Real Time Traffic Management in Junction Areas and Bottleneck Sections on Mainline Railways

by

LEI CHEN

A thesis submitted to

The University of Birmingham

for the degree of

DOCTOR OF PHILOSOPHY

School of Civil Engineering

College of Engineering and Physical Sciences

The University of Birmingham, UK

May 2012

Abstract

As the demand for passenger transport increases year by year, main line railways in many countries are experiencing ever more intensive use of their services, particularly in urban areas. Very often, the existing infrastructure in such areas is overloaded. On many railways, such sections of the infrastructure are described as bottlenecks and a great deal of effort is devoted to the management of the operations in these areas, to ensure optimum use of the available resources and to minimise disruption to services following minor incidents.

The author of this thesis deals with the issues of real time traffic management in junction areas and bottleneck sections on mainline railways in the event of service disturbances. A systematic methodology is proposed for modelling and solving real time train rescheduling problems in junction areas and bottleneck sections, including train re-sequencing and train re-timing.

Firstly, a formal mathematical model, the Junction Rescheduling Model (JRM) is proposed in this thesis, based on Mixed Integer Programming (MIP) to minimise a Weighted Average Delay (WAD). An innovative algorithm based on Differential Evolution algorithm, named DE_JRM is proposed for solving real time train rescheduling problems formulated with JRM.

The performance of the algorithm DE_JRM has been evaluated with a stochastic method based on Monte-Carlo simulation methodology. The evaluation results show that, for both flyover and flat junctions, under all four proposed train delay distributions, the

WAD can be reduced significantly with the train rescheduling algorithm DE_JRM compared with First Come First Served (FCFS) and a conventional Automatic Route Setting (ARS) strategy, and the average computation time of the algorithm DE_JRM is around 2-3 seconds, which is more than satisfactory for the real time applications of train rescheduling. With the ARS strategy, the statistical WAD cannot be decreased significantly compared with FCFS. This indicates that the application of the ARS strategy cannot bring many benefits to decreasing the WAD in these scenarios. It is also found that, with the application of algorithm DE_JRM, the WAD in flat junction scenarios is even lower than the WAD with FCFS and the ARS strategy in flyover junction scenarios.

The author also extends the proposed methodology, including JRM and the algorithm DE_JRM, to model and solve real time train rescheduling problems for bottleneck sections of railway networks. The simulation results show good performance of the proposed methodology. As for all four train delay distributions, the WAD is decreased significantly with the proposed train rescheduling algorithm DE_JRM compared with FCFS and the ARS strategy.

Finally, an integrated system architecture for the traffic management and train control is introduced for system implementation of the proposed methodology of train rescheduling in junction areas and bottleneck sections on mainline railways.

Keywords: Traffic management, Train rescheduling, Junction Rescheduling Model, Bottleneck sections, DE_JRM, Railway junction.

Acknowledgements

I am very grateful to my supervisors, Prof. Felix Schmid and Prof. Clive Roberts, for their patient, continuous and attentive guidance during my study and research in the UK, for their inspiration and very generous support. I have greatly benefited from their personality, knowledge and enthusiasm in work and life. Without their support and encouragement, my study and research in the UK would not have been possible.

I would like to thank my supervisor Dr. Meena Dasigi in Network Rail, for the generous financial support from Network Rail, and a lot of significant suggestions from her during my research work in Network Rail.

I also would like to thank Prof. Bin Ning, as my supervisor at Beijing Jiaotong University (BJTU). I am very grateful to him for his attentive supervision besides his busy daily work as the President of BJTU. I also would like to give my thanks to Prof. Tao Tang and Prof. Chunhai Gao at BJTU for their kind research supervision during my stay at BJTU.

Many thanks to all the members of the Birmingham Centre for Railway Research and Education for all of their kind help.

As well, I would like to thank all of my friends for their continuous support.

Finally I would like to thank my wife, my parents and my parents-in-law, for their great love, which is my most cherished thing in life.

