core, and is a further 4.86 times faster when applied on eight cores. Wang et al. (2014) proposed a rescheduling approach for high-speed trains which improves the decision tree formulation with train speed restriction information. To fulfil the demand for capacity before the rescheduling action, a capacity evaluation model with deduction factor is

applied. A bi-layered rescheduling approach is also proposed to reduce computation complexity, which tackles train rerouting problems in the upper layer and minimises the total delay and number of impacted trains in the lower layer. The illustrated case study shows that the proposed approach can generate a promising solution within around 5 s even in complex scenarios. However, the authors also discuss the drawbacks of problem decomposition: the layered resolution approach leads to multi-objective issues and potential improper rerouting of an trains, which ignores the consideration of Pareto points (Wang et al., 2014). Another algorithm for learning and pruning an oblique decision tree was proposed by Shah and Sastry (1999), who evaluated the algorithm branches by different split rules based on the concept of the degree of linear separability. The authors applied a correlation-based optimisation approach named the Alopex algorithm to find a split rule which optimises the evaluation function at each node. Ismail et al. (2010) proposed a P2P searching approach for fast searching with a decision tree which is based on super peer-based architecture. Within their approach, relevant domains are predicted by Meta-SuperPeer (MSP) which is organised by peers with similar interests. The test results show that the proposed approach has better searching capability with respect to computational time and precision. Fan et al. (2012) illustrated eight different algorithms to solve a train rescheduling problem; three of the algorithms are organised based on decision tree elimination, Ant Colony Optimisation (ACO) and tabu searching, and applied these three algorithms in a benchmark railway network. Within the computational results, decision tree elimination was the fastest algorithm because bad performance solutions are cut in the early stages when a large delay is found, which avoids evaluating too many solutions. However, the optimal solution may be discarded due to its large delay in the early layer which makes decision tree elimination miss the optimal solution. ACO was the second-fastest algorithm and can provide an optimal solution in three quarters of

the designed scenarios. Tabu searching was the worst solution for the train rescheduling problem in the context of the scenarios tested. The optimal results cannot be guaranteed in these scenarios; however, the solutions generated by tabu searching are still better than FCFS and the execution time is promising. Moreover, a hybrid algorithm including ACO and GA is proposed to overcome the searching bottleneck in ACO which is able to find the best solution within the shortest time (Fan et al., 2012). Dai (2016) proposed a blackboard system based on the work of Fan et al. (2012), which provides global control of large-scale railway network. A supervisor is structured to monitor the operation process in the whole network, and different searching algorithms are selected with respect to different delay conditions in the network.

2.3.1.4 Mathematical programming

Aside from graph-based formulations, mathematical programming is also a common approach to modelling train rescheduling problems. The development of mathematical programming started in 1948 when Dantzig proposed the Simplex Method for solving linear programming problems. In general, mathematical programming considers an objective function, also called the cost function, to be minimised or maximised with respect to a series of equality and inequality constraints. Mathematical programming can be classified with respect to the type of objective function and constraints. This section provides an overview of the three most common mathematical programming methods: linear programming, nonlinear programming and integer programming.

The earliest branch of mathematical programming is linear programming. Within a linear programming problem, both the objective function and constraints are linear equations. The first approach to solving a linear programming problem is the Simplex Method proposed by Dantzig (1948); some extensions of the Simplex Method can be found in the

work of Smith (1987). Nonlinear programming concerns mathematical programming problems whose objective function and/or constraints are nonlinear, and is classified into two groups, unconstrained nonlinear programming and constrained nonlinear programming. Unconstrained nonlinear programming is identified when the objective function of nonlinear programming is continuously differentiable (Sinha, 2006). To solve an unconstrained nonlinear programming problem, most algorithms start from an initial solution and search for neighbourhoods based on the calculation of derivative values of the objective function, such as Steepest Descent method (Fliege and Svaiter, 2000), Newton's method (Flett, 2008) and Quasi-Newton's method (Fletcher, 2013). However, most nonlinear programming problems usually include a series of inequality and equality constraints which is named constrained nonlinear programming. The resolution approaches for constrained nonlinear programming can be divided into two groups: direct methods including the method of feasible directions, reduced gradient method and Newton's method (Sanderson and Skelly, 2007), and duality methods including the penalty function method and classical Lagrange method (Aardal et al., 2005). Another branch of mathematical programming is integer programming whose variables are constrained to only integers. Due to the complexity of practical problems, integer programming problems are also categorised into pure integer programming and mixed integer programming problems. Pure integer programming concerns the optimisation of variables restricted to integer values, and mixed integer programming considers variables including integers and continuous variables. To deal with an integer programming problem, the approaches for resolution can be classified into three families: branch and bound (Marinescu and Dechter, 2005), cutting plane methods (Marchand et al., 2002) and meta-heuristic methods (Fischetti and Lodi, 2011).

To model the train rescheduling problem, the train passing times at specific nodes in the railway network are considered, and the passing times are usually optimised as continuous variables. The passing sequence of trains is also a key element within the optimisation and is usually indicated by a series of binary variables. Furthermore, a common objective function set for a train rescheduling problem is the total delay of trains in the control area, which is a linear objective function. MILP is proposed for tackling real-time railway traffic management problems which include train routing and rescheduling under railway perturbation conditions. The proposed MILP models railway infrastructure with accurate granularity, and constraints are applied to reduce the searching space and improve the execution time in simulations. A rail network in France was used to test the heuristic approach under several delay scenarios (Pellegrini et al., 2014). The railway traffic management system was extended with additional elements characterising railway reality and proposed with a RECIFE (REcherche sur la Capacité des Infrastructures Ferroviaires) software platform. A heuristic algorithm was used to solve rerouting and retiming problem of railway traffic and its performance is better than other approaches implemented based on experimental analysis. However, the authors also state the proposed approach has a bottleneck when a large number of alternative routes are free for each train, which makes it difficult to find the optimal solution for the scenarios considered (Pellegrini et al., 2015). Pellegrini et al. (2016) used a detailed analysis method to evaluate the actual effect of real-time traffic optimisation on the railway network according to several KPIs of the French railway operator (SNCF). The RECIFE system generates real-time train rerouting and rescheduling solutions for the designed scenarios and compare them with the human dispatcher's solutions; the computational results show that the solutions made by RECIFE are better than those by the human dispatcher. Valid inequalities and reformation approaches are proposed to ease the searching process of RECIFE. The proposed

approach is able to present all the facets of projection of the problem's convex hull. A case study showed the computational experience of testing on four French rail infrastructures, illustrating the benefits of a shorter execution time. This research provides a method for tackling large-scale train rescheduling problem (Pellegrini et al., 2019). Samà et al. (2016) grouped train alternative routes into several subsets and searched for the best rerouting strategy via a designed ACO based on integer linear programming. The proposed approach was tested on two practical rail networks in France, and the computational experiments showed that the ACO is able to improve the network throughput by up to 56%. Moreover, an innovative differential evolution algorithm was proposed by Chen et al. and applied to solve train rescheduling problems in railway junction areas (Chen et al., 2010) and bottleneck sections (Chen et al., 2015) based on the formulation of mixed integer programming. Weighted Average Delay (WAD) was set as the objective function in optimisation, and three kinds of train order correction approaches were proposed to generate better solutions than FCFS. The simulation results showed that the innovative differential evolution algorithm can generate a better solution in 2 to 3 s against FCFS and automatic route setting, which indicates the algorithm can satisfy the real-time optimisation of railway traffic. The railway traffic rescheduling problem an N-tracked railway network was modelled by Törnquist and Persson (2007) and heuristic strategies were applied to solve the MILP problem. A Swedish railway network whose traffic has a high degree of interaction was used as a case study, and the computational results showed that the proposed approach has a good searching capability, sufficiently fast for real-time control.

2.3.2 Macroscopic models

2.3.2.1 Discrete event model

Microscopic modelling approaches desire the update of train moving status at every time increment, which could require a huge amount of computation and cost long computational time in practice. So, a discrete event modelling approach which presents the whole processor to an event-based model has been proposed to state the railway operation. The first discrete event model was proposed by Breusegem et al. (1991), who described the railway system as series of discrete events consisting of nodes and arcs. The nodes can be indicated by the train ID, location, node type (departure, arrival and passing) and scheduled event time; and the arcs link two neighbouring events which can be defined by train ID, moving status (run or stop) and the minimum processing time between the two events. The headway and connection between trains are described with specific arcs in the station event and junction event, and an event is ready to be executed when all the former arcs to this event are completed (Kecman et al., 2013). The following Figure 2-7 shows an example of a discrete event model.

Figure 2-7 describes a double track line through the area which consists of three stations: station A, station B and station C. Two trains are dispatched from opposite directions and are expected have a connection at station B. Nodes 1 to 4 indicate the travel of train 1 which commutes between station A and station B, and nodes 5 to 8 state the travel of train 2 between station B and station C. the travelling time and dwelling time are modelled as solid arcs with weighted times in the discrete event model, and the connection time between train 1 and train 2 is shown as dotted arcs in the model.

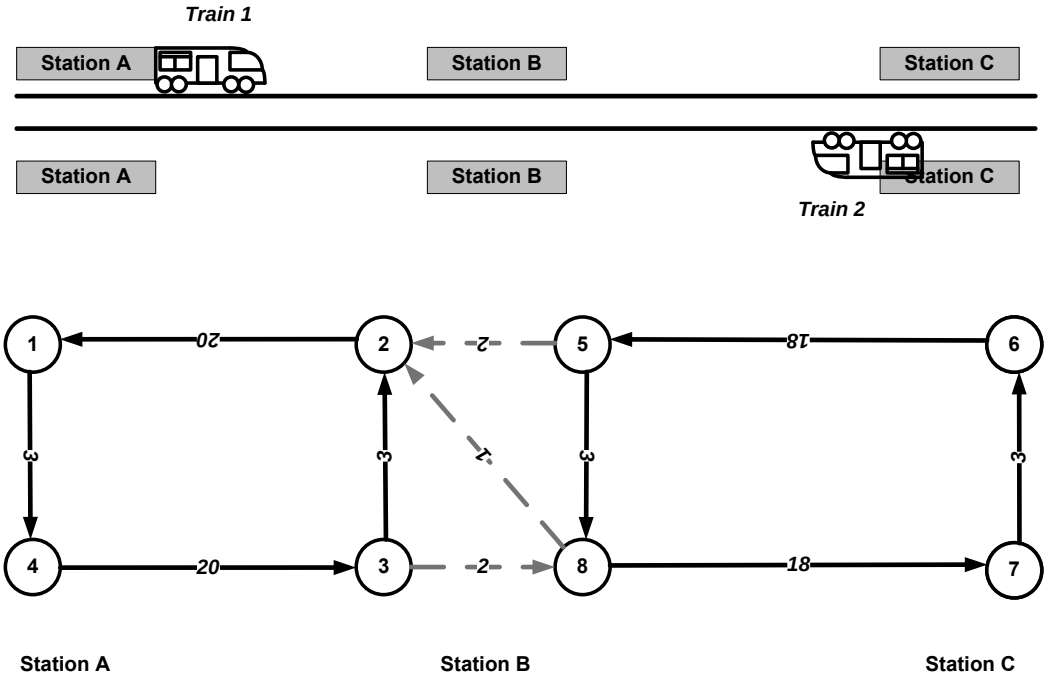


Figure 2-7 Example of discrete event railway system

Discrete event model was applied by Kecman et al. (2013) to describe services in a metro system. Arrival and departure of trains at each platform are considered in this model, and the corresponding values on arcs are related to both trains and platforms. Ho et al. (1997) considered isolated and proposed a traffic controller. A probabilistic approach was discussed to establish the optimal schedule when consecutive new trains enter at uncertain times. A single junction network was studied to test the proposed approach and compared with the FCFS dispatching rule. The computational results showed that the solution generated can improve the performance in terms of the objective function (weighted delay) against FCFS by approximately 10%. The computational time of the proposed approach is short enough to support real-time control. Moreover, the authors also indicated that the discrete event-based model is more efficient than the discrete time-based simulation. Besides that, a greedy travel advanced strategy has been developed to tackle railway traffic management problems under perturbations in the schedule based on discrete event modelling (Dorfman and Medanic, 2004). The proposed method can generate solutions

up to three times better than the default dispatching method. The algorithm can also avoid deadlock in the network, due to the discrete event treating a blocking aspect with a single section. Furthermore, an extended algorithm was proposed for double track corridors including extra nonlocal information and train characteristics data.

2.3.2.2 Time event graph (TEG) model

TEG is a modelling approach based on Petri Net which was first developed by Murata (1989). It was first used to model train operations in railway network by Goverde (2010). An example of a TEG is shown in Figure 2-8.

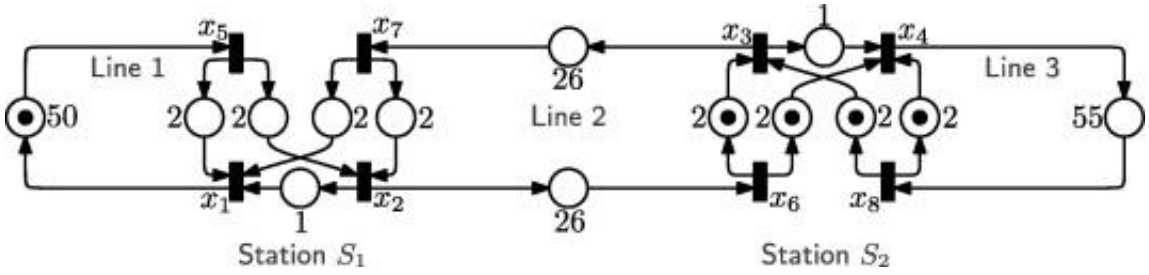


Figure 2-8 Example of time event graph (initial making) (Goverde, 2010)

Within a TEG, transitions corresponding to the events are described by bars (***x₁*** to ***x₈*** in Figure 2-8), and places corresponding to the processes are described by circles; arcs corresponding to the connection between places and transitions which state a possible direction of a future event. Each process includes an attached number which stands for its time consumption. So, the movements of a train can be described by a series of processes which state the travelling time of its journey in a part of the network. Furthermore, train arrival and departure are illustrated by events in station areas. A token is marked in a place to show the position and moving status of an individual (a train) in TEG, so the TEG is also called time-marked graph in some literature. The TEG in Figure 2-8 illustrates a railway network including three rail lines, line 1, line 2 and line 3. Line 1 and line 3 are circle lines which service the local area around stations ***S₁*** and ***S₂***, respectively, and line 2

is a commuter line between stations S_1 and S_2. Services can be applied in a single line or switching among the lines. Five trains are shown in Figure 2-8, which are marked by tokens. From the left side of Figure 2-8, the first train is operated on line 1, the second train to the fifth train are dwelling at station S_2, and the four trains in station S_2 come from line 2 and line 3 and will be dispatched to line 2 and line 3, i.e., the train between events x_6 and x_3 comes from line 2 and will be dispatched to line 2 or line 3, and the train between events x_6 and x_4 comes from line 2 and will be switched to line 3. And similarly, the train operated from events x_8 to x_3 and x_4 states the trains from line 3 and will be dispatched to line 3 and line 2.

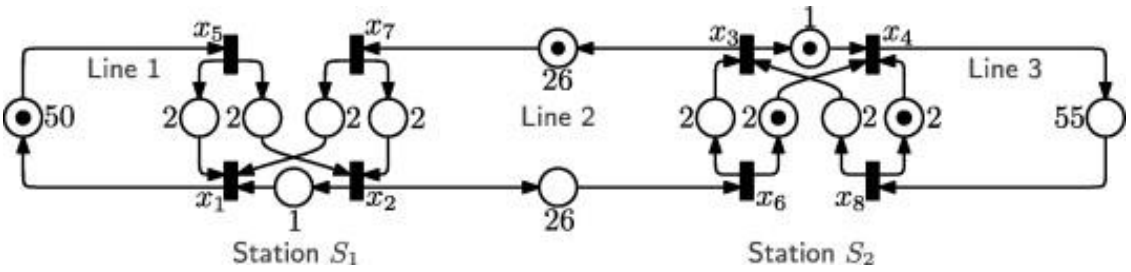


Figure 2-9 Example of time event graph for event 3 (Goverde, 2010)

Figure 2-9 describes the next stage of Figure 2-8. Between the two stages, event 3 (x_3) is triggered. The second train is moved from the process between x_6 and x_3 to the process between x_6 and x_4, and the fourth train is dispatched from station S_2 to line 2. Moreover, the operation of trains across the whole network is illustrated by a series of events.

Goverde (2010) developed an effective delay propagation algorithm for rescheduling problems in a large-scale network based on TEG formulation. With the consideration of the running time and the dwelling time of each train, dwell buffer times are optimised to minimise knock-on delays in the network, and the delay propagation is analysed with the realisability and stability facets of the timetables generated. A optimisation model was proposed by Zhen and Jing (2016) to minimise the waiting time of passengers in stations

and enhance passenger satisfaction. The Beijing subway was modelled in TEG to test the proposed approach, which reduced travel time cost and the cost of cancellations by 13%. A model-predictive-control-based approach is proposed by De Schutter et al. (2002) to resolve a max-linear system with soft and hard synchronisation constraints for a railway network. The main purpose of the optimisation is minimising the sum of deviations from the planned timetable and unachievable connections. Due to the moving time horizon in model-predictive-control approach, the method can be applied in on-line applications to deal changes in the system parameters.

2.3.2.3 Other models

A train rescheduling algorithm was proposed by Chikara et al. (2007) based a railway framework on train rescheduling patterns. The proposed approach can generate a new plan for trains under railway disruption, especially for scenarios which include large scale disruptions and require suspending train operations for up to an hour. An automatic train rescheduling algorithm was developed by Hirai et al. (2006) based on rescheduling pattern description language R. A language processing system called 'R-interpreter' was applied to interpret descriptive contents in R language, and the rescheduling patterns were modelled as 'R-rules' which make a real-time decision based on the events in the railway network. The final computational results show that the algorithm can work efficiently with the designed scenarios and is suitable for practical use.

A prototype of MINT (Manager on Integrated Network of Train traffic) was proposed by Missikoff (1997); it integrates information management and conflict resolution functions in an integrated system based on an objective-oriented design and analysis approach. The knowledge-based application is modelled by a powerful conceptual language (TQL++), and the MINT system can generate executable coding for the considered problem

automatically. Railway traffic management problems are classified into 27 different classes, and 43 rule-based actions are developed to tackle different kinds of problem. The final decision is generated by a series of actions within the MINT system which include concerning, monitoring and conflict solving. A double track line in south of Rome was considered in the case study, and the computational experiment showed the proposed approach can tackle complex traffic control problems.

Barta et al. (2012) analysed the major reasons for freight train delays and developed a model for delay propagation of train connections at intermodal terminals based on a Markov chain. Unexpected events like unplanned maintenance, strikes, railroad works and traffic congestion are considered with operational management of freight trains in the proposed model. The real-time assignment decision is made based on real-time information from the rail network including train movement status, current delay and recoverable delay in the future. The model proposes to examine the historical data and classify the effects of different assignment methods of their capability on absorbing and amplifying train delays.

Due to the challenges on train timetabling problem for a long-haul single-track network, a problem space search meta-heuristic algorithm was developed by Albrecht et al. (2013) for a large-scale railway timetabling problem with the consideration of real-time information on train movements and scheduling track maintenance. The proposed approach generates an integrated timetable for track maintenance, and the approach can be applied as an operational tool to recover the scheduling in disruption conditions. A single-track rail network in Australia which includes 57 crossing loop and 50 trains per day was studied in this article, and the computational experiment shows that the proposed algorithm can quickly generate a large number of alternative train timetables and track maintenance plans in a dynamic environment when disruptions occur.

An Artificial Neural Network (ANN) was applied to mimic the real-time decision-making process of a human dispatcher, and was trained and tested with historical data on conflict resolution in a Turkish railway line (Dündar and Şahin, 2013). The trained ANN was used as a benchmark and compared with a GA whose objective is to generate a rescheduled timetable and minimise the weighted delay in the network. The GA was tested under 51 hypothetical problem instances, and the test outcomes showed that the GA is better than the human dispatcher in terms of computational time and the value of the objective function. Moreover, for small problems, the computational time of GA can be up to 81 s; for large problems, the computational time of GA is about 331 s.

2.3.3 Advanced railway traffic management systems in industry

With the efforts of the researchers and engineers in the railway industry, some railway traffic management systems have been developed with decision support systems attached, to assist human dispatchers make real-time decisions for train conflicts. The most well-known systems are ICONIS and Tranista developed by Alstom and Hitachi, respectively. In this section, ICONIS is used as an example to illustrate the main functions of a traffic management system product.

ICONIS (Integrated CONtrol and Information System) is a traffic management system being developed by Alstom from 2013 in collaboration with experts in operational research. The mainline railway to be monitored and operated with ICONIS is shown in Figure 2-10.

ICONIS covers the security, signalling, passenger information, resource management and maintenance systems. The real-time information (online product plan) including timetable, infrastructure rolling stock and crew is managed by ICONIS, and ad-hoc decisions which is mainly the rescheduled timetable with interlocking and signalling rules are provided by

ICONIS. The online information and actions covered by ICONIS are shown in Figure 2-11.

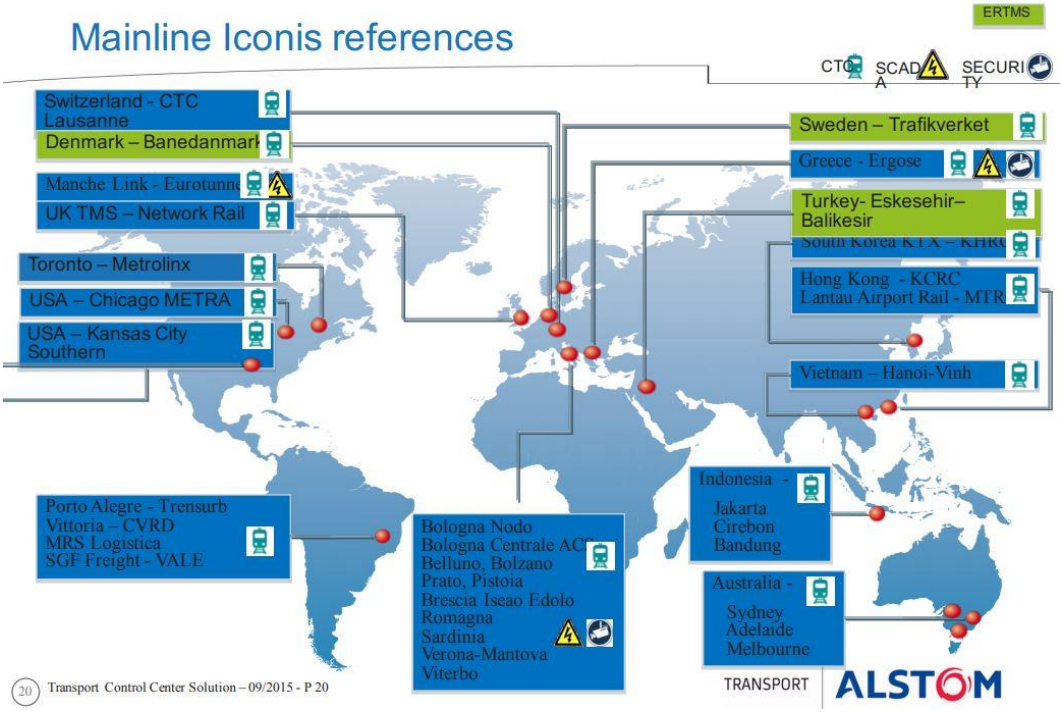


Figure 2-10 Mainline area covered by ICONIS (Alstom, 2015)

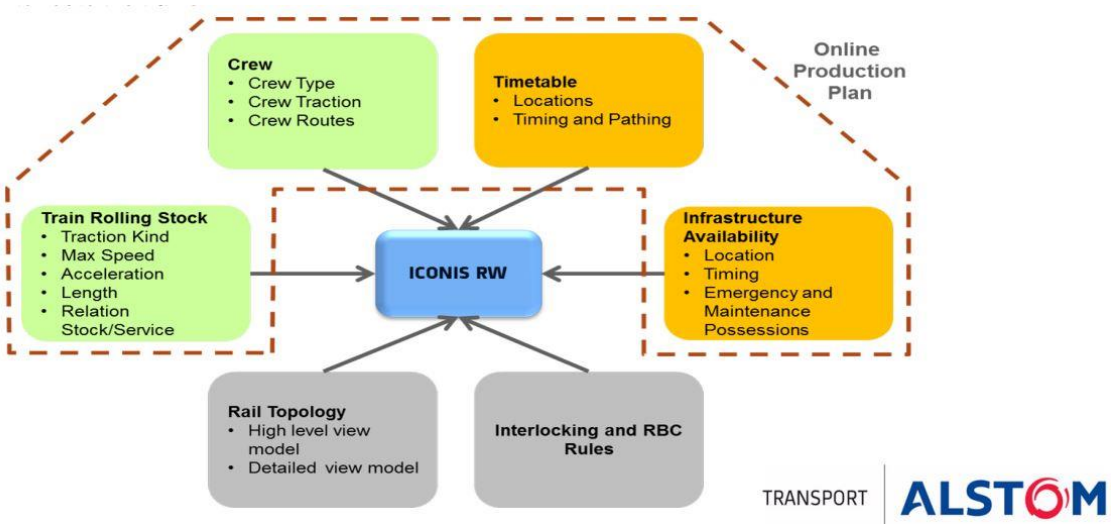


Figure 2-11 Online product plan and data gathered by ICONIS (Alstom, 2015)

For train rescheduling problems, train delays are classified into three groups: small instances (disruption time horizon of 15 min), medium instances (30 min) and large

instances (60 min), and for each kind of instance the computational time limits are set as 2 s, 5 s and 10 s, respectively. However, with the most advanced solving techniques, it is still a challenge to make an optimised decision within the proposed time limit (Fischetti and Monaci, 2017). Due to intellectual property protection of ICONIS, the author cannot find more information on the resolution approach applied in ICONIS. To minimise the computational time of the ICONIS, Fischetti and Monaci (2017) assumed that optimisation can be executed on 'any reasonable hardware', which could be a local cluster with tens of computers in parallel, and a Cplex MILP solver under its default mode is used to generate real-time decisions for train rescheduling problems in a cluster with 1, 4, and 8 quad-core PCs in parallel. The computational time of the Cplex solver in most of the instances considered (a total of 26 instances) is usually under 5 s on 4 and 8 PC clusters, and only three cases have an execution time under 2 s.

2.4 Main challenges

In the section above, the main approaches for solving railway traffic management problems are reviewed. With the innovations of researchers in this area, more advanced algorithms have been proposed. However, there are still drawbacks in the proposed approaches which are illustrated below.

The main obstacle which bottlenecks current research on railway traffic management system is the scalability of the proposed train rescheduling. So, it is still challenging to solve the train rescheduling problem of a large-scale railway network, and the main challenges to extend these proposed approaches are prolonged computational time and solution quality on raising of railway scale considered. Based on the literature reviewed above, most authors have applied heuristic approaches to reduce computational time in solving large-scale train rescheduling problems, which use series strategies to minimise

the searching space of the problem considered and avoid the evaluation of poor-performance solutions within the searching algorithm. However, the development of heuristic approaches usually requires good railway operation experience on train dispatching, and the application of heuristic approaches has to ensure that optimal solutions are not discarded. The heuristic approaches proposed have been restricted to local railway networks, and sometimes restricted to specific railway traffic management problems and scenarios. As a result, it is difficult to apply the heuristic approaches to large-scale train rescheduling problems.

Furthermore, the principle under searching algorithms for train rescheduling problem is evaluating a number of possible solutions and finding the optimal or optimised solution for the problem considered. So, the searching space becomes huge when a large-scale network is considered, and to generate an optimised solution within the time horizon of real-time control becomes difficult. Namely, the quality of the final solution and the computational time are inherently conflicting targets in the optimisation as the network considered is extended.

Moreover, to ensure the feasibility of the final solution, most of the proposed algorithms consider the train rescheduling problem as an entire optimisation problem which considers all the railway timing points and operational constraints together. In other words, most of the proposed approaches deal with the train rescheduling problem in a centralised way. However, our current railway traffic management system is a hierarchical distributed system which operates railway traffic in a greatly self-dominated way. And all the proposed approaches lack a discussion on plugging in the proposed approach to practical railway traffic management system though the approaches show great capability for finding an optimised solution compared to a human dispatcher in simulation; and when dealing with large-scale train rescheduling problem, a traffic management centre may face

problems with huge complexity and synchronisation issues of updating all the trains' movement status in the large-scale networks. To the best of the author's knowledge, few articles discuss how to achieve their innovations in practical railway traffic management system. And with the consideration of the structure of practical railway traffic management system, it is better to develop a decentralised system to tackle railway traffic management problems in large-scale networks which avoids huge computational complexity and data synchronisation for real-time decision-making.

Finally, railway timetables have strong interactions and interconnections, so an optimised solution generated by a real-time decision-making system of a local rail area may not be suitable for the trains' future journey, and sometimes the solution generated is not feasible or optimal globally. Once a wrong decision is made by a local dispatcher, the decision may lead to a domino effect across the whole network, and degrade the operation of train timetables. So, making a train rescheduling decision without taking trains' future journey into account is ill-advised. Namely, the local decision making should include a global view in its optimisation. Few articles talk about the effects of local decision-making on the entire network.

In this thesis, the author is keen to improve the drawbacks mentioned above by applying an MAS. The main benefit of an MAS is its great scalability to large-scale problems, and its decentralised structure. Within an MAS, agents are designed with social ability which ensures the process of their local decision making has the scope of their common global goal. And with the decomposition of a large-scale railway network, the problems are departed into reasonable size which is suitable for real-time traffic control for a single agent; so applying MAS approach to train rescheduling problems can potentially solve the conflict between computational targets of execution time and solution quality. Furthermore, the decentralised structure of MAS is inherently similar to our current

railway traffic management system, so a decision-making system based on MAS is easier to be plugged into current railway traffic management system than centralised approach. As a result, solving train rescheduling problems based on MAS concept comes a promising approach.

2.5 Summary

This chapter provides an overview of railway operation and the current railway traffic management system and reviews recent research in the relevant areas.

The daily operation of a railway system is centralised on train working timetables, and based on the working timetables, the rolling stock and crew schedule are generated. Due to the complexity of railway circumstance, train delays, which lead the trains to depart from their working timetables are inevitable. Train delays may be transferred to other trains and affect trains in the whole network, so rescheduling actions including timetable rescheduling, rolling stock rescheduling and crew member rescheduling are taken by human dispatchers. Timetable rescheduling is emphasised in this thesis.

The closed loop of real-time railway traffic management is illustrated. Current railway traffic management systems are organised by a hierarchical structure with data flows in between. The closed loop in a railway traffic management system is illustrated by data exchange among the physical layer, operational information layer, control layer and static database. Furthermore, the steps of real-time rescheduling are illustrated and the most commonly used strategies to generate real-time dispatching solution are discussed.

This chapter also reviews the state of the art of research on train rescheduling, which is mainly organised by different modelling approaches and their specific searching algorithms. The modelling approaches are categorised into two groups: microscopic and macroscopic approaches. Mathematical formulations such as MILP and graph-based

formulations such as AG, SST and decision tree are the most commonly used approaches for modelling railway operations at microscopic level, and discrete event models and TEG are the main modelling approaches at macroscopic level. An example of an automatic routing system in practice (ICONIS) is illustrated, and the importance of the development of advanced railway traffic management system is discussed.

Finally, the main challenges in current research are summarised as scalability of the decision support system, which means the real time train rescheduling problems for large scale railway networks are not properly resolved in academics and industry yet. The main contribution of this thesis is to propose a systematic methodology for resolving the real time train rescheduling problems in large scale railway networks using MAS concept.

Chapter 3

MILP FORMULATION FOR TRAIN RESCHEDULING PROBLEMS AND NETWORK DECOMPOSITION APPROACH FOR LARGE-SCALE RAILWAY NETWORKS

In this chapter, a typical train rescheduling problem is illustrated, and the main challenges to tackling train rescheduling problems with decentralised system are discussed. An MILP model is proposed in this chapter to formulate a train rescheduling problem for a large-scale rail network, and a network decomposition approach is developed to degrade the rescheduling problem to simple junction level. The main contributions in this chapter are: (1) applying MILP on modelling train rescheduling problem for large scale railway network, and strong coupling of train timetables are indicated by binary coupling variables in the optimisation. (2) a decomposition approach to depart railway network which is designed to degrade searching space of a train rescheduling problem. In the formulation and decomposition approaches, the strong coupling among railway timetables is emphasised.

3.1 Problem statement

Trains are operated under circumstances full of disturbances, and the disturbances usually result in primary delays in the journey of a train. The main reasons for primary delays are technical failures, running at lower speeds than scheduled, prolonged alighting and boarding times of passengers, and bad weather conditions (Albrecht et al., 2008). These disturbances are difficult to predict, and minimising the effects of the disturbances are important in daily railway operation. Rescheduling actions which adjust the passing sequence and passing times of trains at railway junctions or station areas are the most

common strategies for minimising the transmission of primary delays into knock-on delays to other trains in the network. In this section, train rescheduling actions at a single junction are illustrated first, and then train rescheduling for a large-scale railway network is discussed.

3.1.1 Train rescheduling in a single junction rail area

Railway junctions, places where two or more rail routes converge or diverge, are a common and critical element in railway infrastructure. In the context of rail transportation, a junction whose attached routes are in the same level is called flat junction or level junction. Figure 3-1 shows an example of a flat junction.