Contents of Thesis

List of Figures

List of Tables

List of Abbreviations

ARS	Automatic Route Setting
ATC	Automatic Train Control
ATO	Automatic Train Operation
ATP	Automatic Train Protection
BDT	Binary Decision Tree
CBTC	Communication Based Train Control system
CDF	Cumulative Distribution Function
CNP	Constraint Nonlinear Programming
CP	Combinatorial Programming
DAS	Driving Advisory System
DCS	Data Communication System
DE	Differential Evolution
DE_JRM	Differential Evolution for Junction Rescheduling Model
DMI	Driver Machine Interface
DSU	Data Storage Unit
EA	Evolutionary Algorithm
EDF	Empirical Distribution Function
ERTMS	European Railway Traffic Management System

ETCS	European Train Control System
FCFS	First Come First Served
FPESP	Flexible Periodic Event Scheduling Problem
GA	Genetic Algorithm
GSM-R	Global System for Mobile Communications-Railway
IP	Integer Programming
JRM	Junction Rescheduling Model
KKT	Karush–Kuhn–Tucker conditions
KO2	Key Output 2
LP	Linear Programming
MIP	Mixed Integer Programming
MP	Mathematical Programming
MRM	Multi-Resolution Modelling
MTR	Mass Transit Railway
MTS	Multi Train Simulator
NP-hard	Nondeterministic Polynomial-hard
PERT	Program Evaluation and Review Technique
PESP	Periodic Event Scheduling Problem
SA	Simulated Annealing

SIL	Safety Integration Level
SOSC	Second Order Sufficient Conditions
SQP	Sequential Quadratic Programming
STS	Single Train Simulator
SWAD	Statistical Weighted Average Delay
TM	Traffic Management
TPH	Trains Per Hour
TS	Tabu Search
VOBC	Vehicle On-Board Controller
WAD	Weighted Average Delay

Chapter 1. Introduction

1.1 Research Background and Problem Statement

1.1.1 Background

As demand for passenger transport increases world-wide, main line railways in many countries are experiencing ever more intensive use of their services, particularly in urban areas. Very often, the existing infrastructure in such areas is overloaded. However, the construction of new railways in urbanised contexts is expensive and is often faced with insurmountable obstacles, e.g. lack of space and the presence of listed buildings. On many railways, such sections of the infrastructure are described as junctions or bottlenecks and a great deal of effort is devoted to the management of the operations in these areas to ensure optimum use of the available capacity and to minimise disruption to services following minor incidents.

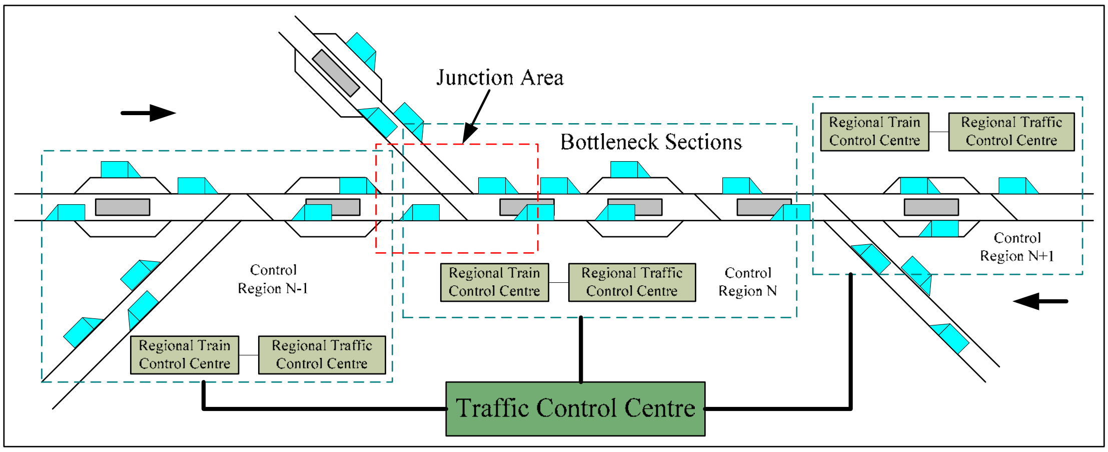

Figure 1-1 Urban Main Line Railway with a Generic Bottleneck Section and its Approaches

On main line railways, bottleneck sections are often at the heart of networks, between junctions where different services converge from a range of origins or diverge to a variety of destinations. A typical urban railway configuration, with a bottleneck section and the associated approach tracks, is shown in Figure 1-1. Well known examples include Lines A and B/D of the RER network in Paris and the planned Thameslink and CrossRail networks in London, as well as the subsurface lines of London Underground.