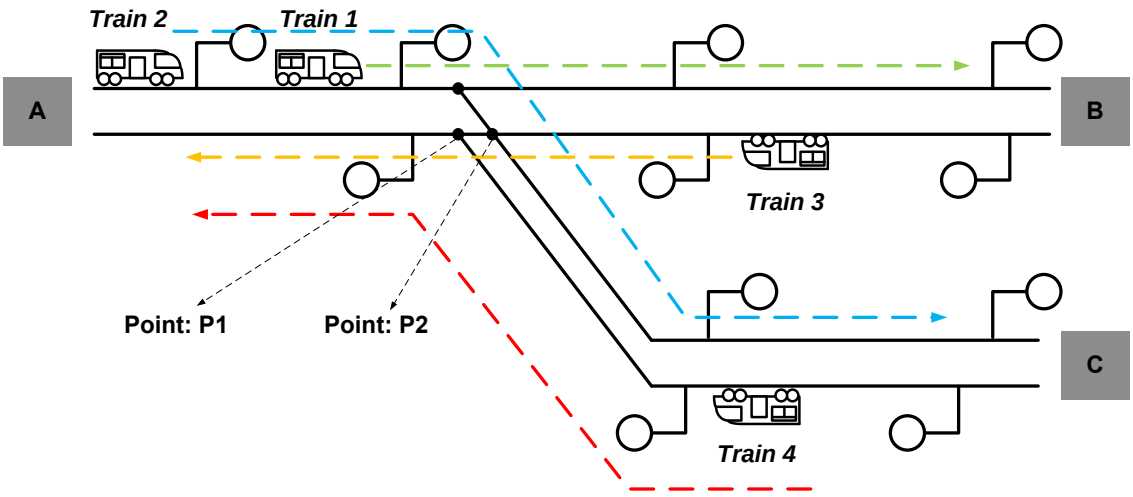


Figure 3-1 Example of single flat junction

In the example above, the junction is a flat junction, and a point machine (also called a switch) is placed in the junction area to make the interlocking actions for rail operation, which ensures the desired routing for an approaching train. The trains are operated among positions A, B and C. Four trains with the same travelling priority are shown in the example, and their moving directions are marked as dotted arcs: train 1 and train 2 are dispatched from A to B and C, respectively; train 3 is dispatched from B to A, and train 4

is dispatched from C to A. Operational conflicts may happen at points P1 and P2, i.e., train 2 and train 3 may ask for their routes at the junction area if they approach it at the same time at point P2. Similarly, train 3 and train 4 are also conflict trains at point P1. As a result, trains have to pass the junction with ordered actions of the point machine and signalling. In normal conditions, train timetables are designed to be conflict-free. However, if a train suffers a significant delay in its operation, the delay will be transmitted to the other trains in the junction area, which degrades the quality of the service and the capacity of the network. So, real-time operation of the junction to avoid generating knock-on delays is critical. In the example above, the timetable for the trains is designed as follows:

Node / Train ID	**A**	**Junction**	**B**	**C**
Train 1	07:00:00	07:01:00	07:02:30	--
Train 2	07:01:00	07:02:00	--	07:03:30
Train 3	07:04:00	07:03:00	07:01:30	--
Train 4	07:05:00	07:04:00	--	07:02:30

In the designed timetable, trains are organised with the passing order: train 1, train 2, train 3 and train 4, and the related interlocking and signalling actions will be achieved with the same sequence with the designed timetable. When the disturbances affect the travel of train 2, it will arrive at the junction area late by about 3 min. With the same passing sequence as designed in the original timetable, the timetable in practice will be:

Node / Train ID	**A**	**Junction**	**B**	**C**
Train 1	07:00:00	07:01:00	07:02:30	--
Train 2	***07:04:00***	***07:05:00***	--	***07:06:30***
Train 3	***07:07:00***	***07:06:00***	07:01:30	--
Train 4	***07:08:00***	***07:07:00***	--	07:02:30

Due to operational constraints, train 3 and train 4 have to wait at the signal before the junction until train 2 has passed the junction area. So, the primary delay of train 2 leads to future knock-on delays on train 3 and train 4 if no rescheduling action is taken. However, it is obvious that changing the train passing sequence to train 1, train 3, train 4, train 2 can avoid transmission of the primary delay on train 2; the retimed timetable can be:

Node / Train ID	**A**	**Junction**	**B**	**C**
Train 1	07:00:00	07:01:00	07:02:30	--
Train 2	***07:04:00***	***07:05:00***	--	***07:06:30***
Train 3	07:04:00	07:03:00	07:01:30	--
Train 4	07:05:00	07:04:00	--	07:02:30

With the retimed timetable above, the rescheduled sequence moves train 2 from the second to the fourth position in the sequence, and train 3 and train 4 are dispatched before train 2. As a result, train 3 and train 4 can pass the junction area as planned in their original timetable, and transmission of the primary delay on train 2 is avoided.

In summary, an example of train rescheduling in single junction area is illustrated in this section. A single junction area is the most elementary section for making train rescheduling decisions in a railway network, and a proper rescheduling action is able to stop the transmission of delay and minimise the knock-on effect in the junction area.

3.1.2 Train rescheduling for a large-scale railway network

In train rescheduling problems for a large-scale railway network, junctions as well as stations and looping areas where it is possible to change the train moving sequence in the network, are considered. To minimise all the delays across the network, rescheduling actions can be taken in every rescheduling area. However, train timetables have strong interconnections with each other, so adjustment to the timetable of a single train can lead to a series of corresponding adjustments to timetables in the entire network. Moreover, trains usually have connections in stations, which is a strong constraint on making rescheduling decisions. Figure 3-2 shows examples of larger railway networks including strong coupling.

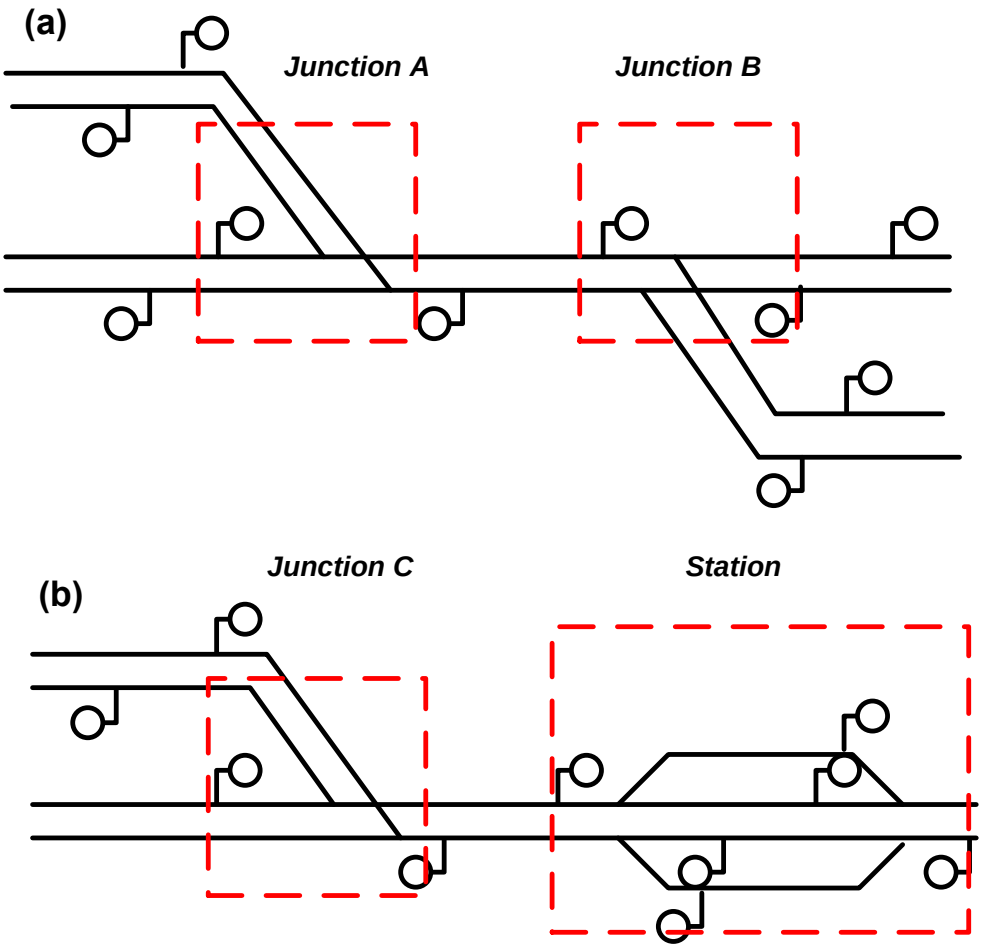


Figure 3-2 Network examples including strong coupling

In Figure 3-2, example (a) is a strongly coupled railway bottleneck section. In the bottleneck section, junction A and junction B are both flat junctions, there are no looping area between the two junctions to change the passing sequence. When facing a train rescheduling problem, the two junctions have to ensure their local solutions are feasible globally. In detail, when junction A makes its local dispatching decisions, it has to consider the dispatching decision made by junction B. Because the trajectories of trains from the right hand side will converge at junction B and pass the shared infrastructure to junction A, making a dispatching decision at junction A has to wait until junction B completes its search of dispatching decisions in its local area. Similarly, when junction B wants to make a dispatching decision for train rescheduling, it also needs to wait for

decisions from junction A. So, in a railway bottleneck section, the decision-making of two junctions includes strong coupling, and it is not possible to make a local dispatching decision without considering dispatching decisions from the other areas. To generate a train rescheduling decision for a bottleneck section, the passing sequence of trains at both junctions should be considered together, or an appropriate approach should be proposed to tackle the strong coupling between the two junctions.

In example (b) in Figure 3-2, a junction and a station are placed on each side of the example, and trains are supposed to stop in the station with dwelling and connections. So when train rescheduling is desired in this network, the passing sequence and passing time of trains at junction C and arrival and departure times of trains at the station are taken into account. Moreover, the dispatching decisions for junction C can potentially affect the dispatching decisions of the station if a rescheduled train has future connections in the station. For example, if a train is delayed at junction C and has a connection with another train at the station, the train waiting for connection at the station has to dwell longer than planned due to the delay of the train dispatched from junction C. That train which has extended is dispatched later than in the planned timetable to junction C, and its delay will also impact the dispatching decision at junction C. So, around a station whose services have connections with each other, there is strong coupling with its surrounding junctions, and the dispatching decisions of the junction and station cannot be generated independently. In practice, the junction area has to make its decisions with respect to the station area because the delayed departure time and poor connection can degrade the passenger journey satisfaction. However, that usually involves a loss of optimality in the entire network.

The above examples show the strong coupling of train timetables in two specific networks. For a large-scale railway network, the coupling of train timetables becomes more complex and is hard to tackle with local knowledge of a junction or station area.

3.2 MILP formulation for train rescheduling problems

In this thesis, MILP is applied to formulate a train rescheduling problem. The MILP model will be optimised with a designed objective function with respect to a series of constraints. Based on previous research by Chen et al. (2015), the MILP formulation is improved with binary variables which state train passing sequence in single junction and train passing sequence in coupled junctions. Moreover, the coupling variables are also used for collaborative process (Condorcet voting) to trade-off conflicts with binary sequence variables. In this section, the MILP formulation for train rescheduling problems in a large-scale railway network will be discussed and illustrated in detail.

3.2.1 Objective function

The MILP is designed to find an optimised solution for the problem considered. To evaluate the performance of a solution in terms of problem considered, an 'objective' is selected to achieve the control purpose of the MILP. In general, the performance of a solution in the context of an optimisation objective is transferred into a value based on an objective function. For a railway traffic management problem, the objective can be set as minimising train delays (Corman et al., 2012; Ho et al., 1997), minimising passenger waiting (Weston et al., 2006), minimising energy consumption (Bocharnikov et al., 2007) and minimising delay penalty (Fan et al., 2012). Because of the complexity of railway operation, the purpose of traffic control for each local area and large network may be different due to different delay conditions across the whole network. In this thesis, the purpose of railway traffic management is set as minimising any unplanned stops and

delays in the whole network, and letting the train pass the controlled area as soon as possible. The objective function is set as the Total Delay (TD) of the trains considered in every junction in the network. The proposed objective function is stated in the equation below:

$$TD = \sum_{J_m \in \mathbb{J}} \sum_{\tau_m \in \mathbb{T}} \left| t_{\tau_m}^{J_m} - t_{\tau_m}^{J_m{}^{\mathcal{O}}} \right| \qquad (3-1)$$

In the context of this thesis, J_m stands for a junction in the controlled area, $\mathbb{J}$ is the set of junctions in the controlled area, and τ_m is a considered train in the set of trains $\mathbb{T}$. The variable $t_{\tau_m}^{J_m{}^{\mathcal{O}}}$ stands for the original passing time of train τ_m at junction J_m in its timetable, and the variable $t_{\tau_m}^{J_m}$ is the rescheduled passing time of train τ_m at junction J_m.

$$J_m, J_n \in \mathbb{J}, \qquad m, n \in \{1,2, \dots, \|\mathbb{J}\|\} \qquad (3-2)$$

$$\tau_m, \tau_n \in \mathbb{T}, \qquad m, n \in \{1,2, \dots, \|\mathbb{T}\|\} \qquad (3-3)$$

The subscript *m* or *n* stands for an arbitrary individual in its set. $\|\mathbb{J}\|$ and $\|\mathbb{T}\|$ stand for the total number of junctions and trains, respectively.

To make a new train dispatching decision, the variable $t_{\tau_m}^{J_m}$ is optimised, which minimises the TD of trains in the whole network with the following constraints in railway circumstance.

3.2.2 Train operational constraints

Operational constraints are mainly organised by a railway kinematic model. Based on the railway kinematic model, the speed limit of the line, occupation of the track circuit and the train operation mode will be considered, to calculate the travelling time of a train from one node to anther node in the railway network.

As a train τ_m enters the control area of the railway traffic management system, the system will get the entry time of the train at its control boundary, and the passing time of the train at junction J_m should not be smaller than the entrance time plus minimum travelling time between the control boundary and junction area. The operational constraint between the control boundary and junction area is shown below.

$$t_{\tau_m}^{J_m} \geq t_{\tau_m}^{b} + t_{\tau_m}^{b,J_m}, J_m \in \mathbb{J}, \tau_m \in \mathbb{T} \qquad (3-4)$$

where $t_{\tau_m}^{b}$ stands for the entrance time of train τ_m at the control boundary, and $t_{\tau_m}^{b,J_m}$ states the minimum travelling time between the control boundary and the considered junction J_m.

In the control area of the traffic management system, the minimum travelling time for train τ_m between two neighbouring junctions should be guaranteed.

$$t_{\tau_m}^{J_n} \geq t_{\tau_m}^{J_m} + t_{\tau_m}^{J_m,J_n}, J_m, J_n \in \mathbb{J}, \tau_m \in \mathbb{T} \qquad (3-5)$$

where $t_{\tau_m}^{J_m}$ and $t_{\tau_m}^{J_n}$ stand for the rescheduled passing times of train τ_m at junction J_m and junction J_n, respectively, and $t_{\tau_m}^{J_m,J_n}$ is the minimum travelling time for train τ_m between junction J_m and junction J_n.

3.2.3 Train sequence constraints

As trains enter the control area, their incoming sequence is detected by the railway traffic management system; for junctions with no looping area between the control boundary and junction, an indicator $k_{\tau_m,\tau_n}^{J_m}$, which indicates the passing sequence of trains, is introduced.

$$k_{\tau_m,\tau_n}^{J_m} = \begin{cases} 1, & \text{train } \tau_m \text{ is dispatched before train } \tau_n \\ 0, & \text{otherwise} \end{cases} \qquad (3-6)$$

As shown in equation (3-6), the value of $k_{\tau_m,\tau_n}^{J_m}$ indicates the passing sequence of train τ_m and train τ_n at junction J_m.

In the junction area, to ensure the safety of train operations, a headway should be inserted between two neighbouring trains. The chosen headway value is a function of train types and their movement status.

$$H_{\tau_m,\tau_n}^{J_m} = f_{headway}(\tau_m, \tau_n, J_m), \tau_m, \tau_n \in \mathbb{T}, J_m \in \mathbb{J} \qquad (3-7)$$

where $H_{\tau_m,\tau_n}^{J_m}$ is the headway between train τ_m and train τ_n at junction J_m. The train sequence constraint is shown below:

$$t_{\tau_n}^{J_m} \geq t_{\tau_m}^{J_m} + H_{\tau_m,\tau_n}^{J_m} * k_{\tau_m,\tau_n}^{J_m}, \tau_m, \tau_n \in \mathbb{T} \qquad (3-8)$$

In equation (3-8), the passing sequence of train τ_m and train τ_n at junction J_m is satisfied, and the entry sequence of train τ_m and train τ_n is kept as the same sequence of passing junction J_m.

3.2.4 Train conflict constraints

In a railway junction area, trains from different routes may have a conflict when passing the junction due to the overlap in occupation of a track circuit in different routing actions at the junction. To avoid train conflicts in a junction area, the routing actions have to be taken in an ordered way, and headways should be inserted between the passing times of conflict trains in the junction area.

$$r_m^{J_m}, r_n^{J_m} \in \mathbb{R}^{J_m}, J_m \in \mathbb{J}, m, n \in \{1,2,\dots,\|\mathbb{R}^{J_m}\|\} \qquad (3-9)$$

$$\mathbb{R}^{J_m} \in \mathbb{R} \qquad (3-10)$$

where $r_m^{J_m}, r_n^{J_m}$ stands for two of the routings in junction J_m, and $\mathbb{R}^{J_m}$ is the set of all the routings at junction J_m. $\|\mathbb{R}^{J_m}\|$ is the total number of routings at junction J_m, and $\mathbb{R}$ stands for the set of routings in the whole network.

A binary indicator is proposed to identify the conflicts between the routing actions in the junction area.

$$r_m = f_{routing}(\tau_m, J_m), \tau_m \in \mathbb{T}, J_m \in \mathbb{J}, r_m \in \mathbb{R}^{J_m} \quad (3-11)$$

$$k_{r_m,r_n}^{J_m} = \begin{cases} 1, & \text{if route } r_m \text{ has conflict with route } r_n \text{ at jucntion } J_m \\ 0, & \text{otherwise} \end{cases} \quad (3-12)$$

where $f_{routing}(\tau_m, J_m)$ stands for train τ_m approaching J_m and asking for its corresponding routing r_m to pass the junction, and the proposed indicator $k_{r_m,r_n}^{J_m}$ stands for the conflict status of two routing actions, r_m and r_n, at junction J_m. The headway between the conflict trains at junction J_m is ensured by equation (3-13).

$$t_{\tau_n}^{J_m} \geq t_{\tau_m}^{J_m} + H_{\tau_m,\tau_n}^{J_m} * k_{\tau_m,\tau_n}^{J_m} * k_{r_m,r_n}^{J_m}, \tau_m, \tau_n \in \mathbb{T} \quad (3-13)$$

In equation (3-13), train τ_m is dispatched before train τ_n at junction J_m from conflict routes, and the headway between train τ_m and train τ_n, $H_{\tau_m,\tau_n}^{J_m}$, is inserted in this equation.

3.2.5 Train dwelling and connection constraints

Within the journey of a train, station dwellings and connection constraints must also be satisfied in the optimisation. In this thesis, a station is stated as S_m and a set of stations is stated as $\mathbb{S}$.

$$S_m \in \mathbb{S}, \qquad m \in \{1,2,\dots,\|\mathbb{S}\|\} \quad (3-14)$$

where $\|\mathbb{S}\|$ stands for the total number of stations in the network.

In general, the dwelling of a train in a station is achieved by its planned arrival time, departure time and minimum dwelling time if the train has no connections in the station. Usually, the time slot between planned arrival time and departure time is a little longer than the minimum dwelling time, and is designed to absorb minor train delays. However, if the train delay exceeds the designed robustness tolerance, the train will depart the station after the minimum dwelling time is satisfied. The train dwelling constraint is illustrated below.

$$k_{\tau_m}^{S_m} = \begin{cases} 1, & \text{the train } \tau_m \text{ has dwelling at station } S_m \\ 0, & \text{otherwise} \end{cases} \quad (3-15)$$

$$t_{\tau_m}^{S_m,dep} = \max\left(t_{\tau_m}^{S_m,arr} + D_{\tau_m}^{S_m} * k_{\tau_m}^{S_m},\ t_{\tau_m}^{S_m,dep^{\mathcal{O}}}\right), S_m \in \mathbb{S}, \tau_m \in \mathbb{T} \quad (3-16)$$

where $k_{\tau_m}^{S_m}$ is an indicator which identifies whether train τ_m has a dwelling at station S_m. The variables $t_{\tau_m}^{S_m,dep}$ and $t_{\tau_m}^{S_m,arr}$ stand for the rescheduled departure time and rescheduled arrival time of train τ_m at station S_m, respectively. $t_{\tau_m}^{S_m,dep^{\mathcal{O}}}$ is the departure time of train τ_m at station S_m in its original timetable, and $D_{\tau_m}^{S_m}$ is the minimum dwelling time of train τ_m at station S_m.

If a train has a connection with another train in the station area, its departure time has to satisfy the minimum connection time between the two connected trains, which guarantees passengers can catch up their following journey. So, for connected services, the train departure time is set as the larger value between planned departure time and the arrival time of the connected train plus the minimum connection time.

$$k_{\tau_m,\tau_n}^{S_m} = \begin{cases} 1, & \text{train } \tau_m \text{ and } \tau_n \text{ have connection at station } S_m \\ 0, & \text{otherwise} \end{cases} \quad (3-17)$$

$$t_{\tau_m}^{S_m,dep} = \max\left(t_{\tau_n}^{S_m,arr} + C_{\tau_m,\tau_n}^{S_m} * k_{\tau_m,\tau_n}^{S_m},\ t_{\tau_m}^{S_m,dep^{\mathcal{O}}}\right), S_m \in \mathbb{S}, \tau_m \in \mathbb{T} \quad (3-18)$$

where $k_{\tau_m,\tau_n}^{S_m}$ is an indicator which illustrates the connection of services between train τ_m and train τ_n at station S_m, and passengers are expected to transfer from train τ_n to train τ_m. $C_{\tau_m,\tau_n}^{S_m}$ states the minimum connection time between train τ_m and train τ_n.

From equations (3-14) to (3-18), constraints in station areas are defined. Equations (3-15) and (3-16) state the train dwelling constraints, which should be satisfied if a train has planned dwelling at a station, and equations (3-17) and (3-18) are connection constraints for services with connections at a station. Services which have connections at a station should satisfy constraints (3-15) to (3-18) at the same time.

3.2.6 Junction coupling constraints

Due to coupling of train timetables in a network, the dispatching decision of a junction may affect the dispatching decision of other junctions. Here, a binary coupling variable is proposed to indicate the coupling between two junctions.

$$k_{\tau_m,\tau_n}^{J_m,J_n} = \begin{cases} 1, & \text{the passing sequence of train } \tau_m \text{ and } \tau_n \\ & \text{is coupled beetween junction } J_m \text{ and } J_n \\ 0, & \text{otherwise} \end{cases} \qquad (3-19)$$

The strong coupling between two junction areas is identified based on the layout of their network, so the value of $k_{\tau_m,\tau_n}^{J_m,J_n}$ is decided by the corresponding route of train τ_m and train τ_n at junction J_m and junction J_n, respectively. The strong coupling between junction J_m and junction J_n is stated in equation (3-20).

$$k_{\tau_m,\tau_n}^{J_m} = k_{\tau_m,\tau_n}^{J_n} * k_{\tau_m,\tau_n}^{J_m,J_n}, \text{IFF } k_{\tau_m,\tau_n}^{J_m,J_n} = 1, \tau_m, \tau_n \in \mathbb{T}, J_m, J_n \in \mathbb{J} \qquad (3-20)$$

In the equation (3-20), the coupling relationship is illustrated if and only if the indicator $k_{\tau_m,\tau_n}^{J_m,J_n}$ equals 1, and the train sequence indicators $k_{\tau_m,\tau_n}^{J_m}$ and $k_{\tau_m,\tau_n}^{J_n}$ should be kept as the same value.

With the constraints considered above, the objective function can be minimised under MILP modelling as shown below:

Minimise: $$TD = \sum_{J_m \in \mathbb{J}} \sum_{\tau_m \in \mathbb{T}} \left| t_{\tau_m}^{J_m} - t_{\tau_m}^{J_m{}^{\mathcal{O}}} \right|$$

Subject to: $J_m, J_n \in \mathbb{J}$

$\tau_m, \tau_n \in \mathbb{T}$

$r_m^{J_m}, r_n^{J_m} \in \mathbb{R}^{J_m}$,

$\mathbb{R}^{J_m} \in \mathbb{R}$

$S_m \in \mathbb{S}$

$k_{\tau_m,\tau_n}^{J_m}, k_{r_m,r_n}^{J_m}, k_{\tau_m}^{S_m}, k_{\tau_m,\tau_n}^{S_m}, k_{\tau_m,\tau_n}^{J_m,J_n} \in \{0,1\}$

$H_{\tau_m,\tau_n}^{J_m} = f_{headway}(\tau_m, \tau_n, J_m), \tau_m, \tau_n \in \mathbb{T}, J_m \in \mathbb{J}$

$r_m = f_{routing}(\tau_m, J_m), \tau_m \in \mathbb{T}, J_m \in \mathbb{J}, r_m \in \mathbb{R}^{J_m}$

$$t_{\tau_m}^{S_m,dep} = \max\left(t_{\tau_m}^{S_m,arr} + D_{\tau_m}^{S_m} * k_{\tau_m}^{S_m},\ t_{\tau_m}^{S_m,dep^{\mathcal{O}}}\right), S_m \in \mathbb{S}, \tau_m \in \mathbb{T}$$

$$t_{\tau_m}^{S_m,dep} = \max\left(t_{\tau_n}^{S_m,arr} + C_{\tau_m,\tau_n}^{S_m} * k_{\tau_m,\tau_n}^{S_m},\ t_{\tau_m}^{S_m,dep^{\mathcal{O}}}\right), S_m \in \mathbb{S}, \tau_m \in \mathbb{T}$$

$$k_{\tau_m,\tau_n}^{J_m} = k_{\tau_m,\tau_n}^{J_n} * k_{\tau_m,\tau_n}^{J_m,J_n}, \text{IFF } k_{\tau_m,\tau_n}^{J_m,J_n} = 1, \tau_m, \tau_n \in \mathbb{T}, J_m, J_n \in \mathbb{J}$$

$$t_{\tau_m}^{J_m} \geq t_{\tau_m}^{J_m{}^{\mathcal{O}}}, J_m \in \mathbb{J}, \tau_m \in \mathbb{T}$$

$$t_{\tau_m}^{J_m} \geq t_{\tau_m}^{b} + t_{\tau_m}^{b,J_m}, J_m \in \mathbb{J}, \tau_m \in \mathbb{T}$$

$$t_{\tau_m}^{J_n} \geq t_{\tau_m}^{J_m} + t_{\tau_m}^{J_m,J_n}, J_m, J_n \in \mathbb{J}, \tau_m \in \mathbb{T}$$

$$t_{\tau_n}^{J_m} \geq t_{\tau_m}^{J_m} + H_{\tau_m,\tau_n}^{J_m} * k_{\tau_m,\tau_n}^{J_m}, \tau_m, \tau_n \in \mathbb{T}$$

$$t_{\tau_n}^{J_m} \geq t_{\tau_m}^{J_m} + H_{\tau_m,\tau_n}^{J_m} * k_{\tau_m,\tau_n}^{J_m} * k_{r_m,r_n}^{J_m}, \tau_m, \tau_n \in \mathbb{T}$$

With the formulation above, the MILP will minimise the TD of the considered trains in the control area with adjustment of variable $t_{\tau_m}^{J_m}$ and binary variable $k_{\tau_m,\tau_n}^{J_m}$. The variables ${t_{\tau_m}^{J_m}}^{\mathcal{O}}$, ${t_{\tau_m}^{S_m,dep}}^{\mathcal{O}}$, $D_{\tau_m}^{S_m}$ and $C_{\tau_m,\tau_n}^{S_m}$ are obtained from the train working timetable which is provided in advance by rail operators, and the indicators $k_{\tau_m}^{S_m}$ and $k_{\tau_m,\tau_n}^{S_m}$ are identified from the planned journey of each considered train in every station in the control area. The coupling indicator $k_{\tau_m,\tau_n}^{J_m,J_n}$ is an indicator mainly identified from the layout of each junction and decided by the routes asked for by approaching trains τ_m and τ_n whose value can be provided online by each junction. The variable $H_{\tau_m,\tau_n}^{J_m}$ and the generation of corresponding routing r_m is decided by each junction with respect to its database and the approaching train. The passing time of train τ_m at the control boundary $t_{\tau_m}^{b}$ is monitored by the railway traffic management system when a train crosses its control boundary, and all the travelling times for considered train τ_m between nodes in the network, $t_{\tau_m}^{b,J_m}$ and $t_{\tau_m}^{J_m,J_n}$, can be supported by the train kinematic model by the data base of the traffic management system, the approach to predict train passing times at junction areas is discussed in Appendix B.

The proposed MILP can be divided into two parts, optimisation of the train passing sequence which is stated by the binary values of indicator $k_{\tau_m,\tau_n}^{J_m}$, and optimisation of the passing time at each junction area of considered trains which is decided by continuous variable $t_{\tau_m}^{J_m}$.

3.3 Overview of MAS application to train rescheduling problems

In the sections above, the MILP formulation for the train rescheduling problem is illustrated, and the coupling of train timetables and their passing sequence at each junction

area is emphasised. However, as discussed in Chapter 2, it is hard to tackle the train rescheduling problem in a large-scale railway network because the computing of optimised solution will be time-consuming and the final result will be of poor quality. As a result, to tackle a large-scale railway traffic management problem, an MAS-based approach is proposed in this thesis to solve the train rescheduling problem in a decentralised way. An MAS is a group of agents which are embedded in a common environment and interact with each other to achieve their shared common goal. The 'agent' is an abstract concept which stands for an intelligent entity with some characters and functions such as: autonomy, activity, reactivity, mobility and social ability (Weiss, 1999). Due to the nature of an MAS, it is designed with a decentralised structure, and agents may play the same or different roles in the entire system. And the most important features of MAS for train rescheduling problem are reactivity and social ability. The reactivity is objective-directive behaviour of an agent, and it is common that a system is designed with one or several objectives. In train rescheduling problems, the goal of an agent can be defined as one of objective functions in section 3.1 and an agent developed for optimisation with respect to an objective function is also easy to be built. However, social ability in an MAS for train rescheduling problems is still absence. The social ability usually covers the protocol of the social behaviour, data structure and the frequency of the exchanged data. So the social behaviour can be diverse for different MASs with different purposes. From the definition of an agent in an MAS, it is important to analyse and discuss the functions of the agents and their common environment when developing an MAS. So, to tackle a large-scale train rescheduling problem, the huge and complex rescheduling problem can be decomposed into local problems and solved by local agents. To achieve an MAS for a large-scale train rescheduling problem, the MAS proposed in this thesis achieves the optimising behaviour with the searching algorithm, and the social behaviour

within the proposed MAS is defined as collaborative approaches in the following chapters. More applications of MAS for transportation problems are illustrated in section 5.1. And the circumstances of railway operation and main properties of a junction dispatcher are analysed as follows.

3.3.1 Circumstances of railway operation

The key problem facing an agent is that of deciding what actions it should perform in order to best satisfy its delegated objectives. To better achieve the actions of the agent in an environment, the properties of the environment should be analysed. Russell and Norvig (2016) recommended the following classification of environment properties to look into a shared MAS environment.

1) Accessible vs inaccessible

An accessible environment is one which is able to provide complete, accurate, up-to-date information about the status of the environment. Otherwise, the environment is inaccessible.

2) Deterministic vs non-deterministic

An environment is called deterministic when a single response is guaranteed for any single action. Namely, there is no uncertainty about the state of the environment resulting from an action. However, in the physical world, there are few deterministic environments which presents great challenges for agent designers.

3) Discrete vs continuous

An environment is discrete if it consists of a fixed, finite number of actions and percepts in it, and a continuous environment is consists of an infinite number of continuous actions

and percepts. Russell and Norvig (2016) stated that a chess game is a typical discrete environment, and taxi driving is an example of a continuous environment.

4) Static vs dynamic

A static environment is one that is unchangeable except for modifications due to actions taken by the considered agent. A dynamic environment is one that has other processes operating on it, and the environment can be changed by any actions not restricted to those of the agent considered.

5) Episodic vs non-episodic

An environment is episodic if the performance of agent actions can be viewed as a number of discrete episodes, and there is no link between to performance of neighbouring agents' actions. In contrast, in a non-episodic environment, different agent action scenarios interact.

To develop an MAS for a train rescheduling problem, the circumstance of train operation is the environment in the jargon of MAS, and the main objective of the proposed MAS for a train rescheduling problem is to minimise the effect of primary delay and avoid generating knock-on delays. However, the reasons which lead to primary delays are not predictable, so the MAS environment for the train rescheduling problem is dynamic and continuous.

As for the MILP formulation in the section above, the information desired for real-time train rescheduling is the train movement, including moving speed and direction, and its passing times at specific nodes in the network. Because the information can only be provided by the track circuit and signalling system in the physical layer of the railway traffic management system, the information can be accessed when a train passes the

boundary of a track circuit and a signal, so the information is inherently discrete. As a result, the MAS environment for the train rescheduling problem is inaccessible.

The MAS actions for the train rescheduling problem is their adjustment solution for the passing sequence and passing time of considered trains in the junction area. As illustrated for MILP formulation, a railway network is a strongly coupled network. Modification of the dispatching solution at a junction area can lead to a domino effect across the whole railway network. Due to the decision-making process having as lack of scope for the entire network, the optimised solution generated for a local dispatching area has better performance than the original timetable of the considered train. However, the adjustment may degrade the performance of dispatching areas outside the controlled area. So, the decision-making for the train rescheduling problem generates retiming and resequencing solutions for dispatching areas, and the currently considered train rescheduling problem can be affected by former solutions made by its local area and by its neighbouring area. Therefore, the MAS environment for the train rescheduling problem is non-deterministic and non-episodic.

So, the circumstance of railway operation is inaccessible, non-deterministic, non-episodic, dynamic and continuous. To embed an MAS into the train operation environment, an agent has to consider the train rescheduling problem in the current stage of its local area and the impacts from the last control stage and neighbouring control area.

3.3.2 Train dispatching agent in a railway junction area

To analyse the environment properties of train operation in a railway network, the agent entity in an MAS for train rescheduling should consider the following functions to achieve its objective:

1) The agents are easy to embed into the current railway traffic management system: The main reason to develop an MAS for a train rescheduling problem is its decentralised structure which is inherently similar to our existing railway traffic management system.
2) The agents are able to access the information desired to generate train rescheduling solutions: due to the isolation of each area of a railway network, an agent may not be able to get up-to-date information from every area in the entire railway network. However, an agent should have the capability to get up-to-date information for its local control area which is used to generate real-time train rescheduling solutions.
3) The agents are capable of generating a real-time solution for the considered problem: the most important facet of real-time decision-making is that the computational time should be small enough for real-time control. So, for an MILP problem, it is reasonable to choose a control area of proper size which ensures the searching space faced by each agent matches its searching capability.

From the functions illustrated above and based on the agent definition used by Weiss (1999), the following definition of a junction agent for the train rescheduling problem for a large-scale railway network in the context of MAS is proposed:

Definition 3-1: A *junction agent* or *junction dispatching agent* is defined as a computational entity which collects information from its local network and provides real-time dispatching solutions to the trains in its local area.

3.4 Decomposition approach for a large-scale railway network

As discussed in the section above, to achieve an MAS for a large-scale train rescheduling problem, an appropriate approach for dividing the railway network should be proposed which ensures the agent can access the desired information in its local area, and that its

local solution searching approach is capable of generating optimised solutions within the designed time limit. In this section, the decomposition approach for dividing the railway network into simple junction levels is proposed. To better explain the decomposition approach, the following figures show the proposal in detail.

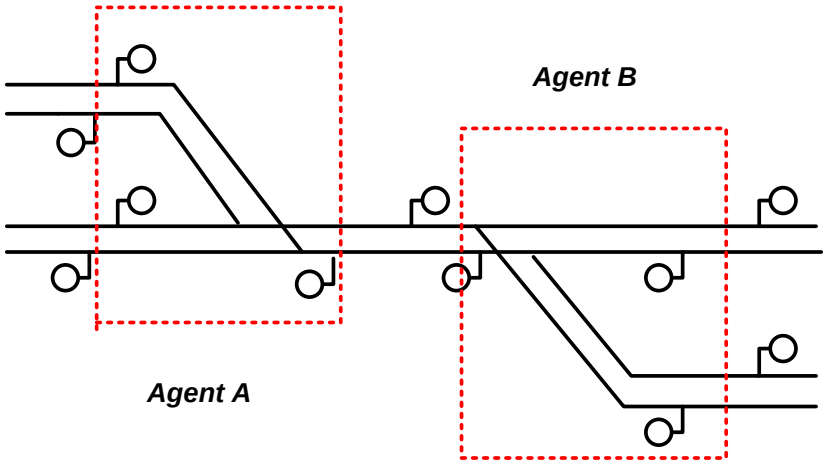


Figure 3-3 Example of decomposition approach to railway bottleneck

In Figure 3-3, the example shows the application of the proposed decomposition approach to a railway bottleneck section. The bottleneck section consists of two flat junctions, so it is divided into two junctions, and for each junction a junction agent is applied for its local dispatching problem. The control boundary of each junction is defined and extends from the first signal before the conflict area to the first after the conflict area. The infrastructure between the two junctions is double track, which ensures the passing sequence of trains from/to each junction area is the same as the passing sequence of trains to/from the other junction. So, the train rescheduling problem is degraded into train rescheduling problems in the two junction agents which are coupled with each other, and the dispatching solution can be generated by a local searching approach in each junction agent, with collaborative strategies between the two agents.

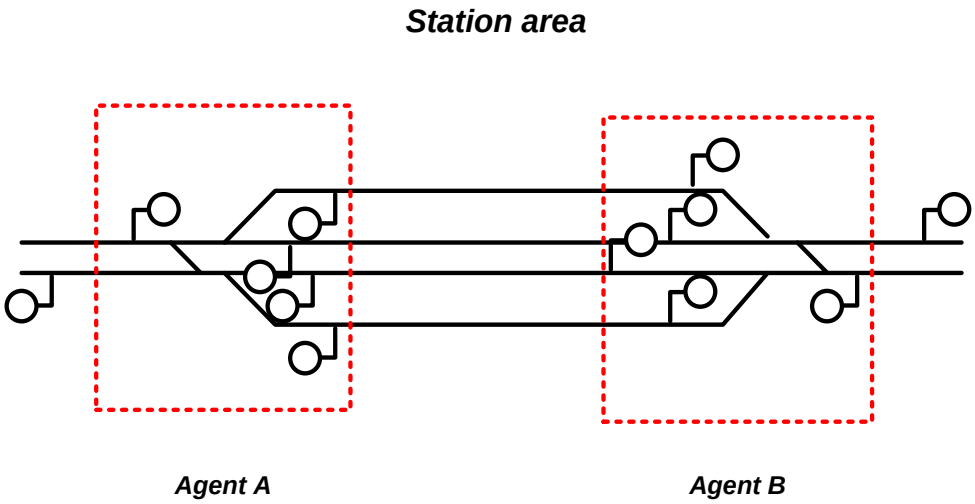


Figure 3-4 Example of decomposition approach to railway station area

In Figure 3-4, an example is shown to explain the decomposition approach for a railway station area. The area where the trains trajectories converge and diverge to the station area can also be viewed as a junction area. In the proposed decomposition approach, the train rescheduling problem in station areas can be organised by the train rescheduling problems in the junction areas surrounding the station. So, the rescheduling decision made by junction agents including reordering and retiming actions is the rescheduled arrival and departure time for the considered train in the station area.

3.5 Summary

In this chapter, the train rescheduling problem in a large-scale railway network is discussed, and the author proposes an MILP formulation for the train rescheduling problem in a large-scale railway network. Furthermore, to develop an MAS to tackle a real-time train rescheduling problem in a large-scale network, the circumstances of train operation are analysed and discussed, and the definition of a junction agent in the context of railway traffic management is proposed along with the main functions to be achieved. Finally, to reduce the complexity of the train rescheduling problem and to satisfy the time

limit of real-time traffic control, a decomposition approach is proposed which divides the railway network into simple junction level.

In the section on problem description, the strong coupling of train timetables in railway junction and station areas is emphasised; that strong coupling leads to a series of challenges on making train dispatching solutions only with the information in a local area. Moreover, in the proposed MILP formulation, the train rescheduling problem is stated as two parts, optimising the passing sequence of the considered train in every junction area and the corresponding passing times of considered trains in the junction areas. To state the strong coupling of train timetables in the train rescheduling problem, a coupling indicator is proposed in the MILP formulation to indicate the coupling between two junction areas, which also addresses the difficulties in decision-making for a large-scale network.

To develop an MAS for real-time train rescheduling resolution, the main properties of railway operation – inaccessible, non-deterministic, non-episodic, dynamic and continuous – were discussed. The main functions of a junction agent are also discussed in this section, and to reduce the searching complexity, a network decomposition approach is developed. In the decomposition approach, the basic unit of the train rescheduling problem is set as a simple junction area; namely, a railway network can be divided into several junction units, and the entire train rescheduling problem is supposed to be solved in a decentralised way.

The key innovations in this chapter are the MILP modelling for a large-scale railway network with consideration of the strong coupling of train timetables, and the decomposition approach to divide the large-scale network. The main benefits of decomposing the network into simple junction level are the smaller searching space in

each area, which satisfies the time limit for real-time traffic control and due to the layouts of every simple junction inherently being the same, so there is no need to develop a specific searching algorithm for each local area. The detailed approach to solving the local train rescheduling problem and the strategies for agent collaboration are not covered; they will be illustrated in the following chapters.

Chapter 4

TWO-AGENT SYSTEM FOR TRAIN RESCHEDULING PROBLEMS IN A RAILWAY BOTTLENECK SECTION

In this chapter, a two-agent system (P2P system) is proposed for train rescheduling problems in railway bottlenecks. GA is chosen as the searching algorithm in the P2P system, and a negotiation strategy is proposed to trade off the optimised solutions generated by each agent in the P2P system. To illustrate the benefits of a P2P system for solving train rescheduling problems in railway bottlenecks, the P2P approach is compared with a centralised approach which uses the same searching algorithm as the P2P system. A British railway bottleneck section is applied to test the P2P and centralised approaches, and the computational experiments show that the P2P system can generate real-time dispatching solutions for the train rescheduling problem considered with better computational time and optimality than the centralised system. The key improvements of applying P2P system are its decentralised structure and decentralised optimisation on solving the train rescheduling problems, which degrades computational complexity of the whole problems and allocates sub-problems to every single agent.

4.1 Overview of P2P system

P2P systems have emerged as promising distributed systems developed for large-scale computing and networking problems since the mid-2000s; Buford and Yu (2010) defined a P2P system as a distributed structured network whose participants can share their own resources such as: processing power, storage capacity, network link capacity, etc., and which is able to provide their actions with their collaborations to the network. In a P2P

system, all the peers have the same rights, rules and responsibilities. The following properties are characteristics found in most P2P systems:

1) Resource sharing

Every peer in a P2P system is able to access and contribute to the shared resources and able to operate the system from its own aspect.

2) Networked and decentralised

All the peers are interconnected and there is no centralised node in the system. The decentralised structure inherently avoids centralised computing with exhausted complexity.

3) Symmetry

The peers in a P2P system usually have equal roles in the system. With an appropriate collaboration approach, the P2P system is immune to single-point failure and ensures the robustness of the entire system.

4) Self-organisation and autonomy

Every peer in the system is organised with respect to its local area and achieves local actions with local knowledge.

In recent years, there has been an increasing number of applications with P2P systems in different areas. The most famous example of a well-known P2P system is the software Napster which provided a P2P music sharing service between two customers from 1999 to 2001. Napster was very popular on campuses whose about 61% of their Internet visits consisting of MP3 files. And with the Napster, most customers were able to find their requirements from other customers in the dormitory without the requirement of external network traffic. Although Napster was blocked finally due to copyright violations, the

software provided a new mode of data exchange which tackles most requests from customers in its local network without data from an external server. Moreover, a P2P model for bike sharing to relieve a shortage of bike parking capacity around train stations has been proposed by van Goeverden and Correia (2018). Their model is established based on the willingness of passengers to share bicycles, and was developed to improve bike sharing around a train station among arriving passengers, with a feeder mode whose purpose is to reduce the shortage of capacity for parked bikes. Analysis based on data published in the Dutch National Travel Survey showed that the proposed approach potentially improves the shortage of bike parking at large train stations by between 13% and 50%. A framework to analyse the reliability of P2P communication systems has been proposed by Baset and Schulzrinne (2010). The developed framework presents a model which estimates the number of relays needed to maintain the transmission of a signal with the desired reliability. The model includes a two-level hierarchical P2P network to find a relay in a certain number of hops and a distributed relay selection technique to minimise the latency of data transmission. Furthermore, a P2P technique has been applied as a novel multimedia streaming system for an urban rail environment (Poderys et al., 2018). The proposed P2P system utilises the unoccupied bandwidth of Communication Based Train Control (CBTC) for the transmission of multimedia to urban trains, delivering the media to customers' devices via P2P technology. A P2P ridesharing model which promotes the usage of private vehicles around metro stations was developed by Masoud et al. (2017) to address the last-mile and first-mile problem. The Log Angeles Metro Red Line was studied, to provide insights into pricing of a ridesharing system.

Although there are few applications of P2P systems in railway traffic management, the P2P technique fits well with the train rescheduling problem in railway bottleneck sections due to the following reasons. Each junction in a railway bottleneck section has the same

weight in the whole network, and due to the strong coupling of train timetables in rail bottlenecks, both of the junctions should consider the decision made by another junction in the bottleneck area. Furthermore, the computational complexity of train rescheduling problems in a bottleneck section is larger than in a normal railway network, so a P2P system becomes a promising approach in rail bottlenecks to solve the train rescheduling problem in a distributed way. With the property of decentralised structure, the P2P system is able to decompose the complex train rescheduling problem to single junction level, and thus, reduce the computational complexity in the network; and with the feature of resource sharing between two peers, each peer can find its optimised solution with no deterioration against centralised approach. And to tackle the coupled train rescheduling problem in railway bottlenecks, P2P concept with data exchange strategies is a promising approach to support the decentralised decision-making process.

4.2 System structure

A bottleneck section in a railway network generally consists of two level junctions (flat junctions) at each end of an intensive infrastructure, and trains from different origins converge their trajectories onto a single line at one of the junctions and head in different directions in the other junction area. As a result, poor operation in train dispatching may decrease the capacity of the bottleneck area and prolong the knock-on delay when disturbance happens, so real-time railway traffic management in the bottleneck area becomes a critical problem.

In a P2P system on a railway bottleneck section, the section can be decomposed to single junction level, and each junction is modelled as a dispatching agent. Figure 4-1 illustrates the split of a bottleneck section. In each junction area, the junction agent focuses on its local control area and gets train movement information from the signalling and

interlocking system; dispatching solutions are generated with its local knowledge of the network layout and its local objective function. Finally, its local solution will be transferred into future signalling and interlocking actions. To ensure the feasibility of the local solution, strategies for data exchange are developed.

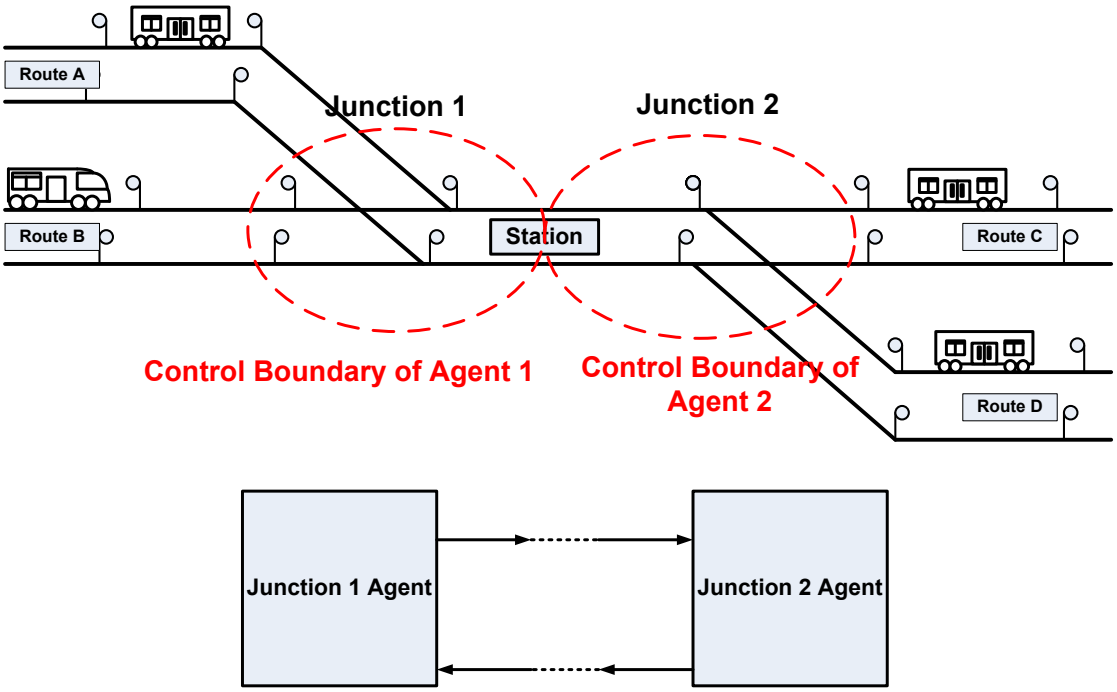


Figure 4-1 P2P structure of a bottleneck in a railway network

4.3 Algorithm development

In a control step of the searching process, one of the junction agents will start its local search only with its local knowledge and then send its interface variables to the other junction agent. The other junction will start its local search based on its local knowledge and the interface variables from the other side. The final decisions are made once their local solutions are feasible globally. An example of a control step is shown in Figure 4-2:

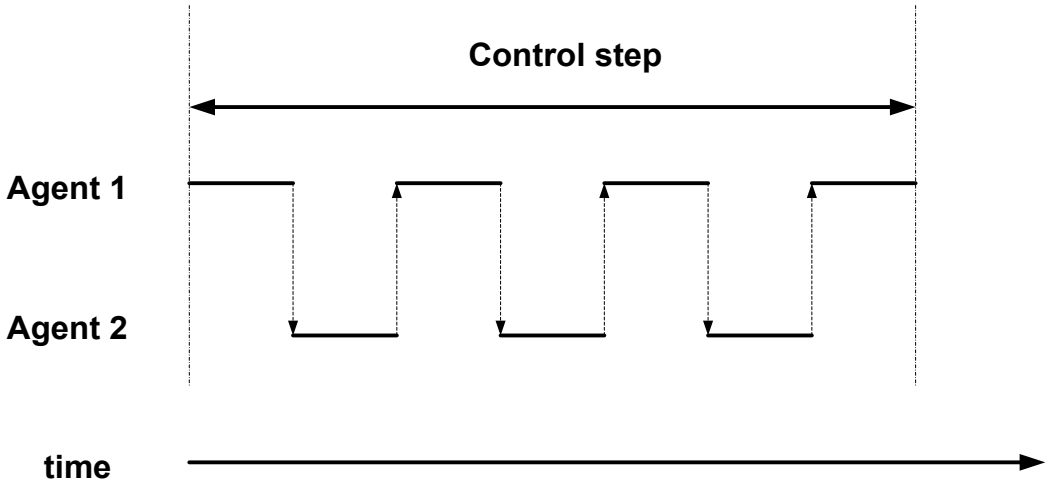


Figure 4-2 Data exchange of the two agents in a bottleneck section in a control step

Furthermore, to achieve the negotiation between two agents, the dominating relationship between trains and junction agent and the interface variables are defined as follows:

Definition 4-1: In the local area of a junction agent J_m, the trains which are dispatched from its local control area to other control areas of the junction agent are called trains *dominated* by junction J_m, stated as $\tau_m \rightarrow J_m$.

Definition 4-2: Within this paper, the *interface variables* are the passing times of the dominated trains of the best solution in the final generation of the agent's local search.

GA is a population-based algorithm for optimisation problem which can map a group of solutions in the whole searching space and improve its solutions with crossover and mutation operators by finding neighbourhoods. The group of solutions is named as a generation in a GA and the top solution in its final generation is usually chosen as the output solution. GA is simply organised by a series of operators and the operators can be replaced easily with respect to different problem properties, developers can adjust different kinds of operators and their corresponding variable settings to improve the performance of GA. The terminating approach for GA ensures its computational time is

usually short, which is a promising feature for real-time decision making. As a result, GA is usually applied to provide optimised solutions for huge size optimisation problem. To resolve real-time train rescheduling problem, GA can respond the top solution of its final generation with short execution time which satisfies the time horizon for real-time traffic control. To satisfy the constraints in railway operation, extra operators can be applied to classical GA to ensure the feasibility of its solutions. And to achieve further collaborative approach in this thesis, GA can be easily modified to other structure to assist the developed collaborative strategy. The final generation of GA can also provide sufficient diversity solutions for the adjustment of developed collaborative strategy. So, GA is chosen and designed as the searching algorithm, and modified into different versions for different purposes in this thesis.

A designed GA is developed as the local searching algorithm. The designed GA keeps the same structure with normal GA but with two extra operators, modification and classification, which are the key innovations for the searching algorithm in the thesis. The modification is a post-correction operator which ensures the individuals generated by crossover and mutation are feasible with respect to train operational constraints; moreover, applying the modification operator enables the algorithm greedier than normal GA in its searching process, which leads to better final solutions and less execution time for the designed GA. The classification operator is designed to classify the individuals in the final generation for the proposed GA. The individuals are classified into two groups with respect to the interaction variables, which are groups of individuals having same and different order with the interaction variable. The proposed classification also simplifies the negotiation process after each iteration of local searching.

Within the designed GA, an individual in each searching generation is chosen as the passing time of all the trains covered by the control areas of both agents in the control

step. Furthermore, for the train rescheduling problem in a railway bottleneck section, we introduce a weighting variable W_{τ_m} to indicate the priority of the considered train τ_m in the network. The value of weighting W_{τ_m} can differ with respect to different control purposes. For instance, a higher weighting value could be applied to emergency trains or high-speed trains, and lower values could be applied to local services. Based on the objective function proposed in last chapter, equation (3-1), the objective function in a single junction agent for a P2P system can be rewritten as Weighted Average Delay (WAD):

$$\mathrm{WAD} = \frac{\sum_{\tau_m,\tau_n \in \mathbb{T}} \left(W_{\tau_m} * \left| t_{\tau_m}^{J_m} - {t_{\tau_m}^{J_m}}^{\mathcal{O}} \right| + W_{\tau_n} * \left| t_{\tau_n}^{J_m} - {t_{\tau_n}^{J_m}}^{\mathcal{O}} \right| \right)}{\sum_{\tau_m,\tau_n \in \mathbb{T}} \left(W_{\tau_m} + W_{\tau_n} \right)}, \quad (4-1)$$
$$\tau_m, \tau_n \in \mathbb{T}, \tau_m \to J_m, \tau_n \to J_n, J_m, J_n \in \mathbb{J}$$

For optimisation of train timetables of considered trains in a junction agent, the variables $t_{\tau_m}^{J_m}$, $t_{\tau_n}^{J_m}$ will be optimised with the constraints (3-2) to (3-20). However, the variable $t_{\tau_n}^{J_m}$, the passing times of considered trains at junction J_m, is unknown due to it strongly replies on decision variable $t_{\tau_n}^{J_n}$ in junction J_n. The optimised $t_{\tau_n}^{J_n}$ are decided by the junction agent of J_n and transferred to junction J_m which will be introduced in the negotiation section.

The pseudocode and detailed operator of the designed GA are illustrated below:

Algorithm 4-1: GA for local searching in a P2P system

---

1 *Initialisation*

2 **WHILE** *(!The local terminating condition is satisfied)*

3 *Crossover;*

4 *Mutation;*

5 *Modification;*

6 *Classification + Sorting;*

7 *Selection;*

8 **END**

9 *Update the interface variable.*

4.3.1 Initialisation

The local searching algorithm starts from the initialisation operator; the operator generates N_{ini} individuals for the considered trains. In each individual, the passing times for the trains are organised into an array, and the passing times are generated randomly from $\left[t_{\tau_m}^{J_m{}^{\mathcal{O}}}, t_{\tau_m}^{J_m{}^{\mathcal{O}}} + t_{slot}\right]$; t_{slot} stands for a specific length of time slot which is acceptable in knock-on delays for train τ_m. The generation of each individual should also satisfy constraints (3-2) to (3-18), and the FCFS is also inserted into the first generation which makes sure the final solution is not worse than FCFS.

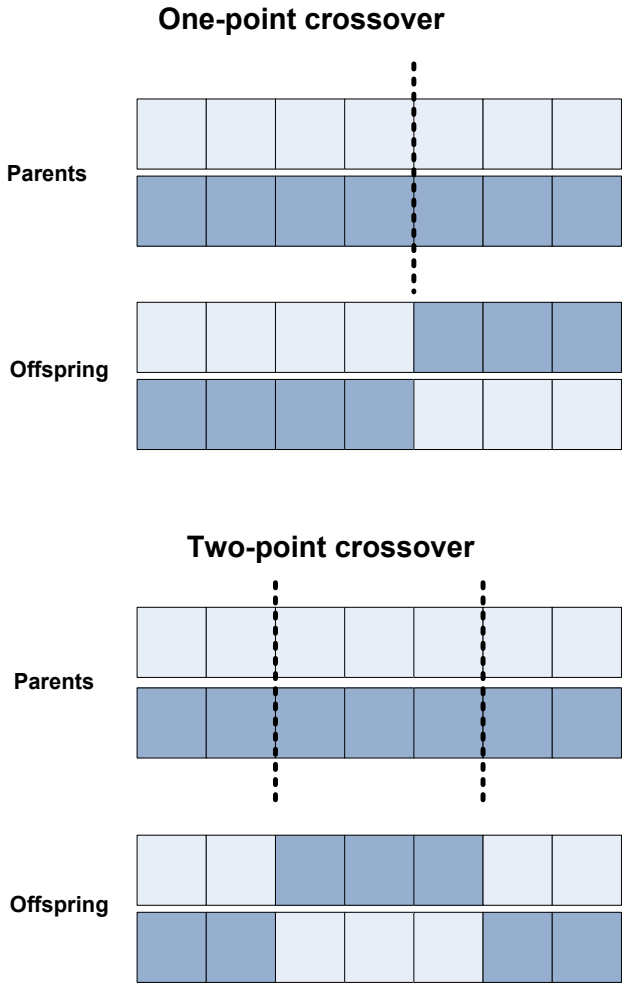


Figure 4-3 One-point and two-point crossover operators

4.3.2 Crossover

Crossover is a process to generate offspring from the two parents, so within the crossover, two of the individuals will be selected from the last generation. Then, one or two crossover points will be randomly chosen to exchange the elements in the genes of the parents. One-point and two-point crossovers are shown in Figure 4-3. The choice of crossover operators depends on the size of the gene.

4.3.3 Mutation

During mutation, a new gene will be generated based on its mutation rules, so this is an operator which may result in a new gene and let the whole algorithm jump out of the local minimal. There are three kinds of mutation rule for a true variable GA, and the choice of mutation operator usually depends on the problem size and problem features:

1) Intermediate mutation

Within intermediate mutation, two of the individuals $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ will be chosen stochastically in a generation. The new individual $\boldsymbol{X}_{mut}$ will be generated from its original value $\boldsymbol{X}_{ori}$ with the scaling parameter $\boldsymbol{F}$ which is randomly selected from (0, 1].

$$\boldsymbol{X}_{mut} = \boldsymbol{X}_{ori} + \boldsymbol{F} * (\boldsymbol{X}_i - \boldsymbol{X}_{ori}) + (\boldsymbol{1} - \boldsymbol{F}) * \left(\boldsymbol{X}_j - \boldsymbol{X}_{ori}\right) \qquad (4-2)$$

2) Line mutation

Line mutation is scaling between the best individual in the considered generation $\boldsymbol{X}_{best}$ and the considered individual $\boldsymbol{X}_{ori}$.

$$\boldsymbol{X}_{mut} = \boldsymbol{X}_{ori} + \boldsymbol{F} * (\boldsymbol{X}_{best} - \boldsymbol{X}_{ori}) \qquad (4-3)$$

3) Extended-line mutation

Extended-line mutation is also a linear scaling with the individuals $\boldsymbol{X}_{best}$ and $\boldsymbol{X}_{ori}$. However, the mutation may be extended to both sides of the line between $\boldsymbol{X}_{best}$ and $\boldsymbol{X}_{ori}$. Here, a probability variable $P_{exlin} \in (0,1]$ is proposed to indicate the placement of the new individual $\boldsymbol{X}_{mut}$.

$$\begin{cases} \boldsymbol{X}_{mut} = \boldsymbol{X}_{ori} + \boldsymbol{F} * (\boldsymbol{X}_{best} - \boldsymbol{X}_{ori}), & \text{IF } 0 < p_{exlin} \leq \frac{1}{3} \\ \boldsymbol{X}_{mut} = \boldsymbol{X}_{ori} + \boldsymbol{F} * (\boldsymbol{X}_{ori} - \boldsymbol{X}_{best}), & \text{IF } \frac{1}{3} < P_{exlin} \leq \frac{2}{3} \\ \boldsymbol{X}_{mut} = \boldsymbol{X}_{best} + \boldsymbol{F} * (\boldsymbol{X}_{best} - \boldsymbol{X}_{ori}), & \text{IF } \frac{2}{3} \leq P_{exlin} \leq 1 \end{cases} \quad (4-4)$$

4.3.4 Modification

Modification is an operator which is designed to ensure the feasibility of the local search in each junction agent. This operator is a post-correction operator and it is uniquely used in this thesis. The modification will modify the passing times of the considered trains based on its local knowledge which consists of the train operational constraints, train sequence constraints, train conflict constraints and the interface variables. Firstly, for the trains which are dominated by the junction agent itself, the former three constraints are modified.

$$\begin{cases} \text{IF } t_{\tau_m}^{Jm} < t_{\tau_m}^{b} + t_{\tau_m}^{b,Jm} \\ \text{THEN } t_{\tau_m}^{Jm} = t_{\tau_m}^{b} + t_{\tau_m}^{b,Jm} \end{cases} \quad (4-5)$$

$$\begin{cases} \text{IF } t_{\tau_n}^{Jm} < t_{\tau_m}^{Jm} + H_{\tau_m,\tau_n}^{Jm} * k_{\tau_m.\tau_n}^{Jm} \\ \text{THEN } t_{\tau_n}^{Jm} = t_{\tau_m}^{Jm} + H_{\tau_m,\tau_n}^{Jm} * k_{\tau_m.\tau_n}^{Jm} \end{cases} \quad (4-6)$$

$$\begin{cases} \text{IF } t_{\tau_n}^{Jm} < t_{\tau_m}^{Jm} + H_{\tau_m,\tau_n}^{Jm} * k_{\tau_m,\tau_n}^{Jm} * k_{r_m,r_n}^{Jm} \\ \text{THEN } t_{\tau_n}^{Jm} = t_{\tau_m}^{Jm} + H_{\tau_m,\tau_n}^{Jm} * k_{\tau_m,\tau_n}^{Jm} * k_{r_m,r_n}^{Jm} \end{cases} \quad (4-7)$$

In the modification operator above, equation (4-5) is the operational constraint for train τ_m, which ensures the minimum travelling time from the control boundary to the junction area. The sequence constraint is guaranteed by equation (4-6), where the headway between two neighbouring trains τ_m and τ_n is checked by it, and the headway between two conflict trains from different routes is ensured by (4-7). If the dwelling and connection are not satisfied for a train in a station, the following modifications should be satisfied.

$$\begin{cases} \text{IF } t_{\tau_m}^{S_m,dep} < \max\left(t_{\tau_m}^{S_m,arr} + D_{\tau_m}^{S_m} * k_{\tau_m}^{S_m}, t_{\tau_m}^{S_m,dep^{\mathcal{O}}}\right) \\ \text{THEN } t_{\tau_i}^{S_i,dep} = \max\left(t_{\tau_i}^{S_i,arr} + D_{\tau_i}^{S_i} * k_{\tau_i}^{S_i}, t_{\tau_i}^{S_i,dep^{\mathcal{O}}}\right) \end{cases} \quad (4-8)$$

$$\begin{cases} \text{IF } t_{\tau_m}^{S_m,dep} < \max\left(t_{\tau_n}^{S_m,arr} + C_{\tau_m,\tau_n}^{S_m} * k_{\tau_m,\tau_n}^{S_m}, t_{\tau_m}^{S_m,dep^{\mathcal{O}}}\right) \\ \text{THEN } t_{\tau_m}^{S_m,dep} = \max\left(t_{\tau_n}^{S_m,arr} + C_{\tau_m,\tau_n}^{S_m} * k_{\tau_m,\tau_n}^{S_m}, t_{\tau_m}^{S_m,dep^{\mathcal{O}}}\right) \end{cases} \quad (4-9)$$

Equation (4-8) ensures the minimum dwelling time of train τ_m at station S_m, and equation (4-9) ensures the minimum connection time between train τ_m and train τ_n at station S_m. For the constraints which indicate the minimum travelling time between junction J_m and junction J_n for train τ_m, it is not possible to guarantee it in its local searching, because the passing time of train τ_m at junction J_n is unknown for junction agent J_m, and constraint (3-5) can only be ensured by interface variables. As a result, the passing time of train τ_m at junction J_n is temporary replaced by the interface variables, namely $t_{\tau_m}^{J_n} \stackrel{\text{def}}{=} \tilde{t}_{\tau_m}^{J_n}$, where $\tilde{t}_{\tau_m}^{J_n}$ stands for the passing time of train τ_m at junction J_n in the interface variables. Then, the operational constraint of travelling time between two junction areas is guaranteed by the following equation.

$$\begin{cases} \text{IF } t_{\tau_m}^{J_n} < \tilde{t}_{\tau_m}^{J_m} + t_{\tau_m}^{J_m,J_n} \\ \text{THEN } t_{\tau_m}^{J_n} = \tilde{t}_{\tau_m}^{J_m} + t_{\tau_m}^{J_m,J_n} \end{cases} \quad (4-10)$$

4.3.5 Classification and sorting

Within the classification, all the individuals of a generation in the searching algorithm are divided into two groups, a group of individuals which have the same sequence as the interface variable from the other agent, and a group of individuals that have a different sequence against the interface variable. In each generation, the first group is placed at the top of the generation and the second group is placed at the bottom. The figure below shows the generation after the classification operator. Furthermore, the sorting operator arranges

the individuals in a group with ascending sequence according to their WAD values. Figure 4-4 illustrates how the classification and sorting operator structures a generation in the GA.

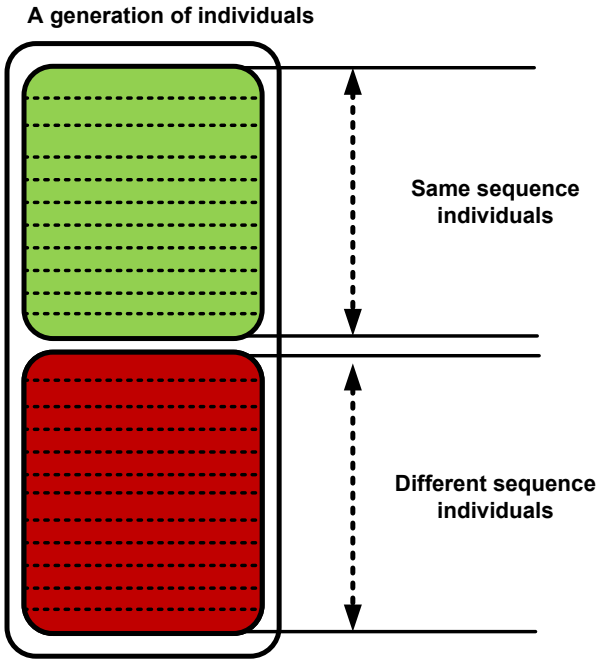


Figure 4-4 Data structure of a generation after operators of classification and sorting

4.3.6 Selection

Selection is an operator which keeps *N* of the individuals in a generation and discards the individuals which are poor in terms of WAD. A percentage parameter P_{sel} is applied to cut the individuals in a generation when the size of generation N_{gen} is smaller than the selection threshold $\widehat{N}_{gen}$, and if the size of N_{gen} is greater than $\widehat{N}_{gen}$, only $\widehat{N}_{gen}$ individuals will remain.

$$\begin{cases} N = P_{sel} * N_{gen} \text{ IF } N_{gen} < \widehat{N}_{gen} \\ N = \widehat{N}_{gen} \text{ IF } N_{gen} \geq \widehat{N}_{gen} \end{cases} \quad (4-11)$$

4.3.7 Terminating condition

The searching algorithm will be terminated when the terminating condition is satisfied. In a P2P system, the terminating condition is that the individual with the smallest WAD value does not change any more in N_{ter} iterations.

4.4 Negotiation strategies

As each agent generates its local solution by itself without knowledge from the other junction agent, the junction agent may search for its local solution too greedily, and the final solution may not be feasible globally. After the classification operator, all the individuals are classified into two groups, feasible individuals and unfeasible individuals. And in each group, the individuals are ranked with their objective function values from low to high. However, the individual which has the lowest WAD value may not be in the feasible group, so the two agents need to trade off between their local optimum and global feasibility to ensure their final solution is optimised and feasible. As a result, a series of negotiation strategies is developed to achieve the final global solution.

Definition 4-3: Within the P2P system, the individual with the lowest WAD value in a generation of junction agent J_m is named as the *best* solution/individual, stated as $\boldsymbol{X}_{best}^{J_m}$, and its WAD value is $WAD_{best}^{J_m}$. On the other hand, the individual with the lowest WAD value in the group other than the group with the best individual is called the *top* individual/solution, stated as $\boldsymbol{X}_{top}^{J_m}$, and its WAD value is $WAD_{top}^{J_m}$.

Here, we introduce an indicator variable $k_{neg}^{J_m}$ where $J_m \in \mathbb{J}$, and $k_{neg}^{J_m} = 0$ stands for the best individual in the upper group in the junction agent J_m; otherwise, $k_{neg}^{J_m} = 1$. So the negotiation between the two junction agents in the bottleneck section is illustrated in the following pseudocode, and detailed in the following section.

Pseudocode of negotiation

IF $k_{neg}^{J_m} = k_{neg}^{J_n} = 0$, *(condition 1)*

    Agreement achieved;

IF $k_{neg}^{J_m} = 1, k_{neg}^{J_n} = 0$, *(condition 2)*

    Calculate Gap_{J_m}*;*

    Send Gap_{J_m} *and a sequence-change request to junction* J_n*;*

    Junction J_n *generates a new group with best individual* $\widehat{\boldsymbol{X}}_{best}^{J_n}$*;*

    Calculate Gap_{J_n}*;*

    Select final solutions from each junction based on the value of Gap_{J_m} *and* Gap_{J_n}*, agreement achieved;*

IF $k_{neg}^{J_m} = k_{neg}^{J_n} = 1$, *(condition 3)*

    Calculate Gap_{J_m} *and* Gap_{J_n}*;*

    Assume $Gap_{J_m} \leq Gap_{J_n}$*; junction* J_n *send a sequence-change request to junction* J_m*;*

    Junction J_m *generates a new group of individuals with* $\widehat{\boldsymbol{X}}_{best}^{J_m}$ *and* $\widehat{\boldsymbol{X}}_{top}^{J_m}$*;*

    IF $\widehat{\boldsymbol{X}}_{best}^{J_m}$ *is in top layer in the new group*

        Agreement achieved;

    IF $\widehat{\boldsymbol{X}}_{best}^{J_m}$ *is in bottom layer in the new group*

        Switch to condition 2;

Condition 1: $k_{neg}^{Jm} = k_{neg}^{Jn} = 0$

In this condition, both of the agents achieve their local optimal solution with the same sequence as each other. So, the final solution can be carried out by merging the best individual in each agent $\boldsymbol{X}_{best}^{Jm}$ and $\boldsymbol{X}_{best}^{Jn}$, and the final agreement is reached.