For such high density networks, train service intervals through the bottleneck section are comparable to those of metro type railways. By contrast, most metro operations are controlled to achieve a particular headway, e.g. 3 minutes, rather than to satisfy a particular timetable. Metro passengers normally board the first train arriving at the platform since all trains travel to the same destination on most modern metro railways.

Because of disruptions exist in railway operations, train delays usually occur. In terms of the generation sources, train delays can be catalogued into two main kinds: original delays and knock-on delays (Carey and Kwiecinski 1994). Original delays occur when some technical failures happened to the railway networks, like rolling stocks system failure, signalling systems failure, bad weather conditions, accidents etc. These are random disturbances which are very difficult to forecast, and most of these delays can be regarded as independent incidents. Knock-on delays refer to the delays transferred to other trains due to the original delays of one or more trains in railway networks. For example, when one train is delayed, this delay may hinder the following trains to occupy the scheduled route due to the headway control constraints. This will cause knock-on delays to the following trains. Generally, with robust timetables, the knock-on delays can be absorbed gradually by the margins in the timetables.

The train service control problem associated with a mainline railway topology of the type shown in Figure 1-1 can be appreciated if it is assumed that each of the five routes leading into the bottleneck section and the bottleneck section itself is operated to a conventional timetable. In this scenario, a relatively short original delay to one train may cause long knock-on delays for following trains on the same route and merging trains on other routes, because of resource conflicts created by crossing moves and the necessary signal overlaps. Any disruption on one of the 'feeding' lines can result in large service gaps in the bottleneck section or in situations where trains have to queue to enter the bottleneck.

1.1.2 Problem Statement

A typical example of train rescheduling problems in a junction area is shown in Figure 1-2. There are two trains, Train 1 and Train 2 approaching the station ahead from different routes, via the same junction point. The nominal train trajectories for the two trains are shown as curve 1 and curve 2 respectively in Figure 1-2. For instance, if Train 1 is delayed from curve 1 to curve 3 because of disturbances, it will cause conflicts with Train 2 at the junction point. Without timely traffic management, Train 2 has to make an unplanned stop before the junction point, as shown with curve 5. This consumes more time and more energy. If the conflict can be detected and Train 2 could receive a train rescheduling decision from the traffic management system in advance, Train 2 can slow down when approaching the junction point with a trajectory as shown with curve 4, and the unplanned stop caused by the delayed Train 1 can be avoided. This will reduce consequential train delays and energy consumption in the event of disturbances.

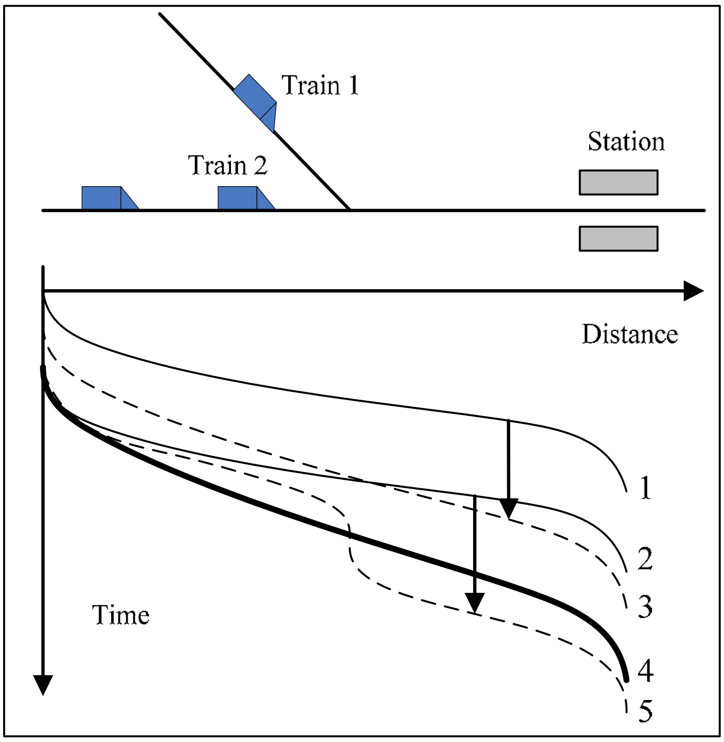

Figure 1-2 Example of train rescheduling

Considering all approaching trains to the junction point in a time window, the rescheduling problem also refers to the optimisation of train sequences and train arrival time at junction points. This can be represented by a special Binary Decision Tree, shown in Figure 1-3, which shows the process of rescheduling trains through a two track junction where a fly-over separates the flows of trains in opposite directions of travel. Every branch of the decision tree denotes a route setting for a train on one of two different routes approaching the junction. Also, the trains' arrival time can be denoted by the length of branches. The optimisation objective is to find the optimal decision tree branch route with the optimal duration (train arrival time).

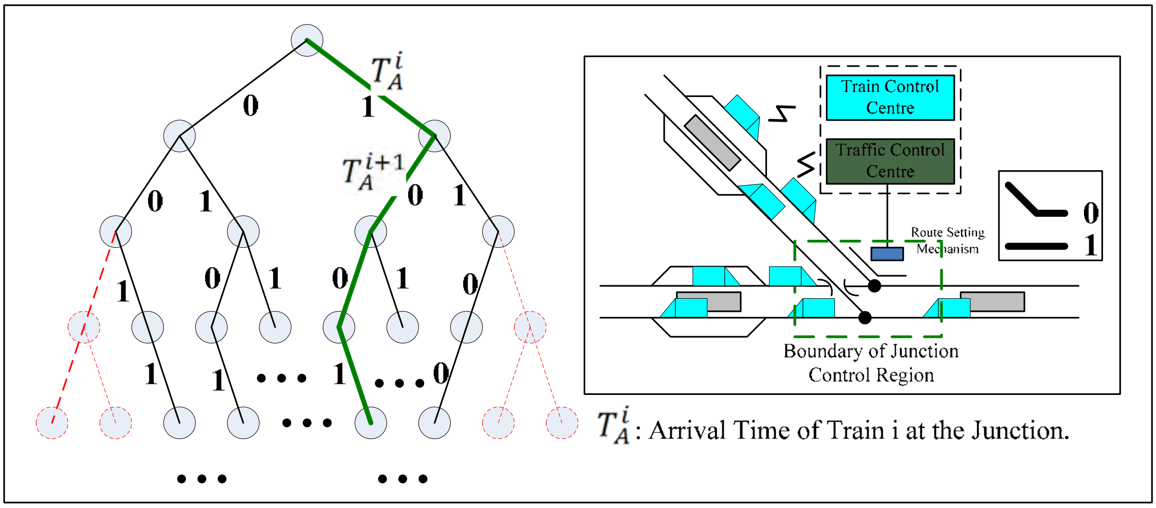

Figure 1-3 Representation of train rescheduling process in junction area with a binary decision tree

Conventional train service management approaches cannot reliably achieve a level of timetable adherence that permits accurate presentation of trains at portals. As a result, there are situations where train sequences must be changed, that is, trains must be rescheduled to minimise the overall delay to a set of services. This is necessary not only to be able to continue to offer a minimum service quality during service disruption but also to minimise the charges that are levied for train delays on many networks, as part of an access charging regime. The associated cost function may be expressed in monetary terms or energy consumption or in weighted delay minutes or the additional overall journey time, as in London Underground's Journey Time Capability Model (Transport-for-London), as well as the particular definition of passenger satisfaction.

Where services are rescheduled, retimed or re-sequenced, it is essential to provide early information to waiting passengers, so that station operations are not impaired by movements and passenger confusion. Therefore, any such decisions must be taken in real time, and looking ahead as far as possible, thus allowing passengers to make informed decisions.

1.2 Objectives and Contributions of the Thesis

The author focuses on solving real time train rescheduling problems in junction areas and bottleneck sections on mainline railways. With regard to the proposed train rescheduling problems, the overall objective is to establish a systematic methodology for real time train rescheduling in junction areas and bottleneck sections, including train re-sequencing and re-timing, which is applied to reduce weighted average delays of trains in the event of disturbances. This mainly includes the following objectives:

a) To formulate real time train rescheduling problems in junction areas.
b) To develop efficient algorithms for solving proposed real time train rescheduling problems in junction areas.
c) To extend the methodology to solve real time train rescheduling problems in bottleneck sections.
d) To present an integrated system architecture of train rescheduling and control for junction areas and bottleneck sections.