Condition 2: $k_{neg}^{Jm} = 1, k_{neg}^{Jn} = 0$

In this condition, one of the agents finds its best solution has a different sequence for the trains not dominated by itself against the interface variable from the other agent, and the other agent finds its local optimum has the same sequence as the interface variable from the other agent. So, in this case, the negotiation protocol should be fired, and one of the agents will trade off its local solution between lower WAD value and global feasibility.

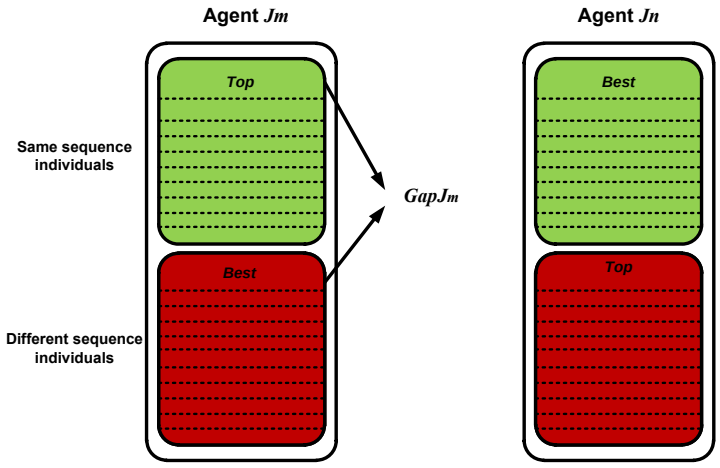


Figure 4-5 Step 1 of negotiation in condition 2; calculate the gap value

Step 1. Calculate a gap value between the WAD values of the top and best solution in junction agent J_m, $Gap_{J_m} = WAD_{top}^{Jm} - WAD_{best}^{Jm}$ (see Figure 4-5). Send the value of Gap_{J_m} with a sequence-change request to agent J_n; the sequence request includes the sequence of the trains which are dominated by agent J_n in the best solution $\boldsymbol{X}_{best}^{Jm}$.



Figure 4-6 Step 2 of negotiation in condition 2

Step 2. The junction agent J_n will generate a new group of individuals with respect to its current individuals and the sequence-change request. Based on the sequence-change request, local searching is applied to make further improvement possible after the sequence of its dominated trains is changed. At the end of step 2, junction J_n holds the primary individuals with its best individual $\boldsymbol{X}_{best}^{J_n}$, its WAD value $WAD_{best}^{J_n}$ and a new group of individuals whose dominated trains have the same sequence as the sequence-change variable; its best individual is $\widehat{\boldsymbol{X}}_{best}^{J_n}$ with the WAD value of $\widehat{WAD}_{best}^{J_n}$. Then, a gap value is calculated as $Gap_{J_n} = \widehat{WAD}_{best}^{J_n} - WAD_{best}^{J_n}$ (see Figure 4-6). In summary, step 2 generates an improvement of the WAD value of junction agent J_m and a deterioration of the WAD value of junction agent J_n if the sequence of the trains dominated by junction J_n is changed.

Step 3. In this step, the junction agent J_n will accept or reject the sequence-change request based on the value of Gap_{J_m} and Gap_{J_n}, and carry out the final solution.

a) If $Gap_{J_n} \leq 0$, which means the sequence-change request and its further local searching lead to a better individual than before, the sequence-change request is accepted, the primary generation in junction agent J_n is discarded, and the final agreement is achieved by merging the solution of $\boldsymbol{X}_{best}^{J_m}$ and $\widehat{\boldsymbol{X}}_{best}^{J_n}$ (see Figure 4-7(a)).

b) If $Gap_{J_n} > 0$ and $Gap_{J_m} > Gap_{J_n}$, which means that the improvement of junction agent J_m is more than the deterioration of junction agent J_n, the sequence-change request is accepted, the primary generation in junction agent J_n is discarded, and the final solution consists of $\boldsymbol{X}_{best}^{J_m}$ and $\widehat{\boldsymbol{X}}_{best}^{J_n}$(see Figure 4-7(a)).

c) If $Gap_{J_n} > 0$ and $Gap_{J_m} < Gap_{J_n}$, this means that the improvement of junction agent J_m is less than the deterioration of junction agent J_m. As a result, the sequence-change request is rejected, the new group in junction agent J_n is discarded, and the final solution is generated by $\boldsymbol{X}_{top}^{J_m}$ and $\boldsymbol{X}_{best}^{J_n}$ (see Figure 4-7(b)).

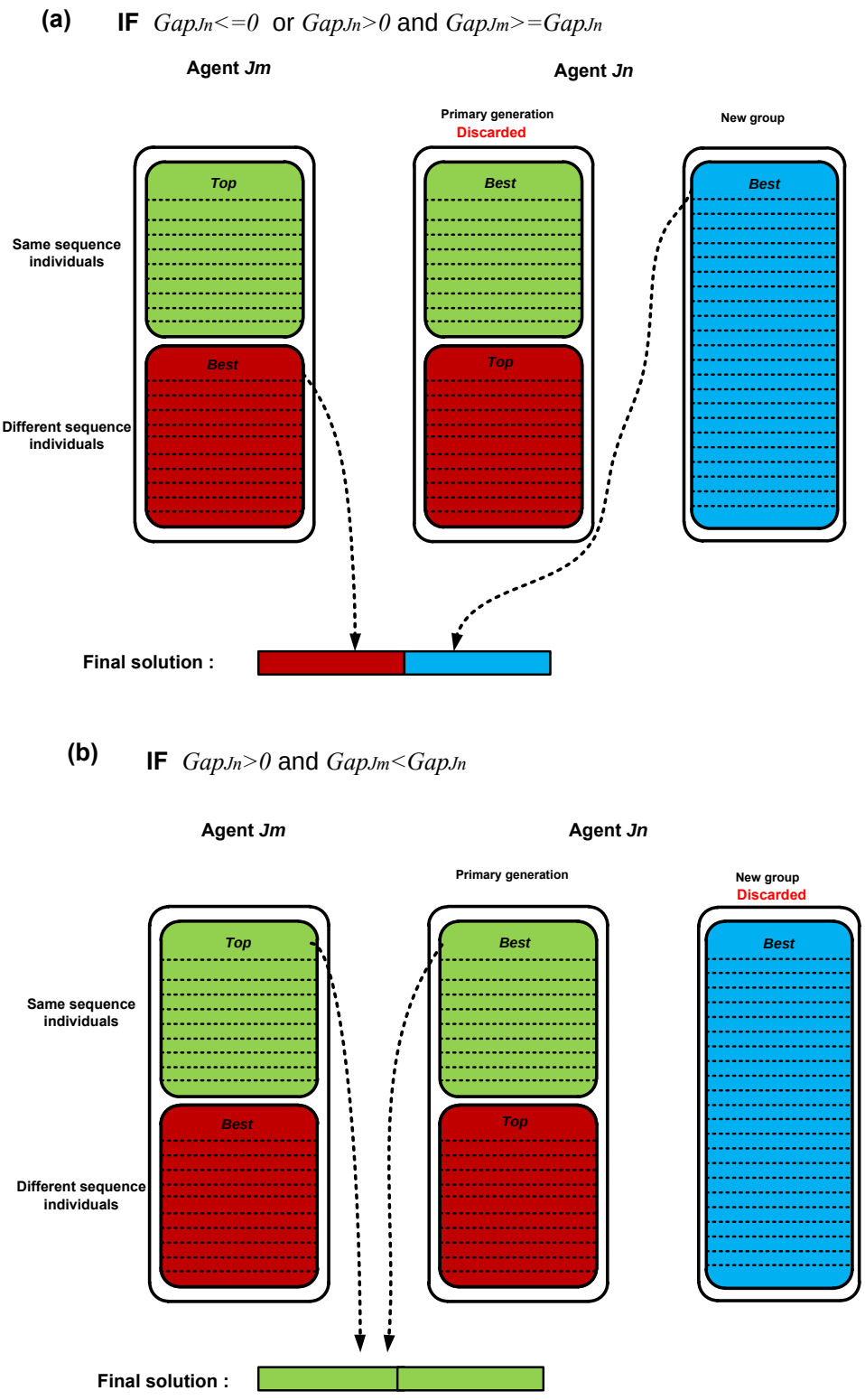


Figure 4-7 Step 3 of negotiation in condition 2

At the end of step 3, the final solutions are generated, and the negotiation indicator k_{neg}^{Jm} is reset as zero.

Condition 3: $k_{neg}^{Jm} = k_{neg}^{Jn} = 1$

In this condition, both of the top individuals are in a different sequence group in each agent, so the trade-off between two agents becomes very complex. To make sure there is final convergence, one of the agents has to change its sequence of dominated trains first and to reduce this problem to condition of 2.

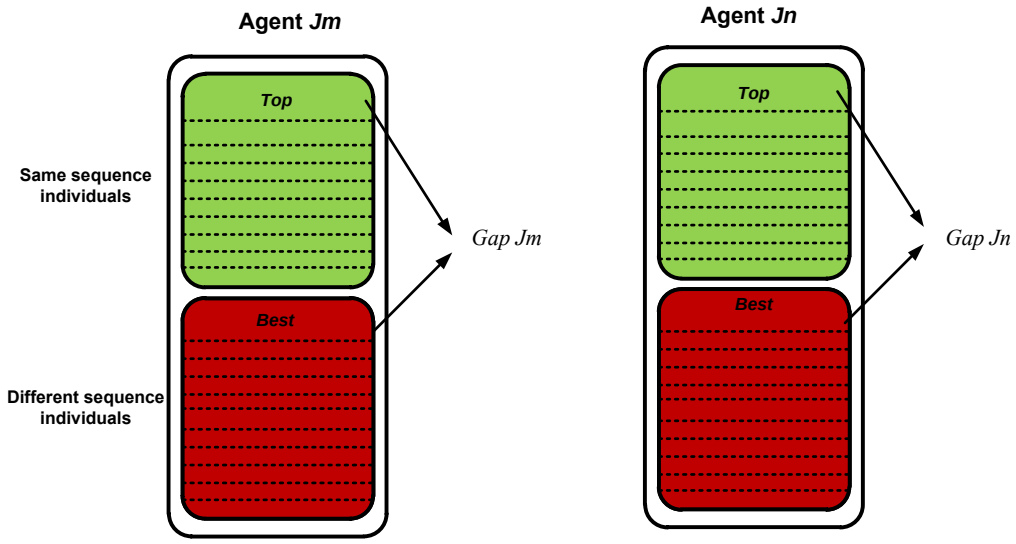


Figure 4-8 First step of negotiation in condition 3

Step 1. Within each agent, the gap values between the top and best individuals are calculated respectively: $Gap_{J_m} = WAD_{top}^{Jm} - WAD_{best}^{Jm}$ and $Gap_{J_n} = WAD_{top}^{Jn} - WAD_{best}^{Jn}$ (see Figure 4-8).

Step 2. Within the second step, the value of Gap_{J_m} and Gap_{J_n} will be contrasted to find which one is smaller. Here, we assume $Gap_{J_m} \leq Gap_{J_n}$, namely, junction agent J_n can have more improvement if the sequence of trains that is dominated by agent J_m is changed. So, the junction agent J_n will generate a sequence-change request. The junction agent J_m will generate a new group of individuals whose sequence of the dominated trains is the same as the sequence-change request. The

new group of individuals will also be classified into two classes with respect to the interface variable from junction agent J_n (see Figure 4-9).



Figure 4-9 Second step of negotiation in condition 3

Step 3. In step 3, the best and top individual, $\widehat{\boldsymbol{X}}_{best}^{Jm}$ and $\widehat{\boldsymbol{X}}_{top}^{Jm}$, in the new group will be identified, and a decision will be made based on the place of the best individual in the new group.

a) If the best individual is in the top layer in the new group, the primary generation in junction agent J_m is discarded; the final agreement is achieved by merging the solution of $\widehat{\boldsymbol{X}}_{best}^{Jm}$ and $\boldsymbol{X}_{best}^{Jn}$ and resetting $k_{neg}^{Jm} = 0, k_{neg}^{Jn} = 0$ (see Figure 4-10(a)).

b) If the best individual is in the bottom layer in the new group, the primary generation in junction agent J_m is discarded. As the sequence of dominated trains in agent J_m has been changed in the new group, the best individual in junction agent J_n will be switched into the top layer with the generation in the next searching iteration, so the indicators will become $k_{neg}^{Jm} =$

$1, k_{neg}^{Jn} = 0$, and the negotiation is degraded as the second condition (see Figure 4-10(b)).

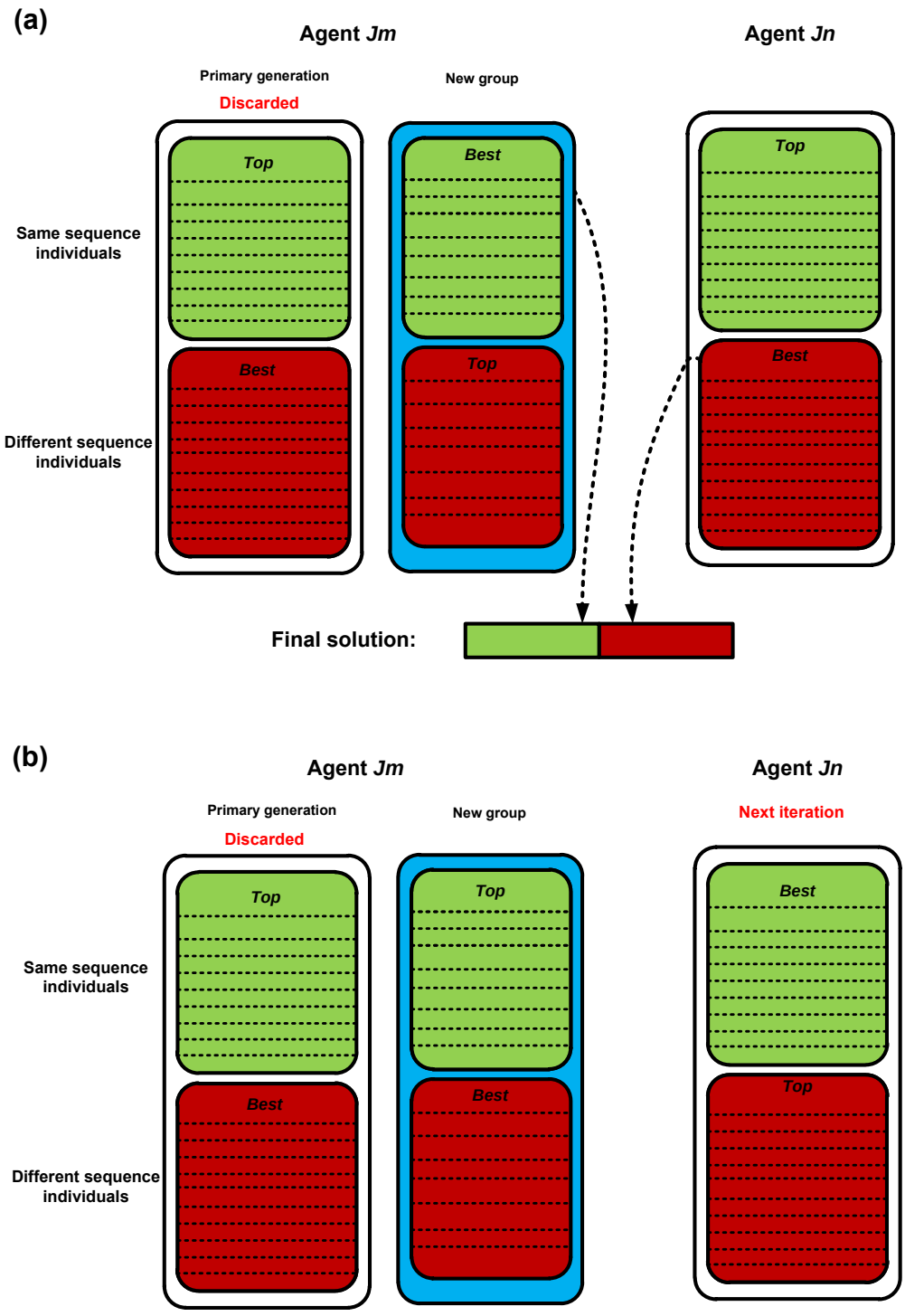


Figure 4-10 Third step of negotiation in condition 3

The whole negotiation will be ended as the two agents achieve agreement of their solutions, namely $k_{neg}^{Jm} = k_{neg}^{Jn} = 0$. The whole process of decision-making in the P2P system for a railway bottleneck section is shown in the flow chart below.

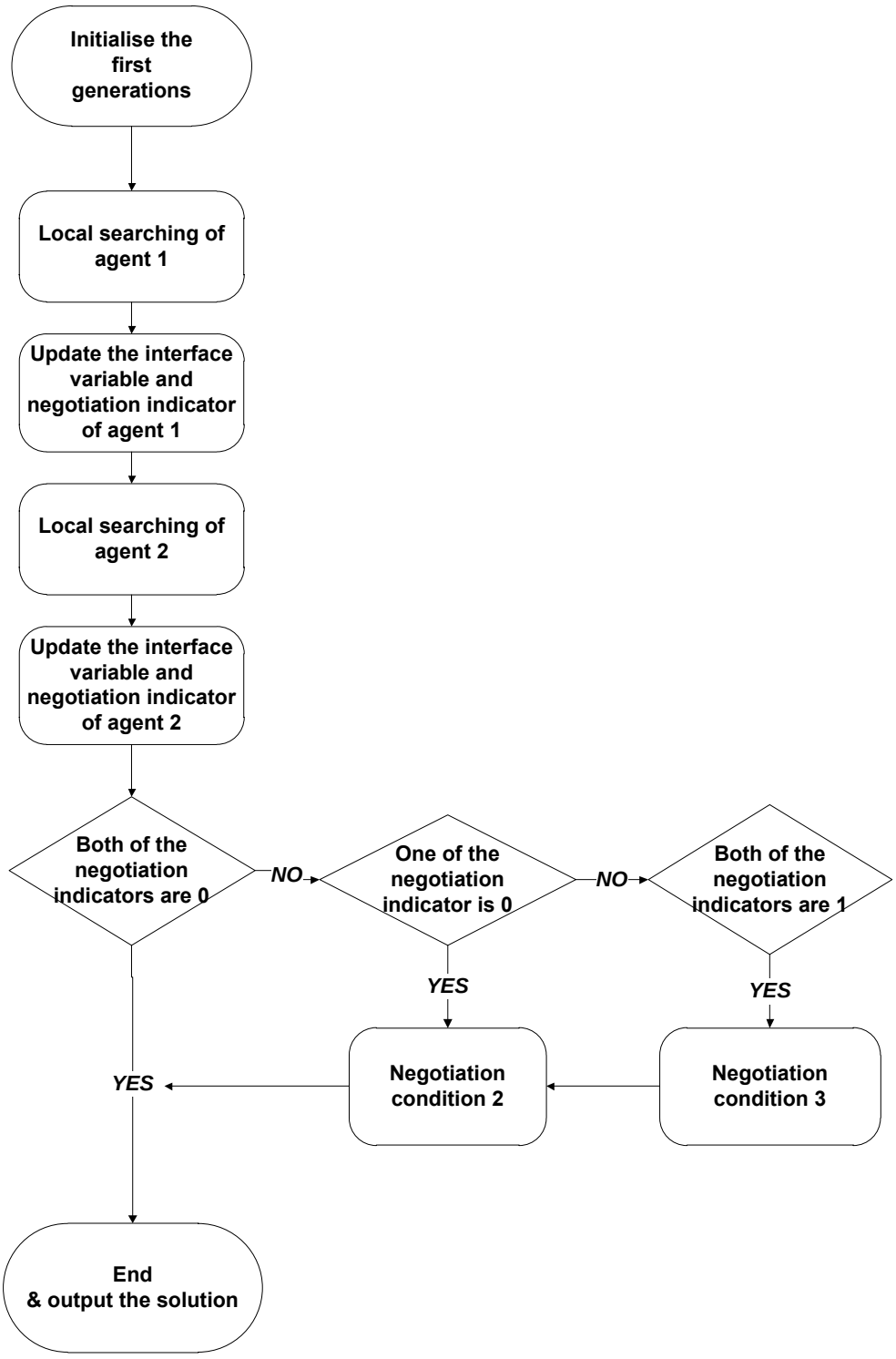


Figure 4-11 Flow chart of decision-making in P2P system

4.5 Centralised system

Due to the coupling constraints within the railway network, a centralised approach is the traditional method for train rescheduling problems. A centralised approach can tackle the whole problem as a large optimisation problem. In this chapter, the centralised approach is also applied to solve a train rescheduling problem in a network bottleneck section. The centralised approach will be compared with a P2P system and an FCFS dispatching rule in the final case study. In this section, the centralised GA is illustrated, and to ensure the P2P system and centralised system are tested based on the same searching capability, the data settings for each approach are exactly the same. The pseudocode of the centralised approach is shown below:

Algorithm 4-2: GA for a centralised system

```
1   Initialisation
2   WHILE (! The terminating condition is satisfied)
3       Crossover;
4       Mutation;
5       Modification;
6       Sorting;
7       Selection;
8   END;
9   Output the best solution.
```

In the algorithm, the control area of both junction agents in the P2P system is merged into a single control area, and all the trains in the control area in the centralised system will be considered in the optimising control step. In addition, the junction passing times of all the trains in the control area of the whole network are organised as a single individual. The

FCFS is also inserted as an individual in the first generation of the algorithm. The same crossover, mutation and selection methods are applied as for the P2P system, and the modification strategies (4-5) to (4-9) are applied in the centralised system. Furthermore, in equation (4-10), the passing time of train τ_m is not an unknown variable, so the minimum travelling time between two junctions is guaranteed by equation (4-11).

$$\begin{cases} \text{IF } t_{\tau_m}^{J_n} < t_{\tau_m}^{J_m} + t_{\tau_m}^{J_m,J_n} \\ \text{THEN } t_{\tau_m}^{J_n} = t_{\tau_m}^{J_m} + t_{\tau_m}^{J_m,J_n} \end{cases} \quad (4-11)$$

In the sorting operator, due to the absence of a classification operator, all the individuals in a generation are simply ranked from the individual with the lowest WAD value to the individual with the highest. Moreover, the terminating condition is also set, as the individual with the smallest WAD value does not change any more in N_{ter} iterations. Finally, the final solution will be divided into a dispatching solution for the two junction agents, and the junction agents will generate their future dispatching and interlocking actions based on the final solution.

4.6 Case study

A bottleneck section in the UK was chosen for testing the P2P and centralised methods for train rescheduling. The bottleneck section (North Stafford Junction and Stenson Junction) is one of the busiest in the UK. Both junctions are level junctions, and the bottleneck section links three large stations in England: Derby, Stoke-on-Trent and Loughborough. So, the bottleneck section faces a great challenge in dispatching trains in daily operation. The layout of the section and its designed volume per hour are shown in Figure 4-12.

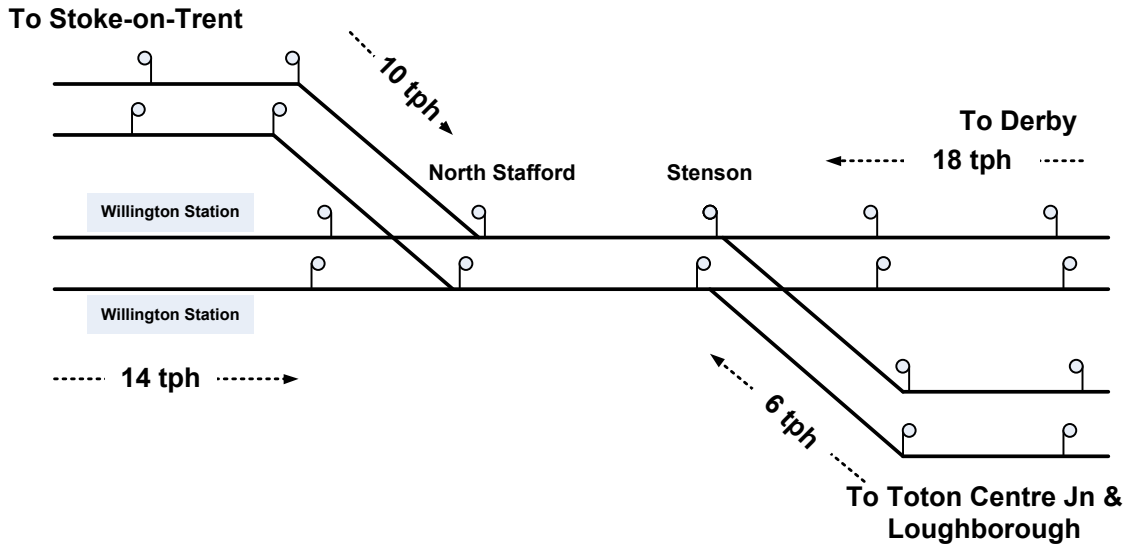


Figure 4-12 North Stafford Junction and Stenson Junction bottleneck section

Within the bottleneck section, there are 10, 14, 18 and 6 trains hourly passing the section from Stoke-on-Trent, Willington, Derby and Loughborough, respectively. Half of these trains will be dispatched to different destinations to the geographic opposite routes.

4.6.1 Scenario design

To test the proposed algorithms in the train rescheduling problem in real railway operational circumstances, a variety of train delay scenarios was designed to formulate different delay conditions. Based on statistical research (Albrecht et al., 2008; Yuan, 2006), the main events which lead to primary train delays are technical failure, an increase in the number of passengers, and bad weather conditions. Technical failures include points failure, signal power failure, track circuit failure and telecoms failure. They usually result in delay to a single train or affect trains on a single line, so to perform a technical failure we inserted train delays for a random train or all the trains from a line. An increase in passenger volume can cause prolonged alighting and boarding times for a train in a station area and may also lead to knock-on delays if connections exist. So, to simulate delays caused by passenger flows, we selected trains from each route randomly and inserted

appropriate delay values. Bad weather is the most common problem which leads to delays over a large area. Bad weather includes snow, ice, fog, high winds and flooding. As a result, trains have to run at lower than scheduled speeds, and inherently it causes large-scale train delays across the whole network. So, under conditions of bad weather, all the trains are delayed in the network, with different distributions. All the delays were inserted at the control boundary of the bottleneck section, aiming to simulate train delays from the circumstance of the whole network and their effects on bottleneck sections.

In scenario design, an appropriate distribution model of delay should be chosen. The series of data sets needs to correspond with random events in railway daily operations. In previous research, uniform, Gaussian and exponential distributions have been used to formulate the delays in railways (Albrecht et al., 2008). In this paper, all of the delay distributions were applied in the scenario design, and different delay conditions were formulated by varying data settings. Trains with negative delays and readjustment of train speed profile are not covered in this paper. The delay probability density functions are shown below, and the detailed scenarios are shown in Table 4-1.

a) Uniform distribution

   The delay in minutes is mapped in the scope of $[a, b]$ min with uniform probability density.

b) Gaussian distribution

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (4-12)$$

   where μ and σ stand for the median value and standard deviation of the distribution, respectively.

c) Exponential distribution

$$f(x|\sigma) = \begin{cases} \frac{1}{\sigma} e^{-\frac{x}{\sigma}}, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (4-13)$$

where σ stands for standard deviation of the distribution.

Table 4-1 Different scenarios and their data settings

Delay condition	Scenario ID	Delay distribution type	Data setting (min)	Proportion of delayed trains	ID of delayed trains
Technical failures	1	Uniform	$a = 0, b = 30$	1/24	Randomly chosen
	2	Gaussian	$\mu = 10, \sigma = 2$	1/24	Randomly chosen
	3	Exponential	$\sigma = 10$	1/24	Randomly chosen
	4	Uniform	$a = 0, b = 8$	1/24	Randomly chosen
	5	Uniform	$a = 0, b = 10$	1/4	All the trains from Willington station
	6	Uniform	$a = 0, b = 10$	3/8	All the trains from Derby
	7	Gaussian	$\mu = 10, \sigma = 2$	3/8	All the trains from Derby
	8	Gaussian	$\mu = 5, \sigma = 1.2$	1/4	All the trains from Willington station
	9	Exponential	$\sigma = 5$	1/4	All the trains from Willington station
	10	Exponential	$\sigma = 5$	3/8	All the trains from Derby
Increase of passenger volume	11	Uniform	$a = 0, b = 20$	1/6	Average selected from each route
	12	Uniform	$a = 0, b = 20$	1/3	Average selected from each route
	13	Uniform	$a = 0, b = 20$	1/2	Average selected from each route
	14	Gaussian	$\mu = 10, \sigma = 2$	1/6	Average selected from each route
	15	Gaussian	$\mu = 10, \sigma = 2$	1/3	Average selected from each route
	16	Exponential	$\sigma = 5$	1/6	Average selected from each route
	17	Exponential	$\sigma = 5$	1/2	Average selected from each route
Bad weather	18	Uniform	$a = 0, b = 20$	1	All of the trains
	19	Exponential	$\sigma = 10$	1	All of the trains
	20	Gaussian	$\mu = 10, \sigma = 2$	1	All of the trains

4.6.2 Computational experiments

Both algorithms, centralised and P2P systems, were applied to the test case shown above. To emphasise the structural difference which is only within the two algorithms, the local searching GA was set at the same level. The data settings are shown below:

- The simulations take place in MATLAB R2015b, on a computer with an Intel i7-4790 processor @ 3.60 GHz and 12.0 GB RAM;
- The purpose of train rescheduling in this case study is to ease train conflicts in a bottleneck section and let the trains pass this area as soon as possible. So, the weightings assigned for trains from Stoke-on-Trent, Willington, Derby and Loughborough are set as $5:7:9:3$, respectively;
- Two-point crossover and extended-line mutation are chosen in the proposed GA;
- $N_{ter} = 50$;
- $N_{ini} = 50$;
- $t_{slot} = 30$ min;
- $P_{sel} = 0.8$;
- $\widehat{N}_{gen} = 200$.

In most testing scenarios, both P2P and centralised systems can generate the same solution with FCFS. However, to show the difference in searching capability between P2P and centralised systems, a scenario will be discarded if three approaches generate the same solution, and a scenario will be kept when one or both of the systems has a different solution against FCFS. Each scenario shown above includes 2000 simulation experiments; the mean values of WAD in each scenario are shown in Figure 4-13 and Figure 4-14, and Table 4-2 summarises the computational time for each system and the percentage of better trials.

In Table 4-2, the detailed computational time for each system is shown. In columns 2-4, the maximum, minimum and average computational time in seconds for the P2P system is shown, respectively. The same indicators for the centralised system are described in columns 5-7; within these indictors, the values are marked as bold when the P2P system has a longer computational time than the centralised system. Furthermore, the percentage of better trials identifies the percentage of solutions from the decentralised system which has a smaller WAD in the same terms as the centralised system (column 8).

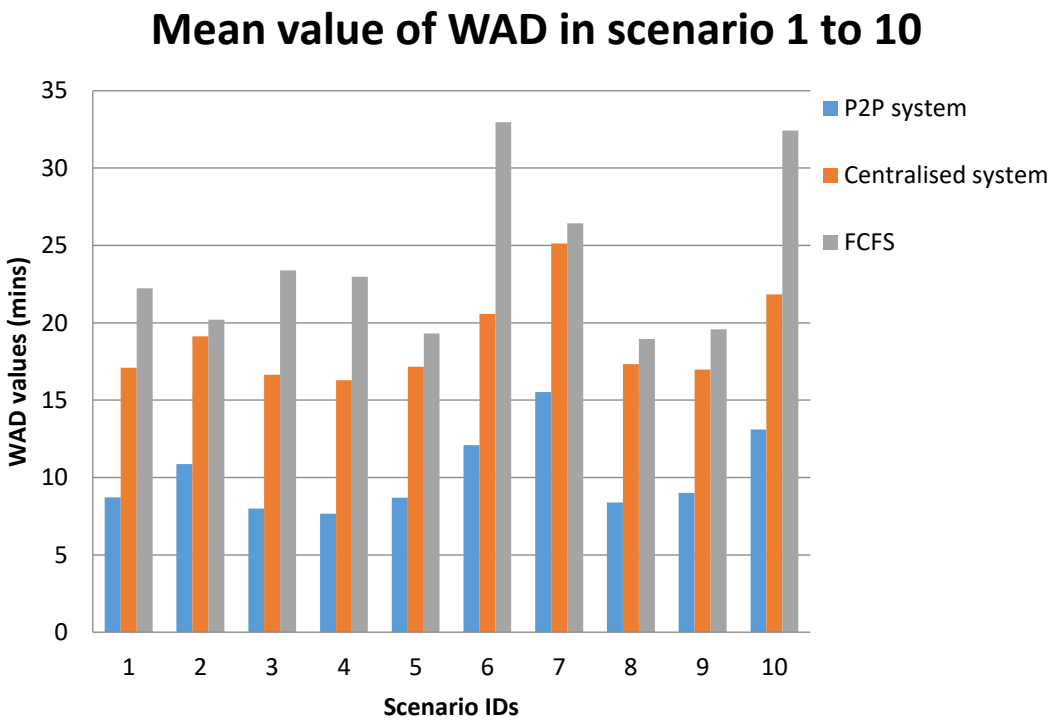


Figure 4-13 Simulation results of scenarios 1 to 10

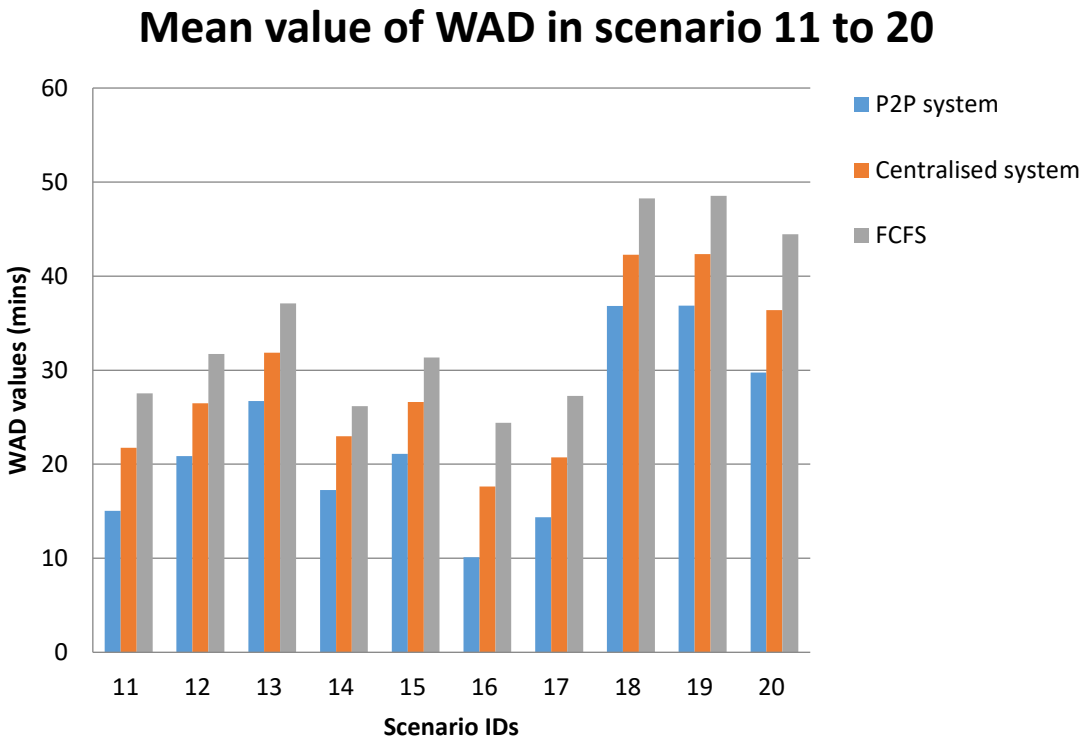


Figure 4-14 Simulation results of scenarios 11 to 20

Table 4-2 Maximum, minimum and average results for computational time and better trial percentage

Scenario ID	P2P system			Centralised system			Better trial (%)
	Max. (s)	Min. (s)	Avg. (s)	Max. (s)	Min. (s)	Avg. (s)	
1	21.0601	8.4085	10.8846	41.4807	9.4225	13.1183	99.5
2	27.3782	6.9108	8.3017	32.0738	6.9732	10.1241	99.95
3	**90.9174**	4.5240	8.8608	33.1190	6.6768	9.8943	99.95
4	27.8462	6.2712	8.3997	40.2951	9.7033	13.0430	99.95
5	29.9834	6.1776	12.0753	40.5447	9.5005	13.5942	99.6
6	29.5310	**7.6536**	9.8120	32.0436	7.0200	10.4141	100
7	24.1490	7.5912	9.7056	37.0190	7.0668	10.0324	100
8	26.2394	6.5520	10.1232	31.9178	7.1760	10.3301	100
9	**37.2158**	6.5208	**12.4386**	36.9722	6.8796	10.3356	99.6
10	29.2502	6.2400	9.5008	32.1674	7.2384	10.7039	99.95
11	28.9694	6.3648	9.6118	41.8395	8.9389	14.3879	99.55
12	31.5905	4.7736	10.1245	42.2451	8.6269	15.4936	99.4
13	29.4062	6.0840	10.2405	44.4603	8.6581	15.1318	99.5
14	38.2986	**7.5192**	9.9213	35.9894	6.7392	10.8976	99.5
15	29.0318	7.3476	10.2237	33.4154	6.8172	10.9810	99.4
16	**58.7258**	5.8656	9.7447	38.3606	7.2886	11.3347	99.45
17	**81.3953**	5.6784	10.8043	35.0534	6.8484	11.8816	98.75
18	**38.4810**	4.4304	**11.5137**	30.2018	6.7548	10.6452	98.4
19	**72.7026**	5.4912	11.3092	42.1499	8.9545	14.0449	98.1
20	**45.8146**	6.5676	**10.5317**	32.1830	7.1604	10.1917	99.5

4.6.3 Analysis and discussion

On the one hand, aside from the data above, the benefits of a P2P system in a structural view are discussed. The solution for searching in a P2P system only includes the timetable of each train in its local junction, and the final solution is obtained by gathering the solution generated by each junction agent. However, within a centralised system, the solution for the whole network is generated by a single dispatching agent. Therefore, due to partition of the network in the P2P system, both the computational complexity and required memory space during solution searching are halved for each of the local agents.