As for the objectives above, this thesis mainly contributes as follows:

(1) A formulated model "Junction Rescheduling Model (JRM)" based on a mixed integer programming (MIP) for train rescheduling in junction areas is proposed, dealing with re-sequencing and re-timing of trains in junction areas.
(2) An algorithm DE_JRM based on Differential Evolution is introduced for solving proposed JRM problems in this thesis. The algorithm DE_JRM integrates a

generic Differential Evolution Algorithm and stochastic greedy modification rules, to be an efficient algorithm for solving JRM problems.

(3) A rescheduling algorithm evaluation method based on Monte-Carlo simulation methodology is presented. The method gives a quantitative result for the evaluation of rescheduling algorithms.

(4) An extension of the proposed methodology for solving real time train rescheduling problems in bottleneck sections is described.

(5) An integrated system architecture train rescheduling and control for junction areas and bottleneck sections is presented. Two system configuration options are discussed, including benefits and shortcomings.

1.3 Outline of the Thesis

This thesis is structured into seven main chapters, as shown in Figure 1-4.

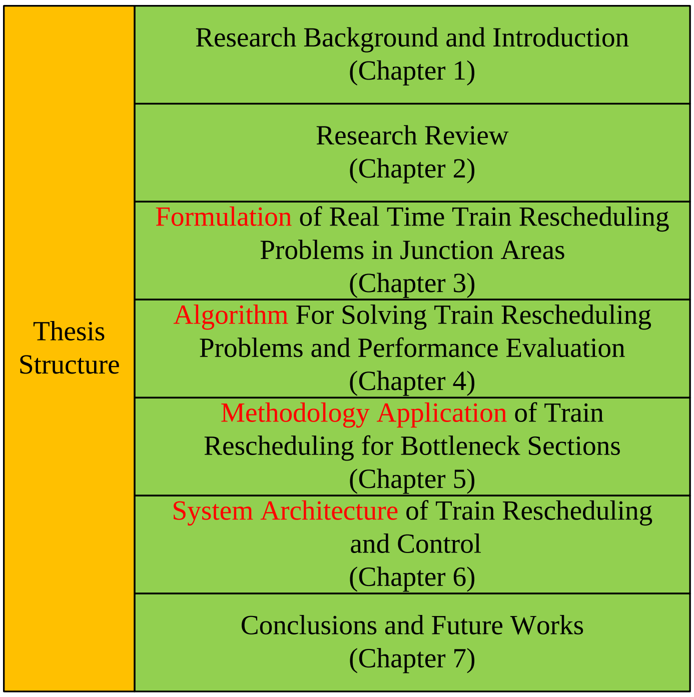

Figure 1-4 Thesis Structure

The overall research motivations and background are introduced in Chapter 1, in particular, the train rescheduling problems to be dealt with in this thesis are presented.

Chapter 2 gives a general introduction and review of related researches which have been done all over the world. The research review is catalogued into two aspects, railway timetabling and train rescheduling.

Chapter 3 formulates the real time train rescheduling problems in junction areas. A Junction Rescheduling Model (JRM) based on a Mixed Integer Programming (MIP) is proposed.

As for JRM, Chapter 4 introduces an innovative algorithm DE_JRM based on generic Differential Evolution, for solving the proposed JRM problems. The algorithm DE_JRM is evaluated with a statistical evaluation methodology based on the Monte-Carlo Simulation method, and the results are compared with First Come First Served (FCFS) and a strategy of Automatic Route Setting (ARS).

Chapter 5 extends the methodology of train rescheduling in junction areas to solve the train rescheduling problems in bottleneck sections. A case study on the Core Area of the Thameslink Route is described.

Chapter 6 proposes an integrated system architecture of train rescheduling and control for junction areas and bottleneck sections which can be applied for the implementation of the proposed train rescheduling methodology in this thesis.

Conclusions and future work are presented in Chapter 7.

Chapter 2. Research Review of Railway Traffic Management and Control

Railway traffic management and control mainly refers to the tasks of increasing the capacity of railway infrastructures, ensuring the safety of railway traffic control and improving the level of railway services. Due to the massive cost of enhancing railway infrastructure to meet rapid growth in railway transport demand, there has been more and more attention paid to efficient railway traffic management and control on mainline railways. Because of the complexity of the railway operations of mainline railways, systematic approaches are required for traffic management and control. These approaches mainly refer to two aspects: efficient railway timetabling and real time train rescheduling (Hansen 2006; D'Ariano 2010; Hansen 2010). The objective of this chapter is to give a state of the art of the research on railway traffic management and control across the world.

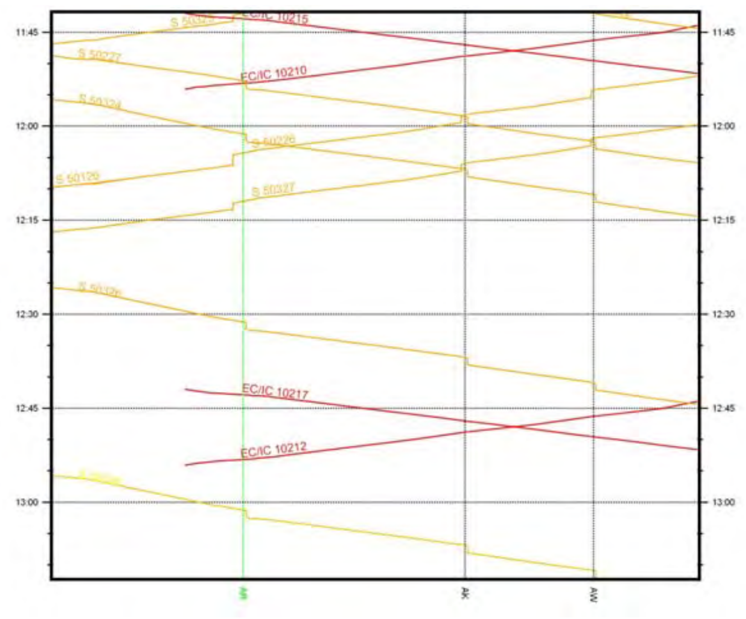

Figure 2-1 Timetable in Time-distance diagram

Timetabling is the fundamental process of constructing a set of schedules (or nominal timetable, which is planned running schedule for trains in railway networks) for trains in railway operations, which includes the scheduled times of the train at a series of specific locations on the train's journey without any conflicts. A typical railway timetable in time-distance format is shown in Figure 2-1. In the process of railway timetabling, the running times of different train types between scheduled stops or specific timing points (junctions etc) need to be calculated in advance based on the characteristics of railway infrastructure and rolling stock. The recovery times and margin times also need to be added on to train running times to cope with driving variations, minor delay incidents etc, which makes the timetables "robust" (Pachl 2002). In real time railway operations, however, the nominal timetable may not be kept due to perturbations to railway systems, and may be modified in the process of train rescheduling for minimising specific costs, for instance train delays. The aim of this chapter is to review various approaches to railway traffic management and control research in terms of railway timetabling and train rescheduling.

2.1 Timetabling

Railway timetabling is a complex process of constructing nominal timetables for trains to increase train throughputs, decrease waiting time or journey time for passengers and improve stability and robustness of the schedules. Generally it needs to be compromised between optimisation of timetabling and robustness of timetabling (Nachtigall and Voget 1997; Vansteenwegen and van Oudheusden 2006; Liebchen, Schachtebeck et al. 2010). For example, adding more margin time into a nominal timetable may improve the robustness of the schedule but it will decrease the throughput of trains.

2.1.1 Optimisation of Timetabling

Generally, railway timetabling is to allocate the railway track resources to trains carrying passengers or cargos across railway networks, satisfying certain railway operation constraints (Mees 1991). In mathematics, the problems can normally be regarded as resource allocation problems with one or more optimising objectives.

Brannlund et al presented a novel optimisation approach for the timetabling problem of different types of services to obtain a profit maximising timetable, while not violating track capacity constraints. They modelled the problem as a very large integer programming problem, and Lagrangian relaxation solution approach was applied for track capacity constraints. Their testing work on a single track railway consisting of 17 stations shows the good performance of the approach in terms of computation times and optimality of the obtained timetable (Brannlund, Lindberg et al. 1998).