On the other hand, column 8 in Table 4-2 shows that the percentage of better trials is always above 98%. Specifically, the P2P system has a greater opportunity of finding a better solution than the centralised system. Furthermore, according to Figure. 4-13, the improvement in WAD value (with respect to FCFS) of the P2P system is also larger than that of the centralised system, especially in scenarios 1 to 10. As the computational complexity is degraded in the P2P system, the local GA performs more greedily than it does in the centralised system. Although the solution made by local searching in a P2P system may not be feasible globally, the global feasibility can be corrected by its further negotiation. Within scenarios 1 to 10, the delay usually occurs in a single agent, so the solution from the P2P system can make more improvement than that from the centralised solution. In the rest of the scenarios, the delay is mapped within the whole network, so the improvement of the P2P system is not as significant as that of the centralised system in scenarios 1 to 10. As a result, both the percentage of better trials and the differences in WAD between the P2P and centralised systems are reduced. Moreover, the proposed negotiation protocol allows each agent in the P2P system to trade off its local solution

with the comparison between gap values, so the negotiation avoids reaching solutions that fall into the local minimum, which leads to better solutions being found in the decentralised system in scenarios 11 to 20. However, as the number of delayed trains increases, the complexity of negotiation is raised. So, the computational time of the P2P system is not usually less than that of the centralised system, especially in scenarios 18 and 20. In other words, proposed negotiation strategies provide temporary constraints to the decision-making process in a junction agent, the exchanged data (interface variables) enable a junction agent overcome its myopia of dispatching decision of the other junction due to the decomposition of the bottleneck section and consider the interaction of the other junction, which ensure the decomposition of network does not degrade the quality of final solution of entire bottleneck section. Moreover, with the decomposition of a bottleneck section, computational complexity is significantly reduced to single junction level, so the computational time of the whole system can be shorter than centralised approach. In some specific scenarios, the computational time comes extremely huge, such as 90.9174 s in scenario 3, 81.3953 s in scenario 17 and 72.7026 s in scenario 19, which results from the P2P system failed on negotiation and conflicts between generated solutions of each agent cannot be solved by degrading the solutions of a single junction. The key reason which leads to negotiation failure is the GA is not usually stable to find optimal solution in every scenario, so a suboptimal solution may be generated when a new interaction variable is received by a junction agent; the P2P system may turn confused due to the new solution comes worse than the previous iteration and the P2P system has to start the searching again which leads to the computational time doubled or tripled.

However, it is hard to extend the P2P system to large railway network including more than two junctions. As the number of junctions in the considered network increases, there are two approaches to tackling the train rescheduling problem with a P2P system: (1)

apply more agents for the considered junctions, (2) let one agent consider more junctions in the network. In the first option, as the number of junction agents is more than two, the negotiation among the junction agents becomes more complex than in a two-agent system, and the P2P negotiation becomes exhausted when it is applied to a system with more than two agents. In the second option, the negotiation between two agents is achievable, but the searching algorithm in each agent should be specially developed based on the layout of the rail network. So, the P2P system still has poor scalability though it has better searching capability than the centralised system, and to tackle a train rescheduling problem for a large-scale railway network, a P2P system is not enough.

4.7 Summary

This chapter presents a systematic P2P-based approach to tackle train rescheduling problem in railway bottlenecks; and the designed GA and negotiation strategies of a P2P system, the main innovations in this chapter, are emphasised. Furthermore, the benefit of a P2P system, both from a structural point of view and in terms of computational experiments, is illustrated. The final case study shows that the P2P system can generate a better solution within a reasonable computational time than the centralised approach, though the P2P system does not win over the centralised system in all facets. The computational time in the test scenarios shows that the P2P system is faster than the centralised system in most conditions. The P2P system performs best in scenarios where small numbers of trains are delayed, and longer computational time may be desired when the number of delayed train increases. In the simulated bad weather conditions, the computational time of the P2P system is slightly longer than that of the centralised system, but the solution generated by the P2P system is better than that generated by the centralised system in terms of objective function. Furthermore, it has been demonstrated

that the P2P system has over 98% likelihood of getting a final solution which is better than the centralised approach in terms of WAD value. However, the advantage of faster generation of a solution does not usually exist when a large-scale delay has happened.

This chapter brings us a step closer to tackling railway train rescheduling problems with a distributed system. The high-quality solution and shorter computational time of the P2P system may have a profound effect on real-time decision-making in railway bottlenecks when larger and more complex problems are faced. In further work, more complex bottleneck networks will be considered; however, as discussed in this chapter, the P2P system still lacks scalability when more junction areas are considered. In the author's view, to tackle a train rescheduling problem in a network which includes more than two junctions, a hierarchical structured system should be developed with a specific agent which organises the collaboration of the junction agents. In the following chapter, a hierarchical MAS including a voting-based collaborative approach will be introduced.

Chapter 5

MULTI-AGENT SYSTEM FOR TRAIN RESCHEDULING PROBLEMS IN A LARGE-SCALE RAILWAY NETWORK

In this chapter, a systematic MAS for train rescheduling problems in a large-scale railway network is developed. The MAS is organised with a hierarchical structure which includes a collaborative agent and several junction agents. GA is also chosen as the local searching algorithm, and to achieve the collaborative approach and make a globally feasible solution for the whole network, a new classification operator is developed for future voting actions on collaboration. A microscopic railway simulator is chosen as the simulation testbed to test the proposed approach in a British railway network (Derby station with its surrounding area) and compared it with the FCFS dispatching approach, and the pros and cons of the proposed approach are discussed based on the computational experiments.

5.1 Overview of MAS applications

An MAS is usually organised with several agents; the entire system can divide the whole problem into several sub-problems and those sub-problems are solved by each local agent. As a result, MAS is a promising approach for a problem which has a large geographic area and huge computational complexity. This section provides an overview of recent applications of MAS in traffic management problems and discuss the main reasons for applying MASs to the considered problems.

An MAS-based approach for road traffic was illustrated by Almejalli et al. (2009). Due to the interconnection between traffic situations at different locations of a road traffic network, control actions for a local traffic problem may result in future traffic problems in other network areas. So, to minimise the average travelling time of considered vehicles

across the whole network, coordinative control strategies are developed to ensure all the control actions are fed to the considered network with the same objective. Thus, a decision support system based on MAS was developed to serve the human road traffic operators in a traffic control centre. For the considered traffic management problem in road traffic, the entire network is divided into several sub-networks; each sub-network is operated by its local evaluation agent, and the local evaluation agents are also able to exert their effects on each other via a higher-level agent which finalises the optimal solution for the whole network. The proposed approach is tested on a public traffic network in the Saudi Arabian city of Riyadh and compared with fuzzy logic systems and neural networks; the computational experiments showed that the proposed MAS is able to find the optimal solution in a reasonable execution time.

Another MAS was proposed by Hamidi and Kamankesh (2018) to ease a crowded traffic situation as well as its negative effects (i.e., waiting time, driver stress, extra pollution, etc.) at peak hours in emergency conditions of road traffic. The proposed MAS consisted of an emergency condition agent, path supervision agents and intelligent vehicle agents with a hierarchical structure. In an intelligent vehicle agent, the path of the vehicle is selected and optimised based on ACO; the final solution (path to terminal) generated by an intelligent vehicle agent is decided by the road quality and the possibility of transfer. The emergency conditions agent considers the quality of traffic from path supervision agents with respect to the normal average speed of vehicles in the network with a variable between 0 and 1, and feeds the vehicle agents with a re-optimised path. The proposed MAS was simulated and tested in JACK software which is an agent development environment written by Java, and the proposed algorithm was written in MATLAB and connected with JACK by an interface. However, the drawback of the proposed MAS is the conflict between execution time and solution quality. Due to the longer execution time

of simulation for a large-scale network, the time interval between reality and simulation may lead to the generated solution not being feasible, so the authors would like to extend their algorithm to improve its effectiveness.

Architectures of MAS for traffic management problems on motorways were discussed by Hernández et al. (2002), in their article, they illustrate and compare two systems for making real-time traffic management control decisions in urban motorway network. One of the MAS is organised with a traditional centralised mechanism and the other is organised with a decentralised structure. All the agents are developed with a knowledge-based technique. A case study was made to test the two MAS in an urban motorway network around Barcelona; the agent simulation environment TRYS was used to structure the proposed MAS. The advantages and disadvantages were discussed from the point of view architecture. The decentralised approach was viewed as a promising approach against a centralised MAS. The authors also stated that the decentralised MAS has a bottleneck in synchronisation issues because the alternative signalling plan generated cannot ensure global feasibility in some specific scenarios.

MAS has also been applied to air traffic flow management to reduce the congestion in an airport area using a reinforced learning algorithm (Agogino and Tumer, 2012). Within the article, the control area is divided with respect to its 2D geographical aspects, and an agent is applied to make real-time actions on setting separation between airplanes, ordering ground delays and rerouting conflict planes in the conditions of changing weather, limited resources and conflicting priorities. FACET, an air traffic flow simulator developed by NASA, was applied to test the proposed MAS with artificial and historical data from Chicago and New York airspaces. The computational experiments showed that the MAS can improve air transport congestion by about 80%, and the applied reinforced learning strategy speeds up the simulation by about 60% after 500 steps of learning.

A data-driven MAS which integrates learning, forecasting and planning technologies in the MAS has been proposed for maritime traffic safety management by Xiao et al. (2017). The MAS consists of station–vessel agents, vessel–vessel agents and virtual agent–physical agents. The whole MAS generates solutions to manage maritime data and simulations for maritime traffic movements as well as safety management; three modes have been developed to simulate maritime transport scenarios, i.e., analysis of sailing trajectory, generation of sea routes for every vessel and planning all routes for vessels considered. The proposed MAS also developed a set of universal APIs which can accept different algorithms for modelling and simulation.

A multi-agent approach for crew member reallocation of passenger railways was proposed by Abbink (2014). Railway operators usually need to make real-time decisions on crew reallocation when operational disturbances cannot absorbed by designed timetables. The tuning of rail timetables can lead to a series of crew member reallocations for considered trains which affect crew member allocations in the entire network. Abbink introduced an MAS to tackle the crew dispatching problem which evaluates the crew allocation with the objectives of efficiency, robustness and quality. To minimise the cost of crew members allocation (i.e., crew members should be transferred to the desired train station or area by other trains or taxis), a negotiation strategy is proposed to trade off the solutions generated by each dispatching agent. A railway network in the Netherland was used in this article, and the computational results are promising compared with human dispatching solutions. However, the author also illustrates the bottleneck of the proposed MAS, i.e., it does not consider the working willingness of the crew members and potentially exceeds crew members' planned working hours or work limit.

The research illustrated above is part of the recent applications of MAS in traffic management topics, and the problems solved by proposed MAS have the following commonalities:

1) The problems cover a large-scale geographical area;
2) The problems include a strong interconnection inside;
3) The control system for the problem has an inherently distributed structure.

As a result, the MASs are all developed with a hierarchical structure which tackles the entire problem in a distributed way, and the entire problem is divided into several sub-problems to be solved by local agents. All the MASs have an agent to organise the collaboration among the local agents with high priority to ensure their objectives are globally optimal and feasible, and the data flow within the hierarchical MAS is chosen as a two-pass structure. Two-pass data transmission in a layered MAS is shown in Figure 5-1.

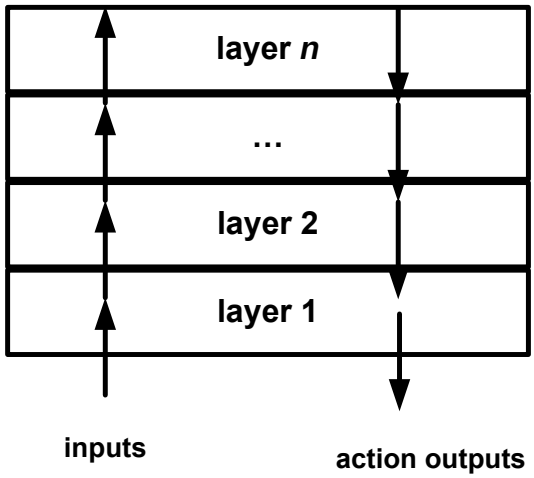


Figure 5-1 Two-pass data flow in a hierarchical MAS

The data flow is indicated by arcs in Figure 5-1; the information is originally collected from the embedded environment, flows up to the top layer and finally flows down to the bottom layer. The main benefit of this kind of data flow is its simple interactions between

two neighbouring layers, and all the interactions between the environment and MAS are made by the bottom layer. In the following sections, the MAS for tackling train rescheduling problems in a large-scale railway network is also designed as a two-pass hierarchical MAS which include junction agents generating real-time dispatching solutions and a collaborative agent controlling the decision-making in the whole network.

5.2 Structure of MAS for large-scale railway traffic management

In this section, the structure of MAS for train rescheduling problems in a large-scale railway network is illustrated. As proposed in Chapter 3, the railway network is decomposed to simple junction level, and a junction agent is applied to solve train rescheduling problems in its local area. So in the context of MAS for train rescheduling problems, the definition of a junction agent (Definition 3-1) is kept in this chapter, and to organise collaboration among the junction agents, a collaborative agent is defined and designed to organise the collaborative strategies after local searching of junction agents. The structure of the MAS for train rescheduling problems is described in Figure 5-2.

Definition 5-1: A *collaborative agent* is an agent which monitors each junction agent under its control area, and ensures the decisions made by each junction agent are feasible globally.

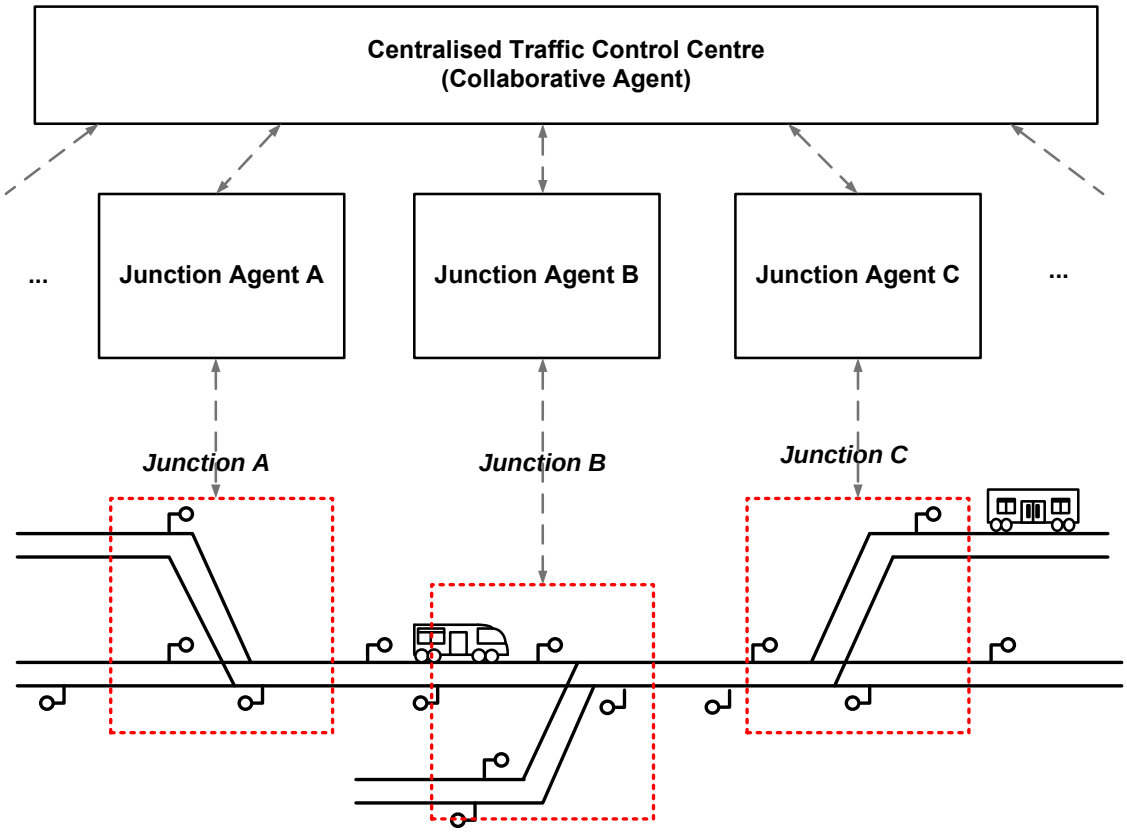


Figure 5-2 Structure of MAS for large-scale train rescheduling problems

The whole MAS consists of three layers: the physical layer which includes the infrastructure of the railway network, the signalling and interlocking and the trains in the considered network, the junction agent layer includes all the junction agents in the considered network, and the collaborative layer includes the collaborative agent. For the collaboration between junction agents and the collaborative agent in railway circumstances, a closed loop to tackle the train dispatching problem is proposed. The junction agents only focus on their local area, make real-time decisions when train rescheduling is inevitable and then submit their local solutions (solution table) to the collaborative agent; the collaborative agent will trade off among the solutions generated by junction agents based on its global inspection and make a final feasible and optimised decision to dispatch the trains in its network. The final decisions will be fed back to each junction agent to generate future actions.

The main reason for designing a hierarchical MAS is that practical railway traffic management system still has a distributed and hierarchical structure. So, it is potentially easy to plug the MAS into our current railway traffic management system. However, the real-time train dispatching decisions are made in a centralised traffic control centre which transfers its decisions to each local sub-area to generate signalling and interlocking actions; and it is obvious that the current system relies strongly on the centralised traffic control centre; as traffic volume increases, the computational complexity will be raised exponentially. In contrast, due to decoupling of the whole system, the large train rescheduling problem is decomposed to each junction agent, and the collaborative agent will only be in charge of the trade-off for each junction agent, so the computational complexity will be reduced. Due to the searching space in each local area being reduced, and each junction agent being able to generate its solution in parallel, the computational time for the whole system can be potentially decreased.

5.3 Algorithm development

In this section, the local searching algorithm, which is embedded in junction agents to make real-time train rescheduling actions, is illustrated. The whole searching process starts from at least one of the considered trains in the whole network being delayed and all the trains in the considered area will be rescheduled. After the local searching in each junction agent, the junction agents will deliver a solution table to the collaborative agent; after all the solution tables are received by collaborative agent, it will start its collaborative strategy to trade off among the solutions generated by every junction agent, and finally find an optimised solution for the entire network.

GA is also kept in the proposed MAS for train rescheduling problems in a large-scale railway network. The GA in this section also use the same initialisation, crossover,

mutation, sorting and selection operators and terminating condition as the GA in Chapter 4. However, the classification operator is replaced with a new operator which classifies all the individuals in final generation of the GA with pairwise relationships of train passing sequence; furthermore, the classification operator is moved to the end of the GA which finally generates a solution table to the collaborative agent for further actions. The pseudocode of the GA in the MAS is illustrated below:

Algorithm 5-1: GA for local searching in MAS

Initialisation;

WHILE *(!the local terminating condition is satisfied)*

    Crossover;

    Mutation;

    Sorting;

    Selection;

END;

Classification;

Output the solution table to collaborative agent.

Within the GA optimisation, the objective function for the entire network (3-1) is also set as the objective of the MAS; to allocate equation (3-1) to each junction agent, the objective function of each junction agent can be set as the total delay of considered trains in the network in its local junction area which is shown below:

$$TD^{Jm} = \sum_{\tau_m \in \mathbb{T}} \left| t_{\tau_m}^{Jm} - t_{\tau_m}^{Jm^{\mathcal{O}}} \right| \qquad (5-1)$$

In the local searching in a junction agent, the dispatching solution for considered trains will be optimised subject to all the constraints other than constraints (3-19) and (3-20). The searching algorithm will be ended when its termination condition is satisfied, and the final generation of GA will be delivered to the classification operator.

Classification

At the end of local searching, a local best solution is found. In a traditional GA, the top solution will be taken as the final solution and output as the dispatching decision. However, within its local searching, constraints (3-19) and (3-20) are not considered; namely, the coupling of train timetables among coupled junctions is not considered, which may lead to the best solution of the local searching not being feasible globally. In this thesis, a voting mechanism is proposed to decide the coupled indicator in equation (3-20); to ease the voting among junction agents, a classification operator is applied.

The local searching will be ended with a generation including the best solution with its local knowledge whose seeds consist of their decision according to the decision variables $k_{\tau_m,\tau_n}^{J_m}$ and $t_{\tau_m}^{J_m}$. With their different decisions on $k_{\tau_m,\tau_n}^{J_m}$ for all the trains passing the junction area, $t_{\tau_m}^{J_m}$ may differ due to different operational constraints raised by decision variable $k_{\tau_m,\tau_n}^{J_m}$. In the classification operator, all the individuals in the final generation will be classified into several pairing groups to indicate the different decisions for decision variable $k_{\tau_m,\tau_n}^{J_m}$, as shown in the following equations.

$$\overleftrightarrow{\tau_m,\tau_n} = \begin{cases} k_{\tau_m,\tau_n}^{J_m} = 1, \left\{ \boldsymbol{X}_1^{J_m}, \boldsymbol{X}_2^{J_m}, \dots, \boldsymbol{X}_{N^{k=1}}^{J_m} \right\} \\ k_{\tau_m,\tau_n}^{J_m} = 0, \left\{ \boldsymbol{X}_1^{J_m}, \boldsymbol{X}_2^{J_m}, \dots, \boldsymbol{X}_{N^{k=0}}^{J_m} \right\} \end{cases} \quad (5-2)$$

$$N^{k=1} + N^{k=0} = \widehat{N}_{gen} \quad (5-3)$$

$\overleftrightarrow{\tau_m, \tau_n}$ states the relationship of ordering between train τ_m and train τ_n when passing the junction. And it consists of the individuals which launch train τ_m before train τ_n, where $k^{J_m}_{\tau_m,\tau_n} = 1$; similarly, the rest of the individuals are classified into a group whose solutions decide to dispatch train τ_n before train τ_m. The subscript of the individual $\boldsymbol{X}^{J_m}$ stands for the index of the individual in each group, and the total number of individuals in each group is $\widehat{N}_{gen}$ which should be the same as the number of individuals in the final generation. In each sub-group, the individuals are still sorted in ascending order of their numerical values for the objective function. The pairing group will not be generated if all the individuals in the final generation have the same order for train τ_m and train τ_n. Finally, the final generation is organised into a solution table which is indexed by $\overleftrightarrow{\tau_m, \tau_n}$ and delivered to the collaborative agent.

5.4 Collaborative strategy: Condorcet voting method

The collaborative strategy among agents in an MAS is usually described as a computational social choice problem (Weiss, 1999). A social choice problem concerns the design and formal analysis of methods of aggregating the preferences of multiple agents. The majority approaches to social choice problem are voting methods which aggregate the preferences of a set of voters over a set of candidates or candidates' features. The voting result will be transferred into a value-based evaluation mechanism which finally determines which candidate should win the election (or, more generally, to choose an alternative from a set of alternatives). In this thesis, the protocol for MAS for generating a global solution for train rescheduling problems in its controlled area is defined as a collaborative strategy, and a voting approach is chosen as the collaborative strategy in the thesis.

There are a variety of voting approaches used in modern world. And the most common voting approaches are first-past-the-post voting (Miller, 1998), the two-round (runoff) system (Green-Armytage and Tideman, 2019), proportional representation (Arnold-Forster, 1884) and ranked voting (Obata, 2003). With the consideration of all the voting approaches, these approaches can be roughly classified into the following four classes:

1) Plurality systems: plurality voting approaches choose the candidate with the highest number of votes (Farrell, 2011). The most common plurality voting approach is the first-past-the-post which is second most electoral system with countries using such as British and America colonies. Furthermore, block voting, plurality-at-large and single non-transferable vote are also used in some counties.
2) Majoritarian systems: majoritarian voting approaches are usually applied to decide the majority of a set alternatives with a single round such as instant-runoff voting (Edwards, 2015) or multiple rounds of ranked voting such as exhaustive ballot (Van De Walle, 2011).
3) Proportional systems: proportional representation is the most widely used approach for election which covers about 80 countries. Single transferable vote is a typical kind of proportional system within which voters choose candidates in a multi-member constituency rather than voting for a party list (Tideman, 1995).

To make the final dispatching solutions of each junction agent feasible globally, a Condorcet voting-based collaborative strategy is proposed which is a majoritarian voting approach. The local solution is searched in each junction with local objectives without considering the coupling in train timetables, so it has a great possibility for each agent generating conflicting solutions which are unfeasible for train operations. Due to the local preference of each agent, the best solution which has the lowest value for its objective function may be cyclised, which is hard to trade off among junction agents. For example,

junction agent A prefers the passing sequence for conflict trains (S1, S2, S3) of S1, S2, S3; the decision of junction agent B is S2, S3, S1, and in junction agent C a solution of S3, S1, S2 is generated. Thus, it is difficult to find a majority decision for the conflict solutions and make an acceptable decision for all the junction agents. So, an appropriate strategy should be proposed to consider all the preferences of junction agents and find the most acceptable majority of the dispatching solutions for each agent.

A Condorcet voting approach, which aims to tackle a social choice problem by a series of voting actions on cyclic preferences, was first proposed by Marquis de Condorcet (1989). Based on the discussion above, the conflict decisions made by junction agents can be called a Condorcet paradox in the context of Condorcet voting, and each cyclic preference is tackled by dividing the cyclic preference into a sequential pairwise election. The final decision is made based on the statistical results of every pairwise election (Gehrlein, 2002). In this section, the detailed approach for adjusting the local solution from each junction agent with a Condorcet voting strategy will be illustrated. The Condorcet voting starts by generating a voting sequence for all conflict decision pairs. Then, the detailed voting approach is illustrated, and the solution selection method is discussed.

After all the junction agents complete their local searching, the collaborative agent will receive all the solution tables from every junction agent and start its voting process to make the final decision for the entire network. As illustrated above, a passing solution in a junction area consists of a series of decision variables $k^{Jm}_{\tau_m,\tau_n}$, so the voting pairs are organised by the decision variables $k^{Jm}_{\tau_m,\tau_n}$. The collaborative agent will check all the indicators $k^{Jm,Jn}_{\tau_m,\tau_n}$ and the related decision variables attached to it, $k^{Jm}_{\tau_m,\tau_n}$ and $k^{Jn}_{\tau_m,\tau_n}$, and a voting pair will be generated if the junction J_m and J_n is strong coupled and the decision

made by each junction on the passing sequence of train τ_m and train τ_n is different, which is stated as follows:

$$\mathcal{V}(J_m, J_n|\tau_m, \tau_n) = 1 \text{ , IFF } k_{\tau_m,\tau_n}^{J_m,J_n} = 1, k_{\tau_m,\tau_n}^{J_m} \neq k_{\tau_m,\tau_n}^{J_n}, J_m, J_n \in \mathbb{J}, \tau_m, \tau_n \in \mathbb{T} \quad (5-4)$$

where $\mathcal{V}(J_m, J_n|\tau_m, \tau_n)$ denotes a voting section which will decide the passing sequence between train τ_m and train τ_n at junction J_m and J_n.

Definition 5-2: Within this thesis, the first two parameters J_m and J_n are called *position variables* in a voting section, and the second two parameters τ_m and τ_n are defined as *voting seeds* in a voting section. The solutions generated by the junction agent J_m for different values of decision variable $k_{\tau_m,\tau_n}^{J_m}$ will be the voting candidates in this voting section. $\mathcal{V}(J_m, J_n|\tau_m, \tau_n) = 1$ stands for the voting between the passing sequence of train τ_m and train τ_n.

The collaborative agent will check all the coupling indicators among junctions and their decisions made on train passing sequence and summarise a voting list including all the voting sections to be executed. To organise the voting sections in an appropriate sequence, the following ordering approach is applied.

1) Voting sections which include the same voting seeds and cover more position variables will be voted first.

2) Voting sections including the same one or two voting seeds and their voting results which may affect the other voting sections will be voted second.

3) Voting sections which only include two junction agents will be voted last.

Once the voting sequence is decided, the voting for each voting section regarding the decision variables with coupling will be started. The voting on each pair of trains is

inherently a trade-off of the passing indicators which have conflicts with each other. From the solution tables for each junction agent, the classified solutions can be indexed with their decision for k_{τ_m,τ_n}^{Jm}, and the voting value is chosen as the objective function value of the best solutions of $k_{\tau_m,\tau_n}^{Jm} = 1$ and $k_{\tau_m,\tau_n}^{Jm} = 0$ among all the junctions which have a conflict in train passing sequence, which is illustrated in equation 5-5.

$$\mathcal{V}(J_1, J_2, \dots, J_{N_v}|\tau_m, \tau_n) \stackrel{\text{def}}{=} \begin{cases} k_{\tau_m,\tau_n}^{J_1,J_2,\dots,J_{N_v}} = 1, \langle \tau_m|\tau_n\rangle = \sum_{J_1,J_2,\dots,J_{N_v}} TD\left(\boldsymbol{X}_{1\,k=1}^{Jm}\right) \\ k_{\tau_m,\tau_n}^{J_1,J_2,\dots,J_{N_v}} = 0, \langle \tau_n|\tau_m\rangle = \sum_{J_1,J_2,\dots,J_{N_v}} TD\left(\boldsymbol{X}_{1\,k=0}^{Jm}\right) \end{cases} \quad (5-5)$$

$$\text{IF } \mathcal{V}(J_1, J_2|\tau_m, \tau_n) = \mathcal{V}(J_2, J_3|\tau_m, \tau_n) = \cdots = \mathcal{V}\left(J_{N_v-1}, J_{N_v}|\tau_m, \tau_n\right) = \mathcal{V}\left(J_{N_v}, J_1|\tau_m, \tau_n\right) = 1$$

where $\mathcal{V}(J_1, J_2, \dots, J_{N_v}|\tau_m, \tau_n)$ denotes voting on the passing sequence of train τ_m and train τ_n among junctions $J_1, J_2, \dots, J_{N_v}$, and the subscript N_v of a junction stands for the number of conflict junctions. The voting values are calculated by the values of objective functions (TD) when different values of decision variables are applied. To deal with a decision which a junction does not include in its solution table, the decision is viewed as extremely bad in its local view, so a very large integer is applied as the value of its objective function. The final decision on train passing sequence depends on the numerical values of $\langle \tau_m|\tau_n\rangle$ and $\langle \tau_n|\tau_m\rangle$ as shown below.

$$k_{\tau_m,\tau_n}^{J_1} = k_{\tau_m,\tau_n}^{J_2} = \cdots = k_{\tau_m,\tau_n}^{J_{N_v}} = k_{\tau_m,\tau_n}^{J_1,J_2,\dots,J_{N_v}} = \begin{cases} 1 & \text{IF } \langle \tau_m|\tau_n\rangle < \langle \tau_n|\tau_m\rangle \\ 0 & \text{, otherwise} \end{cases} \quad (5-6)$$

Once a decision on the passing sequence of trains τ_m and τ_n is made, the collaborative agent will save this part of solution as the final solution and discard all the individuals

which include a different value for the decision variable in the solution table. Then, it moves to the next voting section until all the voting sections are completed.

5.5 Case study

In this chapter, a UK railway network is applied to test the proposed approach for train rescheduling, and the solution generated is compared against an FCFS routing strategy to show its pros and cons. This section starts with a description of the simulator and network, then illustrates the design of the test scenarios; finally, the computational experiments are summarised and discussed. All tests were performed on a computer with an Intel Core i7-4790 CPU @ 3.60GHz processor with 12 GB RAM, and the proposed approaches were commanded in JAVA.