Wong et al studied the passenger interchanges between different lines in urban transit railways. They presented a mixed-integer-programming optimisation model for the schedule synchronisation problem for non-periodic timetables that minimises the interchange waiting times of all passengers, and an optimisation-based heuristic for the model was used. The algorithm testing was undertaken for the Mass Transit Railway (MTR) system in Hong Kong (Wong, Yuen et al. 2008).

In many countries of Europe such as the UK, Switzerland and Germany, passenger railway timetables are usually regulated to be periodic, which makes the timetable easier to remember for railway staff and passengers and also simple to manage. Certain research on railway timetabling focuses on optimising periodic timetables of railway networks.

Nachtigall introduced a concept of “Periodic Networks” for the problems to find a timetable for a selected class of change possibilities where the arising waiting time is minimal for a fixed time interval served railway system, and presented a branch and bound approach for solving the problems (Nachtigall and Voget 1997).

Caprara et al proposed a graph theory formulation for the timetabling problem for a single one-way track linking two major stations with a number of intermediate stations in between. The problem was formulated with linear integer programming, and Lagrangian relaxation is used to derive bounds on the optimal solution value and also applied in heuristic procedures. They reported extensive computational results on real world instances from Italian railways (Caprara, Fischetti et al. 2002; Caprara, Monaci et al. 2006).

Liebchen (2005) considered periodic railway timetable construction problems as the problems of satisfying the maximum number of constraints of an instance of the Periodic Event Scheduling Problem (PESP), which was initially developed by Serafini and Ukovich (Serafini and Ukovich 1989). Liebchen presented a deterministic combinatorial polynomial time algorithm with a cut-based heuristic method (Liebchen 2005; Liebchen and Peeters 2009).

Caimi, Fuchsberger et al made an extensive of PESP called flexible periodic event scheduling problem (FPESP), which allows flexible time slots to be generated for the departure and arrival times instead of exact times. The FPESP formulation increases the chance to obtain feasible solutions, in particular for stations with dense peak traffic. They tested the method on instances of Swiss railways and show the solution time did not increase significantly (Caimi, Fuchsberger et al. 2011).

2.1.2 Robustness of Timetabling

Besides the optimisation of railway timetabling, timetable robustness is also a very important aspect for timetable construction. Generally robustness means the persistence of a system's characteristic behaviour under perturbations or conditions of uncertainty (Hampel 1971). Robustness denotes the capability of the timetable for dealing with small disturbances in real time railway operations. One of the straightforward ways to improve timetable robustness is adding more recovery time and margin time into the nominal timetable (Pachl 2002).

There have been several approaches developed to analyse and improve the robustness of timetables using Max-Plus algebra (Goverde 2007), mathematical programming (Fischetti, Salvagnin et al. 2009), queuing theory (Huisman and Boucherie 2001), stochastic model (Yuan and Hansen 2007; Kroon, Maroti et al. 2008), and simulation method (Middelkoop and Bouwman 2000).

Several commercial systems can be applied as decision support systems for different levels of railway timetable construction. Watson has compared the characteristics of current commercial simulation tools, particularly the simulation tools applied in the UK (Watson 2005). Microscopic simulator RailSys (Radtke and Hauptmann 2004) and Opentrack (Nash and Huerlimann 2004) can model railway traffic flows accurately by calculating accurate train movements interactively in complicated railway networks based on detailed information of characteristics of infrastructure and rolling stocks. Macroscopic simulators like SIMONE (Middelkoop and Bouwman 2001; Vromans, Dekker et al. 2004) can be used for simulation of the effects of small disturbances, which performs an evaluation of the robustness of timetables. These tools will give much help to the process of railway timetabling for railway operations in terms of timetable construction, simulation, analysis and evaluation.