5.5.1 Simulator and simulated network

University of Birmingham railway simulator is a microscopic simulator for railway infrastructure which models train movements with the occupation of a specific track circuit and the actions of signalling and interlocking (Kirkwood, 2015). To deal with a real-time train dispatching problem, a junction panel is applied to define the junction area in the simulation. The real-time dispatching algorithm is attached as a plug-in in this simulator, which collects real-time time data from the simulator and feeds back with the passing sequence of the considered trains. Within the simulator, a junction is defined as a part of the network which is surrounded by several signals, and the track circuit before that signal. Besides that, as a train is approaching the junction, its related routing and interlocking actions will take place with respect to the dispatching rule (the dispatching sequence from the algorithm or FCFS). Figure 5-3 shows an example of the simulator

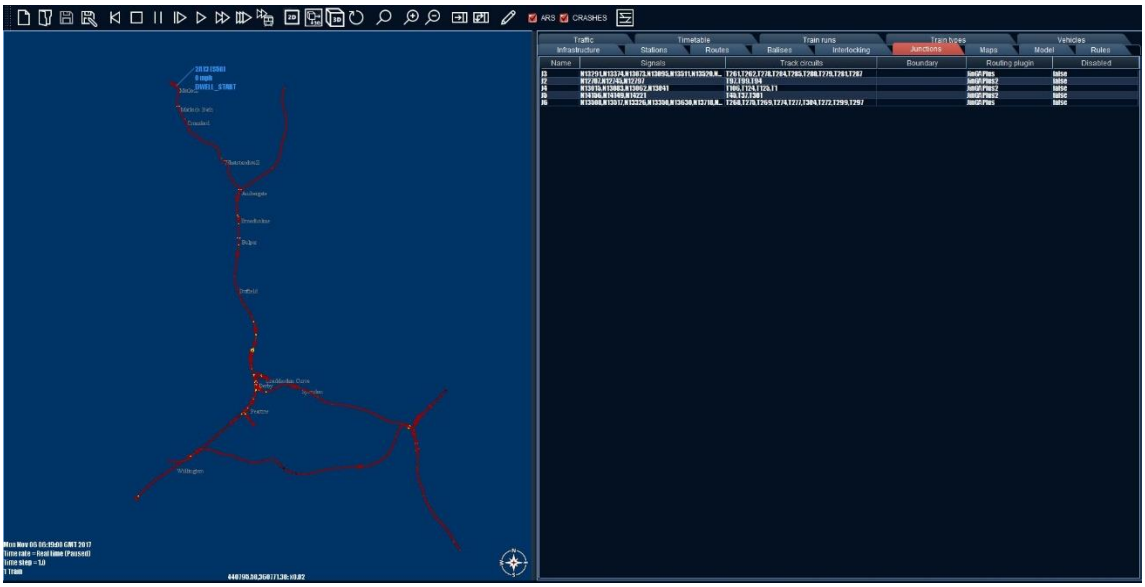


Figure 5-3 University of Birmingham microscopic simulator (example)

In this chapter, Derby station and its surrounding area is chosen to test the MAS-based approach; a brief glance at the network is shown in Figure 5-4. In the simulation, five junctions are defined, covering the main areas where trains may have conflicts. Ambergate Junction is a junction located in the northern part in the network; the trains from/to Matlock station are local services which travel between Derby, Spondon and Matlock which are supposed to stop at every station on their journey. Trains from Clay Cross North Junction are long-distance services with a single stop at Derby station which are launched to Willington and Spondon stations, respectively. The Matlock branch may lead to several kinds of conflict due to the single-track infrastructure. Derby station is surrounded by three junctions (Derby LNW Junction, Derby London Road Junction and Derby Station North Junction); the three junctions can adjust the arrival and departure times of trains in Derby station by their decision on the passing sequence of the trains in their local junction areas. At Derby LNW junction, trains from the Peartree station side can dwell at Derby station on platforms 1 to 6 and even in the sidings. Derby London Road Junction is in charge of the dispatching work for the trains from and to Spondon station. Derby Station

North Junction is in charge of dispatching trains from and to the north side. Forty-nine services were named as *Si*, where *i* is a number, and simulated based on this network which covers normal and peak hours (from 6 a.m. to 10 a.m.) on workdays; the services include local and long-distance passenger trains on the Willington and Matlock branch and Clay Cross North Junction, respectively, local passenger trains and freight trains from Spondon, and long-distance passenger trains dispatched from Derby station; detailed timetables for the simulated services are shown in Appendix A.

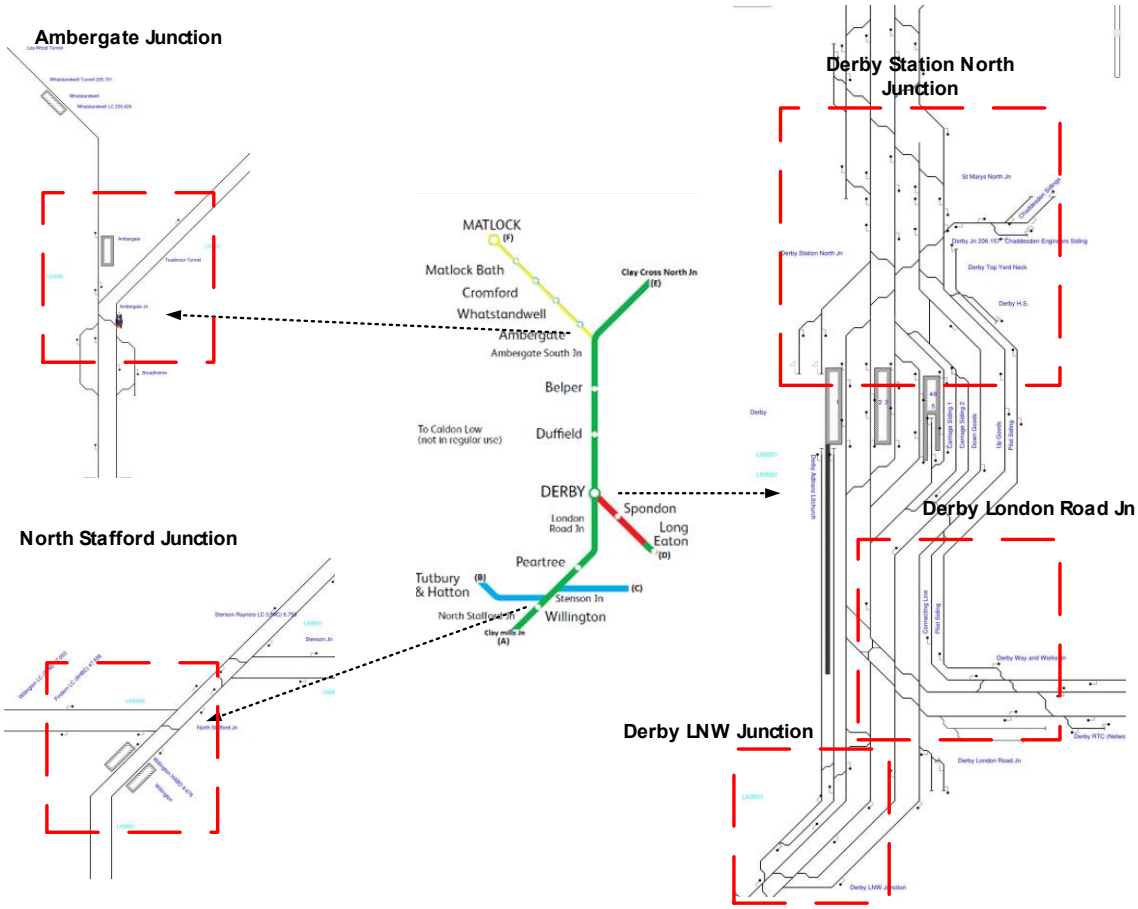


Figure 5-4 Network of Derby Station and its surrounding area and the definition of each junction in the case study

5.5.2 Computational complexity analysis

In the following tables, the first table illustrates the computational complexity at most in a train rescheduling problem, and the computational complexity at most in a junction agent in the proposed MAS. The computational complexity analysis includes the total number of continuous variables $t_{\tau_m}^{Jm}$, binary variables k_{τ_m,τ_n}^{Jm}, railway operational constraints and timetable coupling constraints. And the second table illustrates the exact number of variables and constraints in this case study.

Table 5-1 Computational complexity at most of the whole problem and each junction agent in MAS

Variables or constraints	Total number at most (whole problem)	Total number at most (in MAS)
Continuous variable	$\|\|\mathbb{T}\|\| \cdot \|\|\mathbb{J}\|\| \cdot x$	$\|\|\mathbb{T}\|\| \cdot x$
Binary variable	$\frac{1}{2}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| - 1) \cdot \|\|\mathbb{J}\|\| \cdot x$	$\frac{1}{2}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| - 1) \cdot x$
Railway operational constraints	$\|\|\mathbb{T}\|\| + \|\|\mathbb{T}\|\| \cdot \|\|\mathbb{J}\|\| \cdot x + \frac{1}{2}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| + 1) \cdot \|\|\mathbb{S}\|\| + \frac{1}{2}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| - 1) \cdot \|\|\mathbb{J}\|\| \cdot x + \frac{1}{2}\|\|\mathbb{T}\|\| \cdot \|\|\mathbb{J}\|\| \cdot (\|\|\mathbb{J}\|\| - 1) \cdot x$	$\|\|\mathbb{T}\|\| + \|\|\mathbb{T}\|\| \cdot x + \frac{1}{2}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| + 1) \cdot \|\|\mathbb{S}\|\| + \frac{1}{2}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| - 1) \cdot x + \frac{1}{2}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{J}\|\| - 1) \cdot x$
Timetable coupling constraints	$\frac{1}{4}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| - 1) \cdot \|\|\mathbb{J}\|\| \cdot (\|\|\mathbb{J}\|\| - 1) \cdot x$	$\frac{1}{4}\|\|\mathbb{T}\|\| \cdot (\|\|\mathbb{T}\|\| + 1) \cdot \|\|\mathbb{J}\|\| \cdot (\|\|\mathbb{J}\|\| - 1) \cdot x$

Note: x is the number of passing of a train at junction areas

Table 5-2 The exact values for computational complexity analysis in the case study

Variable or constraints	Whole problem	North Strafford Jn	Derby LNW Jn	Derby London Road Jn	Derby Station North Jn	Ambergate Jn	Collabo-rative agent
Continuous variable	149	27	27	44	25	26	0
Binary variable	2273	351	351	946	300	325	0
Railway operational constraints	4553	459	837	2104	724	429	0
Timetable coupling constraints	16820	0	0	0	0	0	16820

From the complexity analysis above, it is obvious that the entire train rescheduling problem in the case study has huge number of variables to be optimised under significant number of constraints. In general, to solve this MILP optimisation problem, it usually costs about 2 hours in CPLEX optimiser; and with author's experience, solving this problem in centralised way, for example with GA in same parameter setting in this thesis, the huge searching space of the problem may lead to solutions worse than FCFS dispatching rule. In the MAS, the optimisation target is decomposed and allocated to every junction agent in the system, which cuts the searching space into reasonable size for real-time traffic control with considered train rescheduling problem. The key benefit of applying MAS to train rescheduling problem is the allocation of the optimisation to every

junction agent; and the train rescheduling problem is solved in parallel in each junction agent, which reduces the computational time significantly against centralised way. Every junction agent may perform greedier than tackling the whole problem in centralised way. Furthermore, the timetable coupling constraints are tackled by the collaborative agent; though the voting mechanism should readjust the train sequence $k_{\tau_m,\tau_n}^{J_m}$ with the indicator $k_{\tau_m,\tau_n}^{J_m,J_n}$, not all the train-sequence-decisions made by each junction agent have conflicts with each other. So the number of timetable coupling constraints may much smaller that the value in table above due to different test scenario, this feature is guaranteed by classification operator in the MAS. However, a centralised system has to check all the timetable coupling constraints which also prolongs the computational time for centralised system. Therefore, solving train rescheduling problems of large-scale railway networks with MAS can reduce the computational time significantly and potential improve the quality of final solution.

5.5.3 Scenario design

To test the proposed algorithm, three classes of scenarios were designed, i.e., individual train entrance delay, multiple train entrance delay and minor perturbations in station areas. Based on the author's experience, the MAS can generate the same solution as FCFS in the some designed scenarios. In this case study, only scenarios whose solutions are different in terms of the two dispatching approaches are illustrated.

Class I: individual entry delay

A train delay is inserted at the starting point where a train enters the simulation. This event is either entry to the simulation area across the edge of the network, or the beginning of a timetabled journey from a station within the network. The simulation is supposed to show the transition of the single train's delay across the whole network based on different

dispatching approaches. The scenarios are numbered as I_Si, where Si is the delayed train. Each scenario consists of 25 cases corresponding to input delays from 0 to 1440 s with a step of 60 s. The scenarios tested in class I are summarised in the table below.

Table 5-3 Summary of single train initiated delay scenarios

Index	**Train delayed on entry**	**Input delay range**
I_S2	S2	0, 60, 120, …, 1440 s (equivalent to 0, 1, 2, …, 24 min)
I_S8	S8	
I_S9	S9	
I_S13	S13	
I_S16	S16	
I_S22	S22	

Class II. Multiple entry delays

In class II, we delay two individual services at their entry to the simulation. The scenarios are numbered as M_Si_Sj, where Si, Sj states the delay service. In each scenario, 156 cases were tested under different dispatching rules which delay Si from 0 to 1440s with a step of 120 s and Sj from 60 to 1380s with a step of 120 s. The scenarios tested in class II are shown in the table below.

Table 5-4 Summary of Multiple train initiated delay scenarios

Index	**Train delayed on entry**	**Input delay range**
M_S51_S2	S51, S2	For the first train: 0, 120, …, 1440 s (equivalent to 0, 2, …, 24 min) For the second train: 60, 180, …, 1140 s (equivalent to 1, 3, …, 19 min)
M_S52_S22	S52, S22	
M_S8_S13	S8, S13	
M_S1_S8	S1, S8	
M_S28_S4	S28, S4	
M_S16_S52	S16, S52	

Class III. Minor perturbation delays

In this class each service at each scheduled stopping point has the potential for a delay to be added on departure. The delay is added to the later of:

a. the scheduled departure time and

b. the arrival time + the required dwell time.

The values of delays are drawn from the Weibull distribution which is stated in the equation below.

$$f(x;\lambda,k) = \begin{cases} 0, & x < 0 \\ \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, & x \geq 0 \end{cases} \qquad (5-7)$$

where λ and k state the scale and shape parameters, respectively. We developed three kinds of case with different parameter settings, as shown in the table below. The scenarios are numbered as P_n, where n is counted from 1.

Table 5-5 Summary of minor perturbation delay scenarios

Index	**Trains delayed**	**Delay distribution**
P_1 to P_10	All trains potentially delayed on departure	$k = 0.45, \lambda = 4$
P_11 to P_20		$k = 0.5, \lambda = 6$
P_21 to P_30		$k = 0.55, \lambda = 8$

5.5.4 Computational experiments and discussion

Is this chapter, the final evaluation scheme is decided as the total value of the objective functions in five junction dispatching agents. The data setting in each local dispatching agent is shown as follows.

- Two-point crossover and extend-line mutation are chosen in the proposed GA for MAS in large scale railway network;
- Train cancelling and re-platforming are not considered in this case study;
- $t_{slot} = 30$ min;
- $\widehat{N}_{gen} = 100$;
- $P_{sel} = 0.8$;
- $N_{ter} = 50$.

To show the advantages of the proposed MAS approach, the FCFS approach is also illustrated as a baseline to show the improvement using the MAS approach in terms of

objective function. Here, an example scenario (M_S28_S4) is detailed and discussed below. In the table below, each value states the TD in seconds across the whole network; numbers in black stand for the decision made by FCFS and MAS approaches being same in terms of TD, and in the scenario with two values the numbers in red and green stand for MAS and FCFS, respectively.

Table 5-6 Summary of scenarios M_S28_S4

	S4_0 060s	**S4_0 180s**	**S4_0 300s**	**S4_0 420s**	**S4_0 540s**	**S4_0 660s**	**S4_0 780s**	**S4_0 900s**	**S4_1 020s**	**S4_1 140s**	**S4_1 260s**	**S4_1 380s**
S28_ 0000s	60	180	1458/ 1836	1836	1842	1897	2145	2535	2735	3027	3206	3516
S28_ 0120s	180	300	1583/ 1956	1956	1962	2017	2265	2655	2855	3147	3326	3636
S28_ 0240s	303	423	1706/ 2079	2079	2085	2140	2388	2778	2978	3270	3449	3759
S28_ 0360s	1085	1205	2382/ 2861	2861	2867	2922	3170	3560	3760	4052	4231	4541
S28_ 0480s	2022	2142	3377/ 3824	3824	3825	3859	4107	4495	4697	4989	5168	5478
S28_ 0600s	3009	3129	4292/ 4830	4830	4839	4922	5094	5482	5684	5976	6155	6465
S28_ 0720s	4338	4338	5255/ 5613	5613	5619	5674	5922	6312	6512	6804	6983	7293
S28_ 0840s	3837	4338	5256/ 5613	5613	5619	5674	5922	6312	6512	6804	6983	7293
S28_ 0960s	3837	4020	5255/ 5613	5613	5619	5674	5922	6312	6512	6804	6983	7293
S28_ 1080s	5083/ 5225	5203/ 5345	7254/ 7618	6997/ 7127	6997/ 7127	7006/ 7127	7181/ 7310	7483/ 7649	7654/ 7822	8263	8371	8681
S28_ 1200s	5083/ 5225	5203/ 5345	7079/ 7618	7352/ 7618	6997/ 7127	7006/ 7127	7181/ 7310	7483/ 7649	7654/ 7822	8105/ 8263	8190/ 8371	8681
S28_ 1320s	5083/ 5793	5203/ 5913	7080/ 7618	7352/ 7618	7396/ 7611	6993/ 7798	7166/ 7878	7484/ 8217	7746/ 8213	8128/ 8775	8240/ 9090	8555/ 9249
S28_ 1440s	5203/ 5801	5323/ 5921	7074/ 7604	7352/ 7604	7406/ 7629	7633/ 7864	7286/ 7886	7604/ 8225	7866/ 8246	8262/ 8783	8360/ 9098	8675/ 9257

To illustrate the difference between FCFS and MAS approaches, we take the case of S28 with a 1320 s delay and S4 with a 300 s delay as an example, and simulation experiments for other scenarios are shown in Appendix C. S28 and S4 are long-distance services with a single stop in Derby station which start from Long Eaton station and go to Willington and Clay Cross North Junction, respectively. S6 is local service to Matlock station from Long Eaton station with dwelling at Spondon, Derby, Duffield, Belper, Ambergate,

Whatstandwell, Cronford, Matlock Bath and Matlock stations. S6, S4 and S28 enter the simulation at 7:38:00, 7:39:00 and 7:43:00, respectively; due to their same trajectory between Long Eaton and Derby stations, the movement of S6 blocks the movement of S4 and S28. S6 has a dwell at Spondon station, which means S4 and S28 have to wait outside Spondon station and extend their delays. As they approach Derby station, S6, S4 and S28 will dwell at platforms 6, 4 and 5, respectively. After their dwelling at Derby station, S6 and S4 are dispatched to the Ambergate Junction side at 7:52:00 and 7:53:30, respectively. However, S6 and S4 will ask for the same infrastructure between Derby station and Ambergate Junction, so if the Derby Station North Junction makes the decision according to FCFS, dispatching S6 before S4, the future movement of S4 will be blocked by S6 and its dwelling at Duffield and Belper stations and make the delay of S4 in Ambergate junction is extremely huge, though FCFS has a smaller value for its local objective function. In the MAS approach, the Derby Station North Junction launches S4 first followed by S6. Besides that, S24 is also a long-distance service having the same trajectory as S28 but with a dwell at Derby station at platform 3, and S12 is a high-speed train which enters the simulation from Derby station and is dispatched to the Long Eaton station side from the same platform as S28. In the context of FCFS, the train will be dispatched by Derby London Road Junction in the sequence S24, S28, S12; that decision will increase the delay of S28 and attach a larger knock-on delay to S12. However, the MAS decides to launch S28 first, with the sequence S28, S24, S12. Then, the delay of S28 will be smaller than under FCFS, and the delay to S24 caused by its conflict with S28 will be easily absorbed by the dwell time in Derby station. The accumulated delay and detailed passing sequence are shown as bar charts in Figure 5-5 and in the table below ('_1' and '_2' in the legend stand for the first and second times a train passes the junction).

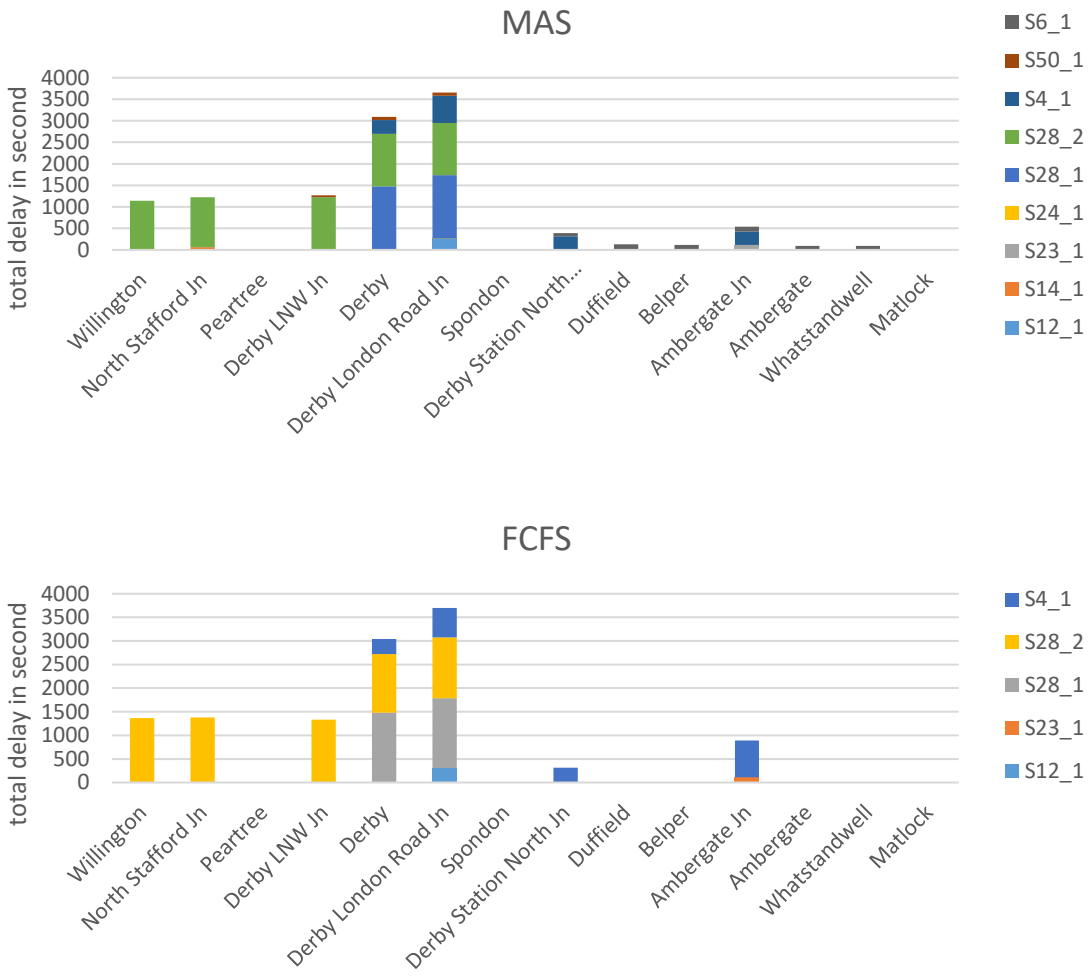


Figure 5-5 Accumulated delay at each junction area and station under FCFS (upper) and MAS (lower) approaches

Passing sequences of trains in each junction in FCFS

The order for **North Strafford Junction** is: S5, S1, S51, S2, S19, S13, S22, S23, S15, S52, S50, S18, S14, S28, S31, S24, S35, S41, S37, S42, S33, S36, S40, S43, S32, S44, S54;

The order for **Derby LNW Junction** is: S5, S1, S51, S2, S13, S19, S22, S15, S23, S52, S14, S50, S28, S18, S24, S31, S41, S35, S37, S42, S33, S43, S36, S40, S44, S32, S54;

The order for **Derby London Road Junction** is: S53, S51, S47, S5, S49, S9, S51, S5, S57, S47, S2, S25, S49, S11, S10, S19, S15, S19, S6, S4, S52, S28, S24, S28, S12, S50, S26, S24, S50, S55, S16, S29, S42, S35, S42, S35, S38, S43, S36, S39, S30, S36, S44, S45,

The order for **Derby Station North Junction** is: S53, S58, S9, S47, S53, S49, S47, S8, S25, S57, S27, S57,S49, S52, S6, S4, S26, S16, S29, S55, S43, S38, S30, S39, S45;

The order for **Ambergate Junction** is: S58, S9, S8, S25, S13, S1, S14, S27, S52, S26, S22, S16, S6, S29, S4, S23, S41, S18, S55, S43, S30, S37, S45, S38, S39, S40.

Passing sequences of trains in each junction in MAS

The order for **North Strafford Junction** is: S5, S1, S51, S2, S19, S13, S22, S23, S15, S52, S50, S18, S28, S14, S31, S24, S35, S41, S37, S42, S33, S36, S40, S43, S32, S44, S54;

The order for **Derby LNW Junction** is: S5, S1, S51, S2, S13, S19, S22, S15, S23, S52, S28, S14, S50, S18, S24, S31, S41, S35, S37, S42, S33, S43, S36, S40, S44, S32, S54;

The order for **Derby London Road Junction** is: S53, S51, S47, S5, S49, S9, S51, S5, S57, S47, S2, S25, S49, S11, S10, S19, S15, S19, S6, S4, S52, S28, S28, S24, S12, S50, S26, S24, S50, S55, S16, S29, S42, S35, S42, S35, S38, S43, S36, S39, S30, S36, S44, S45;

The order for **Derby Station North Junction** is: S53, S58, S9, S47, S53, S49, S47, S8, S25, S57, S27, S49, S57, S52, S4, S6, S26, S16, S29, S55, S43, S38, S30, S39, S45;

The order for **Ambergate Junction** is: S58, S9, S8, S25, S13, S1, S14, S27, S52, S26, S22, S16, S4, S29, S6, S23, S41, S18, S55, S43, S30, S37, S45, S38, S39, S40.

In brief, 11.8% and 20.64% of the solutions made by MAS are better than by FCFS in class I and class II, respectively. The improvement based on FCFS in terms of the values of the objective function is up to 34.11%, 28.7% and 21.27% in class I, class II and class III, respectively. The detailed values of the objective function in class III are shown in the table below.

Table 5-7 Objective function values in class III

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_10
FCFS	3096	2668	1447	2935	1312	1825	1312	1436	3883	1756
MAS	2974	2298	1305	2798	1187	1792	1187	1306	3789	1626
	P_11	**P_12**	**P_13**	**P_14**	**P_15**	**P_16**	**P_17**	**P_18**	**P_19**	**P_20**
FCFS	1709	1900	2049	1514	1866	1269	1599	1626	1389	1089
MAS	1565	1756	1947	1373	1735	1126	1550	1487	1255	941
	P_21	**P_22**	**P_23**	**P_24**	**P_25**	**P_26**	**P_27**	**P_28**	**P_29**	**P_30**
FCFS	1297	686	896	928	672	921	695	1121	906	670
MAS	1155	555	766	791	529	787	553	977	762	597

The computational time in the collaborative agent and each junction agent is illustrated in the figure below.

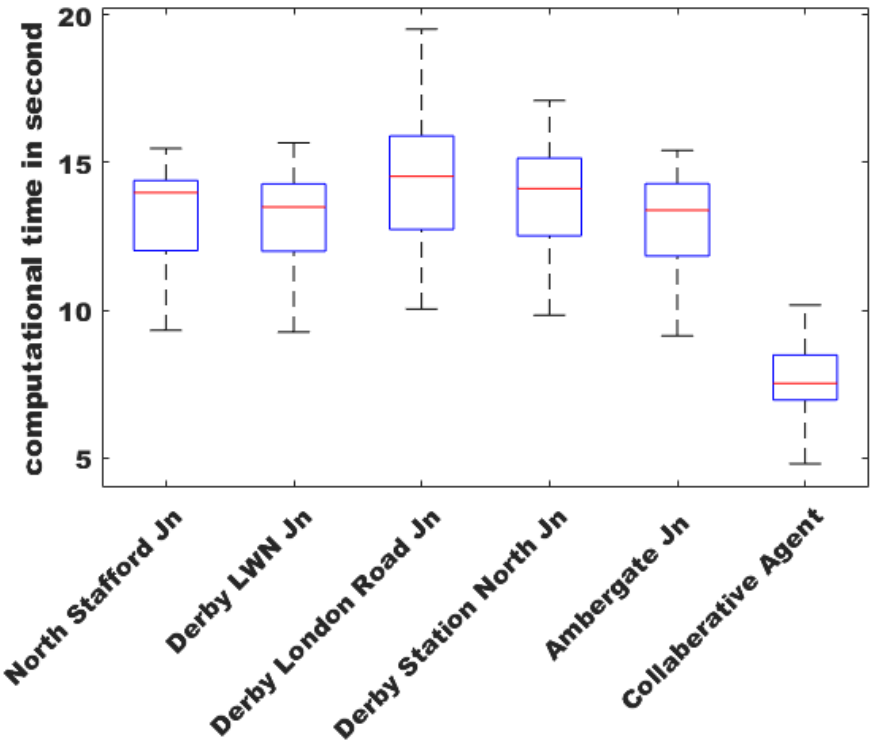


Figure 5-6 Computational time of each agent in the MAS system

In summary, with the decomposition approach proposed in the thesis, searching space of the considered train rescheduling problem is departed into reasonable size which ensures the proposed algorithm can generate an optimised solution within the time horizon of real-time control with up to 34.11% improvement in terms of designed objective function with respect to FCFS. And the multi-layered structure of the MAS improves the scalability of the decision support system, and the voting variables within the system are organised indicators with binary values, which avoid massive data exchange among every side in the system. The mean values of computational times for the junction agents and collaborative agent are 13.4314 and 8.1979 s, respectively. Within the simulation, the local searching in each junction agent is processed in series and finally ends with the collaboration in the collaborative agent; the total computational time in the simulation is up to 97.32 s. However, the local searching in each junction agent can be processed in parallel and the computational time can be 29.64 s. From author's computation experience, solving train rescheduling problems in large scale railway networks are not possible with centralised approach; for example, to solve same scenario in this case study with CPLEX optimiser, the computational time can be several hours, let alone verifying the optimality with Brute Force. Namely, the MAS approach can significantly computational time for the entire system and ensure the generated solutions are better than human dispatching solutions (FCFS). The key benefit from the MAS system is that the collaborative strategy can provide a global view to each junction agent and trade-off among the solutions generated by each junction agent. As a result, an over-greedy solution generated in a simple junction can be corrected by the results from other agents due to its poor preference in other agent areas. The collaborative agent is designed to check the global feasibility of the solutions from each junction agent and find an optimal solution based on the solution tables provided by the junction agents under its control area.

5.6 Summary

In this chapter, the applications of MAS to traffic management problems are reviewed, and an MAS methodology for train rescheduling problems for a large-scale railway network is proposed. The MAS for large-scale railway networks is organised with a hierarchical structure with a two-pass data exchange approach. Two kinds of agents (junction agent and collaborative agent) are designed in the MAS. GA is still chosen as the local searching algorithm, and a Condorcet voting-based collaborative strategy is detailed to achieve a globally feasible solution. The proposed system is tested under a serial of scenarios which are designed based on practical disturbances in railway network. The single train delay and multiple train delays can test the MAS's response to delay cases at the entrance nodes of the simulated infrastructure; and the minor perturbations can test the MAS's response to original delay at station areas led by changes of passenger volume. The simulation results show that the MAS can tackle a train rescheduling problem better than the FCFS routing approach in 19.264% of cases and generate the same solutions as FCFS in the other cases. The solutions from MAS represent improvements of up to 34.11% according to the control objective. The computational time in the simulation is around 1.5 min and potential to be reduced to around 0.5 min in practice which is able to satisfy real-time railway traffic management.

The proposed approach provides a distributed system to solve train rescheduling problems rather than a traditional centralised approach. Due to the decentralised structure of the proposed system, searching space of the local train rescheduling problems for a local junction agent is reduced into suitable size for real-time traffic control and optimisation results are improved due to the fine-tuned problem size. The voting mechanism within the

proposed MAS ensures better scalability based on succinct data exchange within the MAS system, which can extend the system for networks with more junctions.

Chapter 6
CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This thesis provides a comprehensive overview on the content of railway traffic management and our approach to tackling train rescheduling problems in large-scale railway networks. The proposed MAS brings us a step forward to solving large-scale railway traffic management problems with a decentralised system. The thesis proposed a series of innovations including an MILP formulation for train rescheduling problems in a large-scale railway network, and a systemic MAS to tackle the train rescheduling problem with different sizes of network. For railway bottlenecks, a P2P structured MAS is applied to tackle the train rescheduling problem in bottleneck sections, and a hierarchical structured MAS is applied to solve the train rescheduling problem across a large-scale railway network. Moreover, different sizes of rail networks are used to test the proposed MAS, and pros and cons are discussed with respect to computational experiments.

Firstly, an overview of railway operation and timetabling was provided as well as the main functions of railway traffic management systems. The actions generated by railway traffic management systems are classified into three types: real-time rescheduling, delay management and disruption management. Recent research was illustrated with different modelling approaches, and the main challenges to tackling large-scale train rescheduling problems were summarised.

Secondly, the proposed MILP formulation describes the objective of real-time train rescheduling and its related constraints under optimisation. The strong coupling of train timetables among junctions is emphasised in the MILP formulation. Moreover, to apply

the MAS into a railway traffic management system, the railway operational circumstance is investigated, and a decomposition approach to divide the railway network is developed to achieve the MAS in a railway traffic management system.

Thirdly, train rescheduling problems in railway bottlenecks usually include a huge searching space and strong coupling between the two junctions. So, to solve the train rescheduling problem in railway bottleneck sections, a P2P structured MAS is proposed together with a specific developed GA for local searching. Furthermore, a series of negotiation strategies are proposed to help each junction agent to find a globally feasible solution. The P2P system was tested in a bottleneck section in the UK under three kinds of delay scenarios; and the computational experiments show that the P2P system has over 98% likelihood of getting a final solution which is better than the centralised approach in terms of WAD value, and the average computational time of the P2P system is smaller than that of the centralised system which is usually around 10 s in the test scenarios.

Finally, a systematic MAS is proposed to tackle train rescheduling problems in a large-scale railway network. The GA is also chosen as the local searching algorithm, with a new classification operator. A Condorcet voting-based collaborative strategy is proposed; it is achieved in the collaborative agent, and the collaborative strategy ensures the final solutions for each junction agent are feasible globally. The proposed MAS was tested in a UK infrastructure which includes Derby station and its surrounding areas under three classes of delay scenarios. The dispatching solutions generated by MAS were compared with those using an FCFS dispatching rule. The computational results show that there are 11.8% and 20.64% of the solutions generated from MAS is better than FCFS in class I and class II, respectively; and their improvement in terms of their values on objective function are 34.11% and 28.7%, respectively. And in class III, the improvement can be up to 21.27%. Totally, the MAS can generate better solutions in 19.264% of the test cases

against FCFS. According to the computational time of MAS for considered simulation, the mean value of computational time of junction agent and collaborative agent is 13.4313 s and 8.1979 s, respectively; and with a serial execution in simulation, the computational time for making dispatching solution of the whole network is around 1.5 min. However, author discussed it is potentially to reduce computational time of the system to around 0.5 min as junction agents can generate their solutions in parallel in practice.

In a word, MAS is a promising approach for tackling optimisation problems which cover a large-scale geographical area, and the MAS proposed in this thesis provides a systematic approach to tackle train rescheduling problems with a distributed structured system which gives consideration to global feasibility and optimality across the whole railway network. The MAS is able to be embedded into the current railway traffic management system due to its distributed structure, and with the consideration of parallel computing in practice, the computational time of the MAS can be reduced within further computational techniques. To practical control engineering, this thesis detailed describes the collaborative approach for train rescheduling problem within an MAS which defines agent behaviours and their input/output data flow in an MAS.

6.2 Future work

The MAS system proposed in this thesis still has some drawbacks to which the author would like to make further contributions in future work. The plan on future work is illustrated in this section.

The objective of the proposed approach in this thesis is minimising the delay in large-scale railway network. However, practice railway operation usually has other objectives, such as: minimising passenger waiting, minimising energy consumption, etc. So the searching algorithm is expected to optimise more control objectives in author's future research.

The searching algorithm (GA) is able to find an optimised solution in a reasonable computational time. However, the computational time of GA cannot satisfy the real-time control horizon according to industry level (5 s in the ICONIS system). As a result, improvements to the local searching algorithm are desirable. Furthermore, it is hard to reduce the computational time to 5 s (almost 1/3 of computational time in the simulation in this thesis) by heuristic approaches or by applying a new searching algorithm. Based on the author's experience on algorithm development, parallel computing strategies are a promising approach to reduce the computational time. So, applying parallel computing strategies is inevitable in future research.

Besides that, GA is inherently a meta-heuristic algorithm, not an exact algorithm. Namely, the GA may output different solutions for a considered problem with the same date setting. Though it rarely happened in the case studies of this thesis, it may raise robustness issues in practice system. To ensure the robustness of the decision-making process, improvement approaches to GA and more searching algorithms can be studied, which can improve the performance of the searching algorithm in proposed MAS. Moreover, the modification operator in the proposed GA may have some over greedy issues in local searching of a junction agent; the searching agent may converge to a sub-optimal solution due to the operational constraints are ensured strictly in modification operator, which impacts the diversity of a generation and degrades final solution after collaborative among all the junction agents.

Furthermore, this thesis only solve the real-time rescheduling problems with train rescheduling actions without considering rerouting and re-platforming actions. However, rerouting and re-platforming actions are also common methods to minimise the effects of train delays, so improving the proposed MAS with consideration of rerouting and re-platforming actions is also promising.

Finally, rolling stock rescheduling and crew member rescheduling are also attractive topics in rail traffic research. And sometime, rolling stock and crew member rescheduling problems may also have overlaps with train rescheduling problems. So, developing a systematic MAS covering train rescheduling, rolling stock rescheduling and crew member rescheduling is an ambitious research topic for both academia and industry.

Appendix A: Train timetables around Derby station

Train ID / Stations& Junctions	S1	S2	S4	S5	S6	S8
	1E38	1K00	1F02	1D50	2A14	2A09
Clay Mill Jn Enter/Exit	07:00:00	…	…	06:53:00	…	…
TH Enter/Exit	…	07:13:30	…	…	…	…
Willington	…	…	…	…	…	…
North Stafford Jn	07:02:02	07:15:00	…	06:55:20	…	…
Peartree	…	**07:20:00**	…	…	…	…
Derby LNW Jn	07:05:40	07:20:56	…	06:58:44	…	…
Long Eaton Enter/Exit	…	…	07:34:00	07:21:00	07:38:00	…
Spondon	…	…	07:38:30	**07:14:00**	**07:45:00**	…
Derby London Road Jn	…	…	07:38:49	07:09:23	07:45:18	…
Derby	**07:13:00**	**07:30:57**	**07:48:00**	**07:09:00**	**07:52:00**	**07:39:57**
Derby Station North Jn	…	…	07:48:18	…	07:52:17	07:13:25
Chaddesden Curve	…	…	…	…	…	…
Duffield	…	…	…	…	**07:59:30**	**07:07:30**
Belper	…	…	…	…	**08:04:30**	**07:02:30**
BroadHolme	…	…	…	…	…	…
Ambergate Jn	07:21:00	…	07:55:50	…	08:07:30	06:55:37
Ambergate	…	…	…	…	**08:10:30**	**06:55:00**
Whatstandwell	…	…	…	…	**08:14:31**	…
Cronford	…	…	…	…	**08:20:00**	…
Matlock Bath	…	…	…	…	**08:22:30**	…
Matlock	…	…	…	…	**08:26:00**	…
Chesterfield Enter/Exit	07:26:00	…	08:06:00	…	…	…

Train ID / Stations& Junctions	S9	S10	S11	S12	S13	S14
	1C15	1C92	2A89	2A91	1V44	5V43
Clay Mill Jn Enter/Exit	…	…	…	…	07:34:30	08:06:30
TH Enter/Exit	…	…	…	…	…	…
Willington	…	…	…	…	…	…
North Stafford Jn	…	…	…	…	07:31:00	08:01:48
Peartree	…	...	…	…	…	…
Derby LNW Jn	…	…	…	…	07:28:17	07:58:21
Long Eaton Enter/Exit	07:14:00	07:45:00	07:41:00	08:30:00	…	…
Spondon	07:09:00	07:40:00	07:35:30	07:57:30	…	…
Derby London Road Jn	07:05:19	07:36:25	07:31:26	07:53:30	…	…
Derby	**07:05:00**	**07:36:00**	**07:31:00**	**07:53:00**	**07:27:00**	**07:57:00**
Derby Station North Jn	06:59:55	…	…	…	…	…
Chaddesden Curve	…	…	…	…	…	…
Duffield	…	…	…	…	…	…
Belper	…	…	…	…	…	…
BroadHolme	…	…	…	…	…	**07:44:00**
Ambergate Jn	06:54:00	…	…	…	07:14:44	07:30:20
Ambergate	…	…	…	…	…	…
Whatstandwell	…	…	…	…	…	…
Cronford	…	…	…	…	…	…
Matlock Bath	…	…	…	…	…	…
Matlock	…	…	…	…	…	…
Chesterfield Enter/Exit	06:49:00	…	…	…	07:11:00	07:26:30

Train ID / Stations& Junctions	S15	S16	S18	S19	S22	S23
	1K03	2N19	1E42	1D15	1S33	4F01
Clay Mill Jn Enter/Exit	…	…	07:57:00	07:27:00	07:33:30	07:41:30
TH Enter/Exit	07:51:00	…	…	…	…	…
Willington	…	…	…	**07:30:30**	07:35:53	…
North Stafford Jn	07:49:30	…	07:59:23	07:33:00	07:37:30	07:45:00
Peartree	**07:43:30**	…	…	…	…	…
Derby LNW Jn	07:41:27	…	08:02:40	07:36:08	07:39:13	07:48:38
Long Eaton Enter/Exit	…	08:25:30	…	07:55:00	…	…
Spondon	…	**08:18:30**	…	**07:48:00**	…	…
Derby London Road Jn	…	08:13:27	…	07:43:22	…	…
Derby	**07:40:00**	**08:13:00**	**08:13:30**	**07:43:00**	**07:44:00**	**07:56:00**
Derby Station North Jn	…	08:10:10	…	…	…	…
Chaddesden Curve	…	…	…	…	…	…
Duffield	…	**08:04:00**	…	…	…	…
Belper	…	**07:59:00**	…	…	…	…
BroadHolme	…	…	…	…	…	…
Ambergate Jn	…	07:53:09	08:21:30	…	07:52:00	08:10:55
Ambergate	…	**07:52:30**	…	…	…	…
Whatstandwell	…	**07:46:30**	…	…	…	…
Cronford	…	**07:41:30**	…	…	…	…
Matlock Bath	…	**07:38:30**	…	…	…	…
Matlock	…	**07:35:30**	…	…	…	…
Chesterfield Enter/Exit	…	…	08:26:30	…	07:57:00	08:24:30

Train ID / Stations& Junctions	S24	S25	S26	S27	S28	S29
	1G09	1C17	1C20	1F01	1V04	1C22
Clay Mill Jn Enter/Exit	08:14:30	…	…	…	07:48:00	…
TH Enter/Exit	…	…	…	…	…	…
Willington	…	…	…	…	**07:44:00**	…
North Stafford Jn	08:11:40	…	…	…	07:41:40	…
Peartree	…	…	…	…	…	…
Derby LNW Jn	08:07:21	…	…	…	07:37:22	…
Long Eaton Enter/Exit	07:52:00	07:32:00	08:11:00	…	07:21:00	08:30:00
Spondon	07:56:30	07:26:00	08:05:30	…	07:25:00	08:23:30
Derby London Road Jn	07:57:40	07:22:24	08:01:31	…	07:26:03	08:20:00
Derby	**08:06:00**	**07:22:00**	**08:01:00**	**07:21:00**	**07:36:00**	**08:19:30**
Derby Station North Jn	…	07:16:00	07:57:22	07:21:25	…	08:15:20
Chaddesden Curve	…	…	…	…	…	…
Duffield	…	…	…	…	…	…
Belper	…	…	…	07:29:30	…	…
BroadHolme	…	…	…	…	…	…
Ambergate Jn	…	07:08:43	07:50:30	07:31:44	…	08:08:14
Ambergate	…	…	…	…	…	…
Whatstandwell	…	…	…	…	…	…
Cronford	…	…	…	…	…	…
Matlock Bath	…	…	…	…	…	…
Matlock	…	…	…	…	…	…
Chesterfield Enter/Exit	…	07:04:00	07:46:30	07:37:00	…	08:04:30

Train ID / Stations& Junctions	S30	S31	S32	S33	S35	S36
	1C25	1K02	1K04	1K05	1D53	1D54
Clay Mill Jn Enter/Exit	…	…	…	…	08:23:30	08:50:30
TH Enter/Exit	…	08:05:00	09:13:00	08:51:00	…	…
Willington	…	…	…	…	**08:27:00**	…
North Stafford Jn	…	08:06:00	09:14:25	08:48:12	08:28:14	08:52:53
Peartree	…	…	…	…	…	…
Derby LNW Jn	…	08:10:13	09:18:50	08:43:00	08:32:40	08:56:14
Long Eaton Enter/Exit	09:12:00	…	…	…	08:49:00	09:17:00
Spondon	09:05:00	…	…	…	08:44:00	09:12:00
Derby London Road Jn	09:01:23	…	…	…	08:40:25	09:08:23
Derby	**09:01:00**	**08:41:57**	**09:41:57**	**08:42:00**	**08:40:00**	**09:08:00**
Derby Station North Jn	08:56:30	…	…	…	…	…
Chaddesden Curve	…	…	…	…	…	…
Duffield	…	…	…	…	…	…
Belper	…	…	…	…	…	…
BroadHolme	…	…	…	…	…	…
Ambergate Jn	08:48:30	…	…	…	…	…
Ambergate	…	…	…	…	…	…
Whatstandwell	…	…	…	…	…	…
Cronford	…	…	…	…	…	…
Matlock Bath	…	…	…	…	…	…
Matlock	…	…	…	…	…	…
Chesterfield Enter/Exit	08:44:30	…	…	…	…	…

Train ID / Stations& Junctions	S37	S38	S39	S40	S41	S42
	1S35	2A18	1F08	1F79	1V46	1V05
Clay Mill Jn Enter/Exit	08:32:00	…	…	08:54:30	08:36:00	…
TH Enter/Exit	…	…	…	…	…	…
Willington	…	…	…	…	…	08:46:00
North Stafford Jn	08:34:24	…	…	08:56:53	08:32:44	08:42:41
Peartree	…	…	…	…	…	…
Derby LNW Jn	08:37:43	…	…	09:00:14	08:29:17	08:38:20
Long Eaton Enter/Exit	…	08:37:00	08:52:30	…	…	08:23:00
Spondon	…	**08:44:30**	08:57:00	…	…	08:26:30
Derby London Road Jn	…	08:47:45	08:57:20	…	…	08:28:00
Derby	**08:44:00**	**08:52:00**	**09:05:00**	**09:16:00**	**08:28:00**	**08:37:00**
Derby Station North Jn	…	08:52:18	09:05:18	…	…	…
Chaddesden Curve	…	…	…	…	…	…
Duffield	…	08:59:30	…	…	…	…
Belper	…	09:04:30	…	…	…	…
BroadHolme	…	…	…	…	…	…
Ambergate Jn	08:51:27	09:07:27	09:12:53	09:23:00	08:15:35	…
Ambergate	…	**09:10:30**	…	…	…	…
Whatstandwell	…	**09:14:31**	…	…	…	…
Cronford	…	**09:20:00**	…	…	…	…
Matlock Bath	…	**09:22:30**	…	…	…	…
Matlock	…	**09:26:00**	…	…	…	…
Chesterfield Enter/Exit	08:57:00	…	09:22:00	09:30:00	08:12:00	…

Train ID / Stations& Junctions	S43	S44	S45	S47	S49	S50
	1V83	1G12	2N23	5C27	5B21	1D52
Clay Mill Jn Enter/Exit	09:01:00	09:20:00	…	…	…	07:53:30
TH Enter/Exit	…	…	…	…	…	…
Willington	…	…	…	…	…	…
North Stafford Jn	08:58:20	09:16:39	…	…	…	07:55:53
Peartree	…	…	…	…	…	…
Derby LNW Jn	08:54:44	09:12:23	…	…	…	07:59:48
Long Eaton Enter/Exit	…	08:56:00	09:24:00	07:24:00	07:36:00	08:19:00
Spondon	…	**09:02:30**	09:18:30	07:18:00	07:31:00	08:14:00
Derby London Road Jn	…	09:05:35	09:14:27	06:57:51	07:03:54 07:27:22	08:10:00
Derby	**08:53:00**	**09:11:00**	**09:14:00**	**07:00:00** **07:14:30**	**07:07:00** **07:27:00**	**08:10:00**
Derby Station North Jn	08:49:00	…	09:09:40	07:01:05 07:11:09	07:23:35	…
Chaddesden Curve	…	…	…	**07:09:00**	…	…
Duffield	…	…	**09:03:30**	…	…	…
Belper	…	…	**08:58:30**	…	…	…
BroadHolme	…	…	…	…	…	…
Ambergate Jn	08:41:00	…	08:53:07	…	…	…
Ambergate	…	…	**08:52:30**	…	…	…
Whatstandwell	…	…	**08:46:30**	…	…	…
Cronford	…	…	**08:41:30**	…	…	…
Matlock Bath	…	…	**08:38:30**	…	…	…
Matlock	…	…	**08:35:30**	…	…	…
Chesterfield Enter/Exit	08:37:00	…	…	…	…	…

Train ID / Stations& Junctions	S51	S52	S53	S54	S55	S57
	1O06	1O82	5F01	1K07	1F05	5C92
Clay Mill Jn Enter/Exit	07:14:00	07:57:30	…	…	…	…
TH Enter/Exit	…	…	…	09:51:00	…	…
Willington	…	…	…	…	…	…
North Stafford Jn	07:10:35	07:54:49	…	09:48:10	…	…
Peartree	…	…	…	…	…	…
Derby LNW Jn	07:07:06	07:51:14	…	09:43:19	…	…
Long Eaton Enter/Exit	06:49:00	…	…	…	08:07:30	…
Spondon	06:53:00	…	…	…	08:12:00	…
Derby London Road Jn	06:53:47	…	06:45:53	…	08:12:19	07:14:56
Derby	**07:06:00**	**07:50:00**	**06:48:00** **07:20:00**	**09:42:00**	**08:19:00**	**07:17:00** **07:35:57**
Derby Station North Jn	…	07:45:10	06:47:52 07:02:46	…	08:19:20	07:17:57 07:29:11
Chaddesden Curve	…	…	**07:00:00**	…	…	**07:26:00**
Duffield	…	…	…	…	…	…
Belper	…	…	…	…	…	…
BroadHolme	…	…	…	…	…	…
Ambergate Jn	…	07:38:04	…	…	08:27:35	…
Ambergate	…	…	…	…	…	…
Whatstandwell	…	…	…	…	…	…
Cronford	…	…	…	…	…	…
Matlock Bath	…	…	…	…	…	…
Matlock	…	…	…	…	…	…
Chesterfield Enter/Exit	…	07:34:00	…	…	08:32:00	…

Train ID / Stations& Junctions	S58
	2A13
Clay Mill Jn Enter/Exit	…
TH Enter/Exit	…
Willington	…
North Stafford Jn	…
Peartree	…
Derby LNW Jn	…
Long Eaton Enter/Exit	…
Spondon	…
Derby London Road Jn	…
Derby	**06:58:00**
Derby Station North Jn	06:53:15
Chaddesden Curve	…
Duffield	**06:47:00**
Belper	**06:42:30**
BroadHolme	…
Ambergate Jn	**06:37:07**
Ambergate	**06:36:30**
Whatstandwell	**06:30:30**
Cronford	**06:25:30**
Matlock Bath	**06:22:30**
Matlock	**06:19:30**
Chesterfield Enter/Exit	…

Appendix B: Single train simulator

A single train simulator is used to predict train passing times at a junction area based on a railway kinematic model. In this section, we introduce the single train simulator with an overview of the train traction system and show how the single train simulator predicts train passing times at junction areas.

I. Overview of Train Traction

Based on Newton's Second Law, the acceleration or deceleration of a train can be explained by the following equation:

$$Acceleration = \frac{total\ external\ forces}{effective\ mass}$$

$$= \frac{tractive\ effort - train\ resistance - Mg * \sin\alpha}{effective\ mass}$$

Train resistance: Davis equation

Train resistance is a friction of various sorts, which consists by principally of bearing friction and aerodynamic drag. Bearing friction includes a constant which is proportional to weight, and aerodynamic drag is a variable which is roughly proportional to squared speed. The Davis equation can be stated as follows:

$$drag\ force = a + bv + cv^2$$

where *a*,*b* and *c* stand for Davis coefficients for a particular train in particular conditions.

Gradient

Trains may be operated on a slope other than a horizontal level. Here, we assume an angle α between the vertical gravitational force Mg and the horizontal, so Mg can be decomposed into $Mg \sin\alpha$ along the track and $Mg \cos\alpha$ perpendicular to the track.

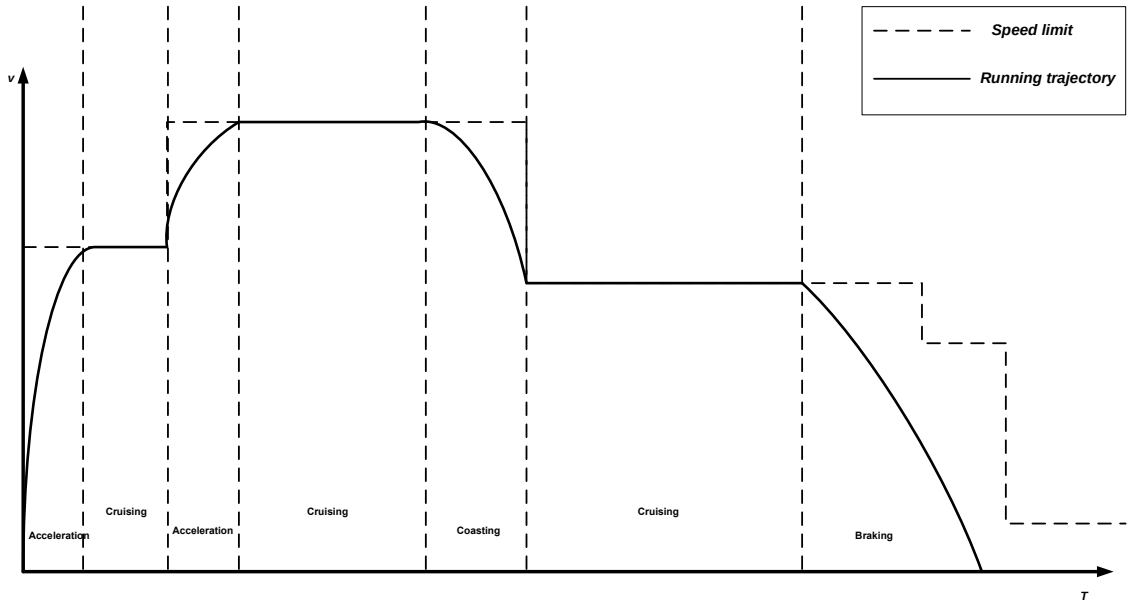
Effective mass

The effective mass of a train corresponds to three parts, the tare mass of the train, passenger or freight load and rotary allowance. The rotary allowance is generated by the rotational sense with the acceleration of the train and may vary from 5% to 15% with different numbers of motored axels. The effective mass is shown in the following equation:

$$effective\ mass = actual\ tare\ mass * (1 + rotary\ allowance) + passenger/freight\ load$$

II. Train Movement Simulation

The movement of a train can be seen as four types of moving section: acceleration, cruising, coasting and braking. A typical movement of a train is shown in the following figure. In the single train simulator, we assume the train is operated under the largest permitted speed (speed limit). To simplify the simulation, the coasting behaviour, passenger or freight load and track gradient are ignored. As a result, the train movement is viewed as a series of sections of acceleration, cruising and braking.



Acceleration

Because the drag force corresponds to train running speed, it will lead to great error against that seen in practice. In the simulator, the acceleration section of train movement is replaced by an acceleration homogeneous strip modelling approach. The train speed is increased with a step of 1 km/h, thus the initial speed and final speed are known. For example, acceleration of a train from 0 to 100 km/h is decomposed into 100 fractions, and the running distance and time covered by acceleration are calculated by the sum of distance and time in each fraction. The following equations show the calculation of running distance d_a and time t_a in an acceleration section:

$$d_a(v_0, v_t) = \sum_{i=v_0}^{v_t} \frac{i^2 - (i-1)^2}{\frac{maximum\ tractive\ effort - (a + b * \frac{i + (i-1)}{2} + c * (\frac{i + (i-1)}{2})^2)}{actual\ tare\ mass * (1 + rotary\ allowance)}}$$

$$t_a(v_0, v_t) = \sum_{i=v_0}^{v_t} \frac{1}{\frac{maximum\ tractive\ effort - (a + b * \frac{i + (i-1)}{2} + c * (\frac{i + (i-1)}{2})^2)}{actual\ tare\ mass * (1 + rotary\ allowance)}}$$

Cruising

Cruising is a section where the train is operated at constant speed (usually at the speed limit). As the train is operated at constant speed, the resistance to the train is also constant. The running distance and time can be calculated by Newton's formula:

$$S = v * t$$

where *S* is the running distance along railway track, and *v* is the running speed (speed limit).

Braking

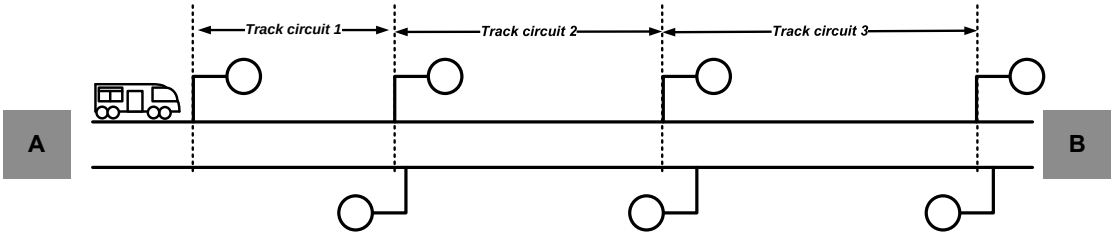
In the braking section, a constant braking rate b is applied, so the braking distance and braking time are calculated as follows:

$$d_b(v_0, v_t) = \frac{v_0 - v_t}{b}$$

$$t_b(v_0, v_t) = \frac{{v_0}^2 - {v_t}^2}{2b}$$

III. Estimation of Junction Passing Times of Trains

To estimate the passing time of a train at junction area, the journey of a train is divided into several parts which are identified by the track circuit. The following example shows a train travelling from point A to point B via track circuits 1, 2 and 3.



At the beginning of the single train simulator, a table of historical data called *Dynamic Data* which records the clearance time of every track circuit is saved (at the beginning, the dynamic data is empty). A track circuit is occupied by a train from when the head of the train enters the track circuit until the tail of the train leaves the track circuit. So, the travel time $t_T^{l_\tau + l_{tc}}$ of a train in a track circuit is calculated by the train movement model covering $l_\tau + l_{tc}$, where l_T is the length of the train, and l_{tc} is the length of the track circuit. The clearance time of a track circuit is evaluated by the occupation of the most

recent train and is saved as $t_\tau^{tc,start} + t_\tau^{l_\tau + l_{tc}}$, where $t_\tau^{tc,start}$ is the time at which train τ arrives at track circuit *tc*.

The single train simulator will predict the train passing times at a considered junction area with the passing sequence provided by the optimising algorithm. The prediction includes the following steps:

1) As a train is considered in the simulator, the trajectory of the train to the target junction is considered, and the clearance times of track circuits in the trajectory are taken out from the dynamic data.

2) In the acceleration section, the movement of a train is considered as a series of acceleration with a step of 1km/h. In the cruising section, the movement of a train is considered with a series of movements at in 0.01 km steps.

3) The braking distance $d_{b,\tau}$ and braking time $t_{b,\tau}$ are calculated from the update of the speed limit along the trajectory of train τ.

4) The simulator will calculate the movement of the train with iterations of a single step in its movement. After each step, the current position (distance from the start point) and current time will be updated. And the simulator will judge its future behaviour from the braking distance and braking time against the clearance time of its future track circuit.

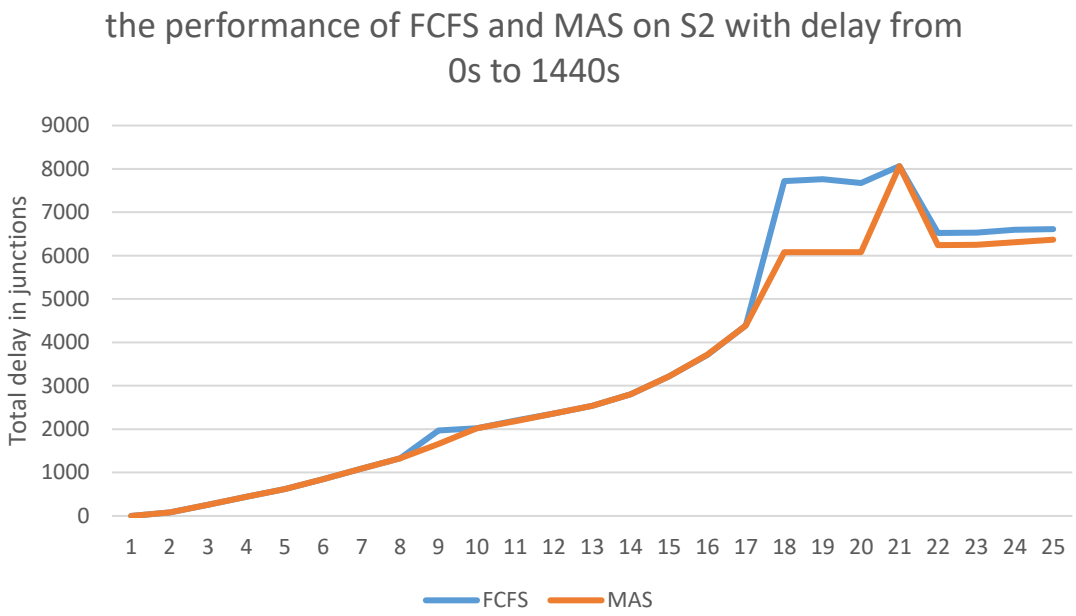
5) If a train stops to wait for the clearance of its future track circuit, the train will wait until the clearance time of the track circuit, and the current time in the simulator will be set to the clearance time of the track circuit.

The passing time of a train in the junction area is finally calculated from the full distance from the start point to the area covered by the junction, and the estimated passing time is calculated by the start time plus all the time spent on the journey to the junction.

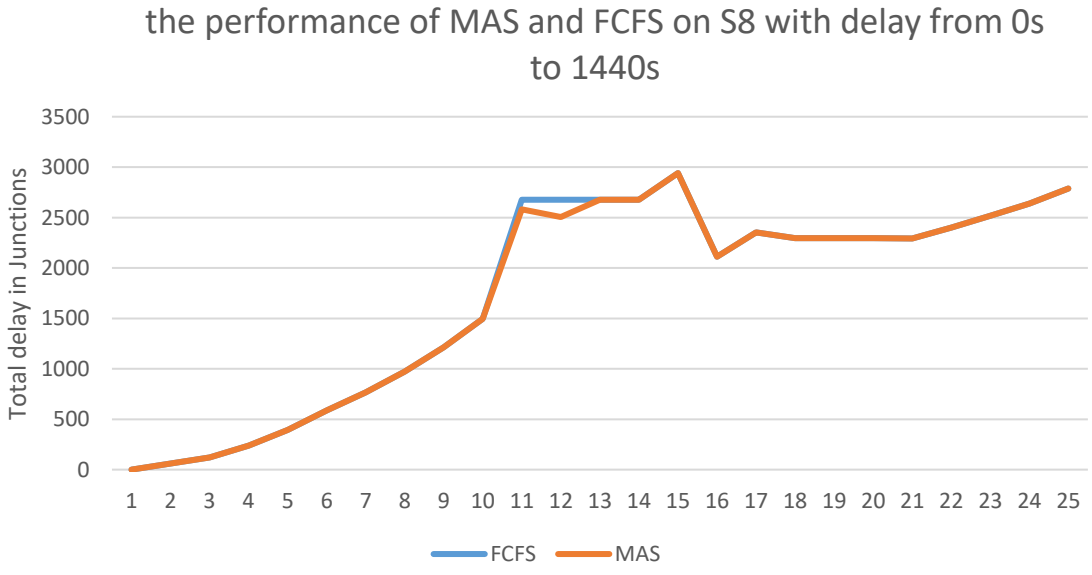
Appendix C: Computational experiments of the case study in Chapter 5

Class I:

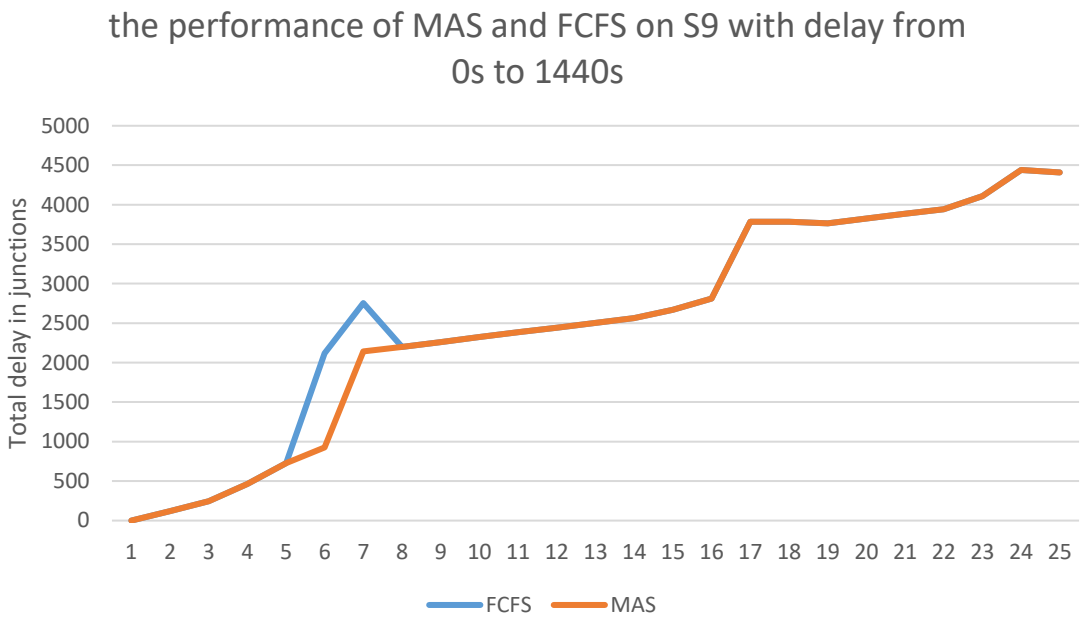
I_S2:



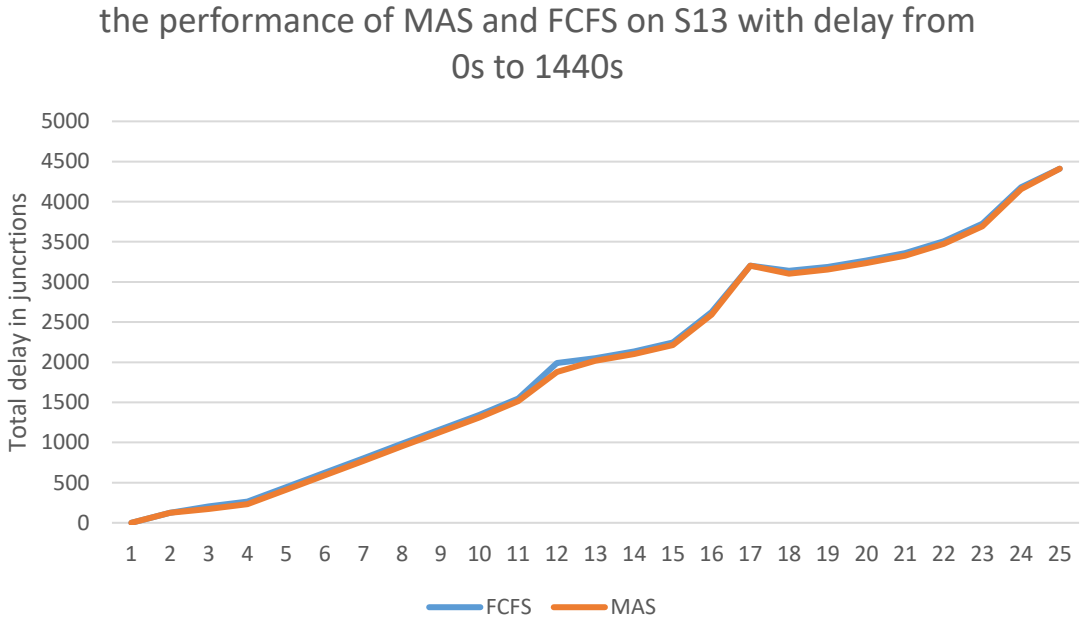
I_S8:



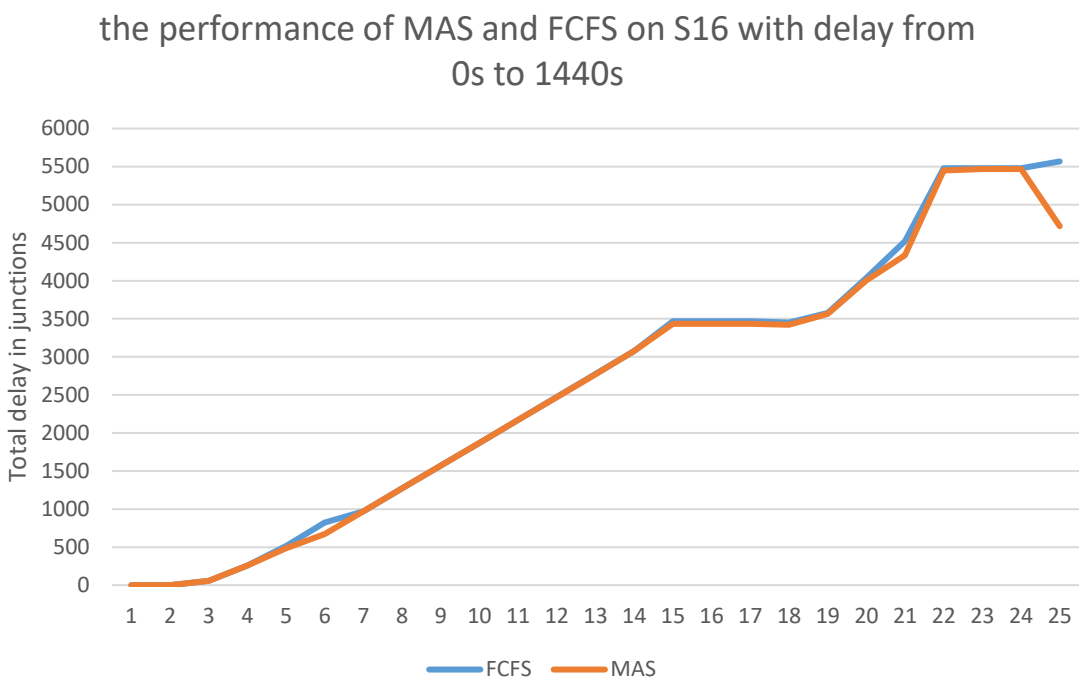
I_S9:


the performance of MAS and FCFS on S9 with delay from 0s to 1440s
Total delay in junctions
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
FCFS
MAS

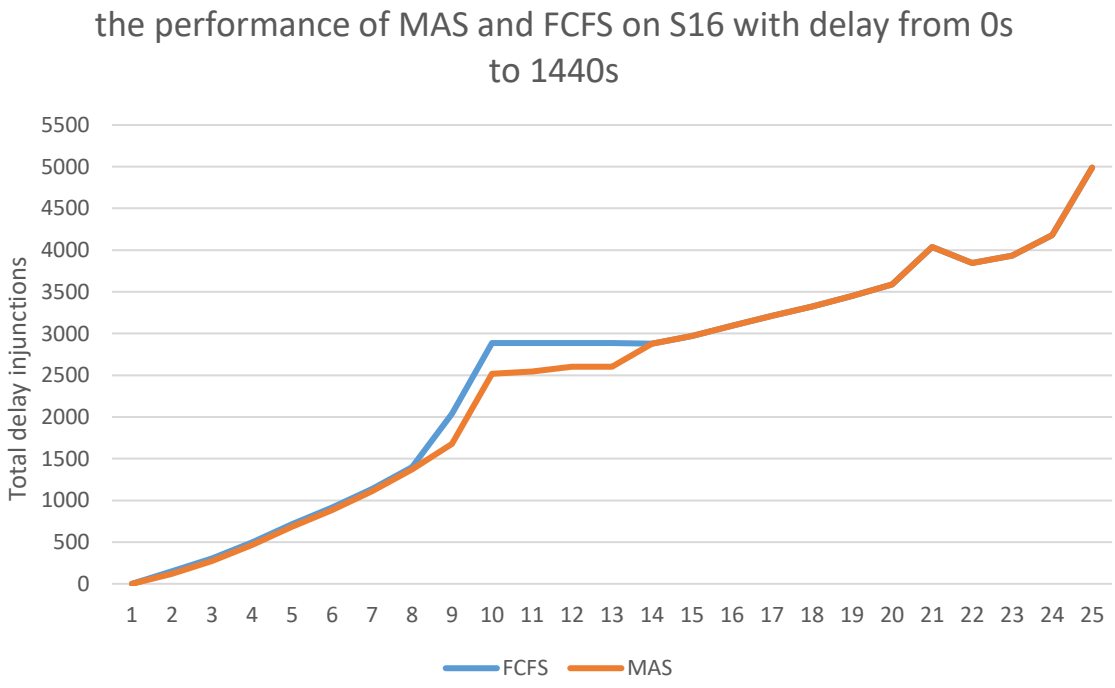
I_S13:


the performance of MAS and FCFS on S13 with delay from 0s to 1440s
Total delay in juncrtions
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
FCFS
MAS

I_S16


the performance of MAS and FCFS on S16 with delay from 0s to 1440s
Total delay in junctions
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
FCFS
MAS

I_S22


the performance of MAS and FCFS on S16 with delay from 0s to 1440s
Total delay injunctions
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
FCFS
MAS

Class II:

M_S51_S2

	S2_0060s	S2_0180s	S2_0300s	S2_0420s	S2_0540s	S2_0660s	S2_0780s	S2_0900s	S2_1020s	S2_1140s	S2_1260s	S2_1380s
S51_0000s	80	440	849	1329	2024	2359	2802	3713	6082/7722	6082/7677	6241/6524	6239/6504
S51_0120s	200	560	969	1449	2144	2479	2922	3833	6155/7842	6202/7797	6361/6644	6348/6624
S51_0240s	320	680	1089	1569	2264	2599	3042	3953	6322/7962	6322/7917	6481/6764	6515/6744
S51_0360s	440	800	1209	1689	2384	2719	3162	4073	6422/8082	6398/8037	6601/6884	6635/6864
S51_0480s	875	1235	1644	2124	2819	3154	3597	4508	6829/8517	6877/8472	7020/7319	7070/7299
S51_0600s	1355	1715	2124	2604	3299	3634	4077	4988	7341/8997	7357/8952	7516/7799	7550/7779
S51_0720s	1835	2195	2604	3084	3779	4114	4557	5468	7789/9477	7837/9432	7996/8279	8030/8259
S51_0840s	2315	2675	3084	3564	4259	4594	5037	5948	8289/9957	8317/9912	8476/8759	8466/8739
S51_0960s	2872	3232	3641	4121	4816	5151	5594	6505	8831/10514	8874/10469	9033/9316	9019/9296
S51_1080s	3275	3635	4044	4524	5219	5554	5997	6908	9257/10917	9277/10872	9396/9719	9470/9699
S51_1200s	3755	4115	4524	5004	5699	6034	6477	7388	9710/11397	9730/11352	9868/10199	9950/10179
S51_1320s	4235	4595	5004	5484	6179	6514	6957	7868	10329/11877	10237/11832	10396/10679	10383/10659
S51_1440s	4715	5075	5484	5964	6659	6994	7437	8348	10689/12357	10707/12312	10836/11159	10910/11139

M_S52_S22

	S22_ 0060s	**S22_ 0180s**	**S22_ 0300s**	**S22_ 0420s**	**S22_ 0540s**	**S22_ 0660s**	**S22_ 0780s**	**S22_ 0900s**	**S22_ 1020s**	**S22_ 1140s**	**S22_ 1260s**	**S22_ 1380s**
S52_ 0000 s	120	464	881	1366	2518/ 2855	2604/ 2855	2845	3060	3300	3566	3814	4148
S52_ 0120 s	360	704	1121	1606	2758/ 3095	2844/ 3095	3085	3300	3540	3806	4054	4388
S52_ 0240 s	1228	1572	1929	2451	3626/ 3963	3940/ 3963	3953	4168	4408	4674	4922	5256
S52_ 0360 s	1618	1962	2610	2864	4315/ 4388	4137/ 4388	4378	4593	4833	5099	5346	5651
S52_ 0480 s	2400	2744	3161	3646	4798/ 5135	4884/ 5135	5125	5340	5580	5858	6085	6428
S52_ 0600 s	3249	3593	4010	4495	5725/ 6069	5818/ 6069	6059	6274	6514	6760	7209	7336
S52_ 0720 s	4345/ 4639	4677/ 4971	5411	5597/ 5885	7037/ 7374	7056/ 7374	7077/ 7364	7261/ 7579	7795/ 7819	8051/ 8075	8297/ 8333	8667
S52_ 0840 s	5428/ 5811	5765/ 6155	6174/ 6572	6650/ 7057	7819/ 8546	7880/ 8546	8105/ 8536	8345/ 8751	8560/ 8991	8816/ 9257	9062/ 9505	9839
S52_ 0960 s	6177	6509	6989	7487	8814/ 8912	8900/ 8912	8902	9117	9357	9623	9871	1020 5
S52_ 1080 s	6707	7039	7568	8016	9105/ 9442	9191/ 9442	9432	9647	9887	1015 3	1040 1	1073 5
S52_ 1200 s	6993	7337	7754	8239	9728	9728	9718	9933	1017 3	1043 9	1068 7	1102 1
S52_ 1320 s	7113	7457	7874	8359	9804/ 9848	9575/ 9848	9838	1005 3	1029 3	1055 9	1080 7	1114 1
S52_ 1440 s	7233	7577	7994	8479	9924/ 9968	9717/ 9968	9958	1017 3	1041 3	1067 9	1092 7	1126 1

M_S8_S13

	S13_ 0060s	**S13_ 0180s**	**S13_ 0300s**	**S13_ 0420s**	**S13_ 0540s**	**S13_ 0660s**	**S13_ 0780s**	**S13_ 0900s**	**S13_ 1020s**	**S13_ 1140s**	**S13_ 1260s**	**S13_ 1380s**
S8_0 000s	126	232	592	952	1312	1894/ 1957	2032/ 2101	2593	3105	3235	3472	4058
S8_0 120s	246	352	712	1072	1432	2014/ 2077	2152/ 2221	2713	3225	3355	3592	4178
S8_0 240s	519	625	985	1345	1705	2287/ 2350	2425/ 2494	2986	3498	3628	3865	4451
S8_0 360s	893	999	1359	1719	2079	2661/ 2724	2868	3360	3872	4002	4239	4825
S8_0 480s	1339	1445	1805	2165	2525	3107/ 3170	3314	3806	4318	4448	4685	5271
S8_0 600s	2130/ 2735	2250/ 2855	2486/ 2975	2846/ 3085	3206/ 3445	3998/ 4090	3926/ 4234	4487/ 4726	4997/ 5238	5129/ 5368	5568/ 5605	6191
S8_0 720s	2735	2855	2975	3085	3445	4090	4234	4726	5238	5368	5605	6191
S8_0 840s	3001	3121	3241	3341	3647	4292	4436	4928	5440	5570	5807	6393
S8_0 960s	2413	2533	2653	2751	3111	3693/ 3756	3831/ 3900	4392	4904	5034	5271	5857
S8_1 080s	2541	3013	3133	3253	3351	3933/ 3996	4071/ 4140	4632	5144	5274	5511	6097
S8_1 200s	2541	2889	3591/ 3876	3996	4053/ 4116	4236	4311/ 4380	4872	5384	5514	5751	6337
S8_1 320s	2645	2889	3501	4087/ 4272	4333/ 4392	4512	4551/ 4620	5112	5624	5754	5991	6577
S8_1 440s	2915	3021	3501	4087	4931	5051	5171	5382	5894	6024	6261	6847

M_S1_S8

	S8_0 060s	**S8_0 180s**	**S8_0 300s**	**S8_0 420s**	**S8_0 540s**	**S8_06 60s**	**S8_07 80s**	**S8_0 900s**	**S8_10 20s**	**S8_11 40s**	**S8_1 260s**	**S8_1 380s**
S1_0 000s	60	239	587	973	1495	2295/ 2400	2295/ 2675	2113	2295/ 2436	2295/ 3219	2399	2639
S1_0 120s	300	479	827	1213	1735	2535/ 2640	2535/ 2915	2353	2535/ 2676	2535/ 3459	2639	2879
S1_0 240s	542	721	1069	1455	1977	2777/ 2882	2777/ 3157	2595	2777/ 2918	2777/ 3701	2881	3121
S1_0 360s	902	1081	1429	1815	2337	3137/ 3242	3137/ 3517	2955	3137/ 3278	3137/ 4061	3241	3481
S1_0 480s	1262	1441	1789	2175	2697	3497/ 3602	3497/ 3877	3315	3497/ 3638	3497/ 4421	3601	3841
S1_0 600s	1622	1801	2149	2535	3129	3873/ 3962	3873/ 4237	4155	3873/ 3998	3873/ 4781	3961	4201
S1_0 720s	2182	2361	2709	3095	3675	4417/ 4588	4417/ 4724	4463	4417/ 4786	4417/ 5569	4521	4761
S1_0 840s	2679	2858	3206	3592	4114	4914/ 5117	4914/ 5540	4732	4914/ 5055	4914/ 5838	5018	5258
S1_0 960s	2789	2968	3316	3702	4224	5024/ 5232	5024/ 5655	4842	5024/ 5165	5024/ 5948	5128	5368
S1_1 080s	3062	3241	3589	3975	4497	5297/ 5407	5297/ 5851	5115	5297/ 5438	5297/ 6221	5401	5641
S1_1 200s	3422	3601	3949	4335	4857	5657/ 5762	5657/ 6085	5475	5657/ 5798	5657/ 6581	5761	6001
S1_1 320s	3782	3961	4309	4695	5217	6017/ 6122	6017/ 6397	5835	6017/ 6158	6017/ 6941	6121	6361
S1_1 440s	4142	4321	4669	5055	5579	6377/ 6482	6377/ 6757	6195	6377	6377	6481	6721

M_S16_S52

	S52_0060s	**S52_0180s**	**S52_0300s**	**S52_0420s**	**S52_0540s**	**S52_0660s**	**S52_0780s**	**S52_0900s**	**S52_1020s**	**S52_1140s**	**S52_1260s**	**S52_1380s**
S16_0000s	120	578	2151	1984	2604	3666	5736	5798	6356	6813	6933	7053
S16_0120s	178	636	2209	2042	2662	3724	5778	5809	6356	6947	6991	7111
S16_0240s	605	1054	2571	2469	3089	4151	6205	6236	6356	6947	7667	8082
S16_0360s	1091	1549	3122	2955	3575	4637	6691	6722	6842	6962	7667	8669
S16_0480s	1691	2149	3722	3555	4175	5237	7291	7322	7442	7562	7682	8669
S16_0600s	2291	2749	4322	4155	4775	5837	7891	7922	8042	8162	8282	8684
S16_0720s	2891	3349	4922	4755	5375	6437	8491	8522	8642	8762	8882	9284
S16_0840s	3554	4012	5443/5863	5418	5985	6848	9154	9185	9305	9425	9545	9947
S16_0960s	3554	4012	5443/5863	5418	5985	6848	9154	9185	9305	9425	9545	9947
S16_1080s	3685	4143	5559/6038	5549	6121	6957	9285	9316	9436	9556	9676	10078
S16_1200s	4554	5012	6825/7027	6418	6990	7813	10154	10185	10305	10425	10545	10947
S16_1320s	5468/5568	5177/6026	7660/8495	6583/7432	7178/8004	8827	11168	11199	11319	11439	11559	11961
S16_1440s	4719/5568	5177/6026	7660/8495	7432	8004	8827	11168	11199	11319	11439	11559	11961

Appendix D: Author's publications

LIU, J., CHEN, L., ROBERTS, C., LI, Z. & WEN, T. A Multi-agent Based Approach for Railway Traffic Management Problems. 2018 International Conference on Intelligent Rail Transportation (ICIRT), 12–14 December. 2018, 1–5.

LIU, J., CHEN, L., ROBERTS, C., NICHOLSON, G. & AI, B. 2019a. Algorithm and Peer-to-peer Negotiation Strategies for Train Dispatching Problems in Railway Bottleneck Sections. *IET Intelligent Transport Systems* [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-its.2019.0020. (Accepted 05/08/2019)

References

AARDAL, K., NEMHAUSER, G. L. & WEISMANTEL, R. 2005. *Discrete optimization* Amsterdam; Boston: Elsevier.

ABBINK, E. 2014. *Crew Management in Passenger Rail Transport.* Ph.D. thesis, Erasmus Research Institute of Management.

AGOGINO, A. K. & TUMER, K. 2012. A multiagent approach to managing air traffic flow. *Autonomous Agents and Multi-Agent Systems,* 24**,** 1–25.

ALBRECHT, A. R., PANTON, D. M. & LEE, D. H. 2013. Rescheduling rail networks with maintenance disruptions using Problem Space Search. *Computers & Operations Research,* 40**,** 703–712.

ALBRECHT, T., HANSEN, I. A. & PACHL, J. 2008a. Railway timetable & traffic : analysis, modelling, simulation. *In:* HANSEN, I. A. & PACHL, J. (eds.) *Railway timetable and traffic.* Hamburg : Eurailpress.

ALMEJALLI, K., DAHAL, K. & HOSSAIN, A. An intelligent multi-agent approach for road traffic management systems. 2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC), 8–10 July 2009, 825–830.

ALSTOM. 2015. ICONIS TMS [Online]. Available: http://kajt.org/onewebmedia/ICONIS%20TMS-%20KAJT%20nov%202015.pdf [Accessed 17/06/2019].

ARNOLD-FORSTER, H. O. 1884. Proportional representation : a dialogue In: PROPORTIONAL REPRESENTATION, S. & ELECTORAL REFORM, S. (eds.). London : published by the Proportional Representation Society.

BARBER, F., INGOLOTTI, L., LOVA, A., TORMOS, P. & SALIDO, M. A. 2009. Meta-heuristic and Constraint-Based Approaches for Single-Line Railway Timetabling.

In: AHUJA, R. K., MÖHRING, R. H. & ZAROLIAGIS, C. D. (eds.) *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems.* Berlin; Heidelberg: Springer.

BARTA, J., RIZZOLI, A. E., SALANI, M. & GAMBARDELLA, L. M. Statistical modelling of delays in a rail freight transportation network. Proceedings of the 2012 Winter Simulation Conference (WSC), 9–12 Dec. 2012, 1–12.

BASET, S. & SCHULZRINNE, H. 2010. Reliability and relay selection in peer-to-peer communication systems. Proceedings of the Fourth International Conference on Principles, Systems and Applications of IP Telecommunications, IPTComm 2010, Munich, Germany, 2–4 August 2010, 111–121.

BŁAŻEWICZ, J., DOMSCHKE, W. & PESCH, E. 1996. The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research,* 93**,** 1–33.

BOCHARNIKOV, Y. V., TOBIAS, A. M., ROBERTS, C., HILLMANSEN, S. & GOODMAN, C. J. 2007. Optimal driving strategy for traction energy saving on DC suburban railways. *IET Electric Power Applications*, 1, 675–682. doi: 10.1049/iet-epa_20070005

BREUSEGEM, V. V., CAMPION, G. & BASTIN, G. 1991. Traffic modeling and state feedback control for metro lines. *IEEE Transactions on Automatic Control,* 36**,** 770–784.

BUFORD, J. F. & YU, H. 2010. Peer-to-Peer Networking and Applications: Synopsis and Research Directions. *In:* SHEN, X., YU, H., BUFORD, J. & AKON, M. (eds.) *Handbook of Peer-to-Peer Networking.* Boston, MA: Springer US.

CACCHIANI, V., HUISMAN, D., KIDD, M., KROON, L., TOTH, P., VEELENTURF, L. & WAGENAAR, J. 2014. An overview of recovery models and algorithms for

real-time railway rescheduling. *Transportation Research Part B: Methodological,* 63**,** 15–37.

CAPACITY FOR RAIL. 2015. *Review of Existing Practice to Improve Capacity on the Railway Network*. Capacity for Rail.

CHEN, L., ROBERTS, C., SCHMID, F. & STEWART, E. 2015. Modeling and Solving Real-Time Train Rescheduling Problems in Railway Bottleneck Sections. *IEEE Transactions on Intelligent Transportation Systems,* 16**,** 1896–1904.

CHEN, L., SCHMID, F., DASIGI, M., NING, B., ROBERTS, C. & TANG, T. 2010. Real-Time Train Rescheduling in Junction Areas. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit,* 224**,** 547–557.

CHIKARA, H., TOMII, N., YOSHIAKI, T., SHIGEKI, K. & ATSUSHI, F. 2007. *Train rescheduling algorithm using pattern description language and integrated algorithm framework*. Proceedings of The Eleventh IASTED International Conference on Artificial Intelligence and Soft Computing, ASC '07, Palma de Mallorca, Spain, 29–31 August 2007, 250–255.

CORMAN, F., D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. 2009. Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies,* 17**,** 607–616.

CORMAN, F., D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. 2010a. Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport,* 2**,** 219–247.

CORMAN, F., D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. 2010b. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological,* 44**,** 175–192.

CORMAN, F., D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. 2012. Optimal inter-area coordination of train rescheduling decisions. *Transportation Research Part E: Logistics and Transportation Review,* 48**,** 71–88.

CORMAN, F. & MENG, L. 2015. A Review of Online Dynamic Models and Algorithms for Railway Traffic Management. *IEEE Transactions on Intelligent Transportation Systems,* 16**,** 1274–1284.

DAI, L. 2016. *Intelligent Real-Time Train Rescheduling Management for Railway System*. Ph.D. thesis, University of Birmingham.

DANTZIG, G. 1948. Programming in a linear structure. *Econometrica*, 17, 73.

D'ARIANO, A., CORMAN, F., PACCIARELLI, D. & PRANZO, M. 2008. Reordering and Local Rerouting Strategies to Manage Train Traffic in Real Time. *Transportation Science,* 42**,** 405–419.

D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. 2007a. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research,* 183**,** 643–657.

D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. 2007b. Conflict Resolution and Train Speed Coordination for Solving Real-Time Timetable Perturbations. *IEEE Transactions on Intelligent Transportation Systems,* 8**,** 208–222.

D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. 2008. Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C: Emerging Technologies,* 16**,** 232–245.

D'ARIANO, A. & PRANZO, M. 2009. An Advanced Real-Time Train Dispatching System for Minimizing the Propagation of Delays in a Dispatching Area Under Severe Disturbances. *Networks and Spatial Economics,* 9**,** 63–84.

DE CARITAT CONDORCET, J.-A.-N. 1989. *The Political Theory of Condorcet*, trans. MCLEAN, I. & SOMMERLAD, F. Oxford: University of Oxford, Social Studies Faculty Centre.

DEMITZ, J., HÜBSCHEN, C. & ALBRECHT, C. 2004. Timetable stability - using simulation to ensure quality in a regular interval timetable. *WIT Transactions on The Built Environment*, 74, 14.

DE SCHUTTER, B., VAN DEN BOOM, T. & HEGYI, A. 2002. Model Predictive Control Approach for Recovery from Delays in Railway Systems. *Transportation Research Record,* 1793**,** 15–20.

DOLLEVOET, T. & HUISMAN, D. 2014. Fast heuristics for delay management with passenger rerouting. *Public Transport,* 6**,** 67–84.

DORFMAN, M. J. & MEDANIC, J. 2004. Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research Part B: Methodological,* 38**,** 81–98.

DÜNDAR, S. & ŞAHIN, İ. 2013. Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. *Transportation Research Part C: Emerging Technologies,* 27**,** 1–15.

EDWARDS, G. S. 2015. Empowering Shareholders, or Overburdening Companies? Analyzing the Potential Use of Instant Runoff Voting in Corporate Elections. 68, 1335-1367.

FAN, B. 2012. *Railway traffic rescheduling approaches to minimise delays in disturbed conditions.* Ph.D. thesis, Univeristy of Birmingham.

FAN, B., ROBERTS, C. & WESTON, P. 2012. A comparison of algorithms for minimising delay costs in disturbed railway traffic scenarios. *Journal of Rail Transport Planning & Management,* 2**,** 23–33.

FANG, W., YANG, S. & YAO, X. 2015. A Survey on Problem Models and Solution Approaches to Rescheduling in Railway Networks. *IEEE Transactions on Intelligent Transportation Systems,* 16**,** 2997–3016.

FARRELL, D. M. 2011. Electoral systems : a comparative introduction Basingstoke, England

New York : Palgrave Macmillan, 2011

FERRAVANTE, R. 2006. ERTMS – DELIVERING FLEXIBLE AND RELIABLE RAIL TRAFFIC. Belgium: European Railway Agency.

FISCHETTI, M. & LODI, A. 2011. Heuristics in mixed integer programming. In: COCHRAN, J. J. (ed.) *Wiley Encyclopedia of Operations Research and Management Science*. Hoboken, NJ: John Wiley & Sons, Inc.

FISCHETTI, M. & MONACI, M. 2017. Using a general-purpose Mixed-Integer Linear Programming solver for the practical solution of real-time train rescheduling. *European Journal of Operational Research,* 263**,** 258–264.

FLETCHER, R. 2013. *Practical Methods of Optimization* [electronic resource]. Hoboken, NJ : Wiley.

FLETT, T. M. 2008. *Differential analysis : differentiation, differential equations, and differential inequalities*. New York: Cambridge University Press.

FLIEGE, J. & SVAITER, B. 2000. Steepest descent methods of multicriteria optimization. *Mathematical Methods of Operations Research*, 51, 479–494.

GEHRLEIN, W. 2002 Condorcet's paradox and the likelihood of its occurrence: different perspectives on balanced preferences. *Theory and decision*, 52, 171–199.

GHAEMI, N., CATS, O. & GOVERDE, R. M. P. 2017. Railway disruption management challenges and possible solution directions. *Public Transport*, 9, 343–364.

GOVERDE, R. M. P. 2010. A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies*, 18, 269–287.

GOWER, T. L. 2019. Passenger and Freight Rail Performance. London: Office of Rail and Road.

GREEN-ARMYTAGE, J. & TIDEMAN, T. N. 2019. Selecting the Runoff Pair. *Public Choice*, 182.

HAMIDI, H. & KAMANKESH, A. 2018. An Approach to Intelligent Traffic Management System Using a Multi-agent System. *International Journal of Intelligent Transportation Systems Research,* 16**,** 112–124.

HERNÁNDEZ, J. Z., OSSOWSKI, S. & GARCíA-SERRANO, A. 2002. Multiagent architectures for intelligent traffic management systems. *Transportation Research Part C: Emerging Technologies,* 10**,** 473–506.

HIRAI, C., TOMII, N., TASHIRO, Y., KONDOU, S. & FUJIMORI, A. 2006. An algorithm for train rescheduling using rescheduling pattern description language R. *WIT Transactions on The Built Environment*, 88, 551–561.

HO, S. C. & HAUGLAND, D. 2011. Local search heuristics for the probabilistic dial-a-ride problem. *OR Spectrum,* 33**,** 961–988.

HO, T. K., NORTON, J. P. & GOODMAN, C. J. 1997. Optimal traffic control at railway junctions. *IEE Proceedings - Electric Power Applications,* 144**,** 140–148.

ISMAIL, A., QUAFAFOU, M. , NACHOUKI, G. & HAJJAR, M. 2010. A decision tree for queries routing in hierarchical peer-to-peer network. *The 7th International Conference on Informatics and Systems (INFOS)*, Cairo, Egypt, 28–30 March 2010.

JOSYULA, S. P., TÖRNQUIST KRASEMANN, J. & LUNDBERG, L. 2018. A parallel algorithm for train rescheduling. *Transportation Research Part C: Emerging Technologies,* 95**,** 545–569.

KECMAN, P. 2014. *Models for Predictive Railway Traffic Management.* Ph.D. thesis, Delft University of Technology.

KECMAN, P., CORMAN, F., D'ARIANO, A. & GOVERDE, R. M. P. 2013. Rescheduling models for railway traffic management in large-scale networks. *Public Transport,* 5**,** 95–123.

KIRKWOOD, D. 2015. *Simulation, test and performance evaluation of railway control strategies and algorithms* Ph.D. thesis, University of Birmingham.

LEW, A. & MAUCH, H. 2007. *Dynamic programming : a computational tool* Berlin: Springer.

LIU, J., CHEN, L., ROBERTS, C., LI, Z. & WEN, T. 2018. A Multi-agent Based Approach for Railway Traffic Management Problems. *International Conference Intelligent Rail Transportation.* Singapore, 12-14 December 2018.

MARCHAND, H., MARTIN, A., WEISMANTEL, R. & WOLSEY, L. 2002. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics,* 123**,** 397–446.

MARINESCU, R. & DECHTER, R. AND/OR branch-and-bound for solving mixed integer linear programming problems. *In:* VAN BEEK P. (ed.) *Principles and Practice of Constraint Programming - CP 2005*. Lecture Notes in Computer Science, vol 3709. Berlin; Heidelberg: Springer, 857.

MASCIS, A. & PACCIARELLI, D. 2002. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research,* 143**,** 498–517.

MASOUD, N., NAM, D., YU, J. & JAYAKRISHNAN, R. 2017. Promoting Peer-to-Peer Ridesharing Services as Transit System Feeders. *Transportation Research Record,* 2650**,** 74–83.

MILLER, N. R. 1998. The Alternative Vote and Coombs Rule versus First-Past-the-Post: A Social Choice Analysis of Simulated Data Based on English Elections. *Public Choice*, 158, 399-426.

MISSIKOFF, M. An object-oriented approach to an information and decision support system for railway traffic control. Proceedings of 1st International Conference on Conventional and Knowledge Based Intelligent Electronic Systems, KES '97, 21-23 May 1997. 633–641.

MURATA, T. 1989. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE,* 77**,** 541–580.

NIELSEN, L. K. 2011. *Rolling Stock Rescheduling in Passenger Railways: Applications in Short-term Planning and in Disruption Management.* Ph.D. thesis, Erasmus University, Rotterdam.

OBATA, T. 2003. A Method for Discriminating Efficient Candidates with Ranked Voting Data. *European Journal of Operational Research*, 151, 233-238.

OETTICH, T. A. S. 2002. A New Integrated Approach To Dynamic Schedule Synchronization And Energy-saving Train Control. *Computers in Railways VIII,* 61**,** 10.

Open ACCESS GOVERNMENT. 2019. *Upgrading the UK's Railway Infrastructure* [Online]. Open Access Government. Available: https://www.openaccessgovernment.org/upgrading-the-uks-railway-infrastructure/45291/ [Accessed 2019].

PACCIARELLI, D. 2013. Introduction and Classification of Existing Rescheduling Literature. Proceedings of *IAROR*, Copenhagen.

PELLEGRINI, P., MARLIÈRE, G., PESENTI, R. & RODRIGUEZ, J. 2015. RECIFE-MILP: An Effective MILP-Based Heuristic for the Real-Time Railway Traffic Management Problem. *IEEE Transactions on Intelligent Transportation Systems,* 16**,** 2609–2619.

PELLEGRINI, P., MARLIÈRE, G. & RODRIGUEZ, J. 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B: Methodological,* 59**,** 58–80.

PELLEGRINI, P., MARLIÈRE, G. & RODRIGUEZ, J. 2016. A detailed analysis of the actual impact of real-time railway traffic management optimization. *Journal of Rail Transport Planning & Management,* 6**,** 13–31.

PELLEGRINI, P., PESENTI, R. & RODRIGUEZ, J. 2019. Efficient train re-routing and rescheduling: Valid inequalities and reformulation of RECIFE-MILP. *Transportation Research Part B: Methodological,* 120**,** 33–48.

PODERYS, J., FAROOQ, J. & SOLER, J. A Novel Multimedia Streaming System for Urban Rail Environments Using Wi-Fi Peer-to-Peer Technology. 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), 3–6 June 2018 2018. 1–6.

RUSSELL, S. J. & NORVIG, P. 2016. *Artificial Intelligence: A Modern Approach, Global Edition*, third edition. Harlow: Pearson, 2016.

SAMÀ, M., PELLEGRINI, P., D'ARIANO, A., RODRIGUEZ, J. & PACCIARELLI, D. 2016. Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B: Methodological,* 85**,** 89–108.

SANDERSON, M. R. & SKELLY, J. 2007. *Macromolecular Crystallography: Conventional and High-Throughput Methods* [electronic resource]. Oxford: Oxford University Press.

SCHACHTEBECK, M. & SCHÖBEL, A. 2010. To wait or not to wait---and who goes first? Delay management with priority decisions. *Transportation Science*, 44, 307–321.

SCHÖBEL, A. 2001. A Model for the Delay Management Problem based on Mixed-Integer-Programming. *Electronic Notes in Theoretical Computer Science,* 50**,** 1–10.

SHAH, S. & SASTRY, P. S. 1999. New algorithms for learning and pruning oblique decision trees. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* 29**,** 494–505.

SHEERAN, R. & KAY, M. 2015. *A Short Guide to Network Rail*. London: National Audit Office.

SINHA, S. M. 2006. *Mathematical programming : theory and methods*. New Delhi: Elsevier.

SMITH, D. K. 1987. Mathematical Programming: Theory and Algorithms. *Journal of the Operational Research Society,* 38**,** 666.

TIDEMAN, N. 1995. The Single Transferable Vote. *Journal of Economic Perspectives*, 9, 27-39.

TÖRNQUIST, J. 2007. Railway traffic disturbance management—An experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. *Transportation Research Part A: Policy and Practice,* 41**,** 249–266.

TÖRNQUIST, J. & PERSSON, J. A. 2007. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological,* 41**,** 342–362.

VAN DE WALLE, S. 2011. The Order of Questions in a Suevey on Citizen Satisfaction with Public Service: Lessons from a Split-Ballot Experience *Public Administration*, 89, 1436-1451.

VAN GOEVERDEN, K. & CORREIA, G. 2018. Potential of peer-to-peer bike sharing for relieving bike parking capacity shortage at train stations: An explorative analysis for the Netherlands. *European Journal of Transport and Infrastructure Research,* 18**,** 457–474.

VEELENTURF, L. P., KIDD, M. P., CACCHIANI, V., KROON, L. G. & TOTH, P. 2016. A Railway Timetable Rescheduling Approach for Handling Large-Scale Disruptions. *Transportation Science,* 50**,** 841–862.

WANG, L., MO, W., QIN, Y., DOU, F. & JIA, L. 2014. Optimization Based High-Speed Railway Train Rescheduling with Speed Restriction. *Discrete Dynamics in Nature and Society,* 2014**,** 14.

WANG, P. & GOVERDE, R. M. P. 2016. Multiple-phase train trajectory optimization with signalling and operational constraints. *Transportation Research Part C: Emerging Technologies,* 69**,** 255–275.

WANG, P., MA, L., GOVERDE, R. M. P. & WANG, Q. 2016. Rescheduling Trains Using Petri Nets and Heuristic Search. *IEEE Transactions on Intelligent Transportation Systems,* 17**,** 726–735.

WEISS, G. (ed.) 1999. Multiagent systems: *A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA;

London: MIT Press.

WESTON, P., PRESTON, J., TAKAGI, R., BOUCH, C. & ARMSTRONG, J. 2006. Minimising train delays in a mixed traffic railway network with consideration of passenger delay. 13th International Conference on Advanced Systems for Public Transport, CASPT 2015.

XIAO, Z., FU, X., ZHANG, L., PONNAMBALAM, L. & GOH, R. S. M. Data-driven multi-agent system for maritime traffic safety management. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 16–19 October 2017. 1–6.

YUAN, J. 2006. *Stochastic modelling of train delays and delay propagation in stations.* Ph.D. thesis, Delft University of Technology.

ZHEN, Q. & JING, S. 2016. Train rescheduling model with train delay and passenger impatience time in urban subway network. *Journal of Advanced Transportation,* 50**,** 1990–2014.