2.2 Rescheduling

In the last section, the research development of railway timetabling has been briefly reviewed. During real time railway operations, there are several causes of service perturbations that may lead to different levels of train delays. For example, the delays could be caused by extra dwell time, longer running time, system failures, and temporary speed restriction due to construction work etc. As the delays occur, it is quite difficult to follow the nominal timetables, so the planned schedule needs to be modified in real time. The objective of real time train rescheduling is to find optimal solutions for recovery from the disturbances as soon as possible and minimise the cost which has arisen due to the disturbances. The cost function could be in the form of total delays, penalty charge or weighted delays etc, which depends on the objectives of the rescheduling. Sometimes only near optimal solutions are available in reasonable time, which is necessary for solving real time train rescheduling problems. A ‘limit computation time’ requirement is a big challenge for solving real time train rescheduling problems.

There has been much research focusing on modelling and solving real time train rescheduling problems with efficient modelling of the problems, simplifying the complexity of the problems and increasing the speed of decision making. In this chapter the methods applied in real time train rescheduling are reviewed and discussed.

2.2.1 Rule based Approach

Because of the complexity of railway traffic management, rule based approaches have been widely used to solve train rescheduling problems, which are too difficult to be solved in real time. Efficient dispatching measures can be summarised from experienced dispatchers who successfully deal with lots of difficult railway traffic situations successfully. These dispatching measures can be processed into certain rules and pre-set into decision support systems for real time train rescheduling, which can also be called knowledge based train rescheduling and this type of decision support systems can be called expert systems.

For solving online rescheduling problems of mass rapid transit (MRT) trains after sudden increases in passenger flow, a knowledge based system for improving the performance of MRT systems and for enhancing functions of the typical automatic train control (ATC) systems was proposed by Chang and Thia, with the use of predictive fuzzy control. The proposed approach was mainly used for adjusting the train dwell time. The results showed that dwell time adjustment is an effective means of maintaining the quality of train service after sudden load disturbances (Chang and Thia 1996).

An expert system called “UWS” has been successfully applied for train operation adjustment of the Tokaido and Sanyo Shinkansen lines from 1995 in Japan. The system mainly includes a problem solving architecture ESTRAC which is designed to emulate experts' problem solving processes on the computer for generating a practical rescheduling plan for disturbed railway traffic within a short time (Hyoudou, Seto et al. 1997).

A dispatching support system with expert knowledge in fuzzy rules of the "IF-THEN" type was described by Fay for a railway operation control system. Rule-based expert knowledge in a decision support system was modelled with a Fuzzy Petri Net notion which combines the graphical power of Petri Nets and the capabilities of Fuzzy Sets. The system architecture of a train traffic control assistant system was also presented (Fay 2000).

In the UK, the Automatic Route Setting (ARS) system has been widely applied to deal with real time train rescheduling at railway junctions. The overall description of ARS can be found in (Kuhn 1998). ARS can also be regarded as rule-based traffic control systems. ARS mainly provides the functions of selecting and setting routes automatically for approaching trains from different origins or to different destinations in terms of train classes, present delays and destinations. The conflicts between trains approaching a junction are solved by ARS based on certain predefined rules. One drawback of ARS is that the system only considers delay minimisation at a single junction node for the first approaching trains on each route, without consideration of the effects of rescheduling on the following trains, as well as other junctions in railway networks.

2.2.2 Simulation Approach

Simulation can also be a powerful tool to support resolving resource conflicts in train traffic rescheduling. Simulation methods have the capability of showing dynamic characteristics of train traffic rescheduling strategies. Simulation analysis methods are suitable for solving problems that have no analytical or mathematical solutions.

Cheng analysed the existing event-driven simulation and network-based simulation for railway train traffic rescheduling, and proposed a new simulation method suitable for different stages of multiple rescheduling strategies. A simple example was used to illustrate the proposed method, and available simulation strategies are also discussed. The results of computational experiments showed that the proposed method has the same results as previous methods under the same inputs, and also the execution time is at the same level (Cheng 1996). Another hybrid method of network-based simulation and event-driven simulation was proposed by Cheng to reduce the shortcomings of the previous methods (Cheng 1998).

Simulation methods can also be used for validation of the rescheduling strategies. Ho developed a traffic controller for a railway junction to minimise total weighted delays. A dynamic programming method was applied to model the traffic flow in railway junctions. With a Multi-Train simulator (MTS), the performance of the controller was evaluated (Ho, Norton et al. 1997).

In 2004, Jacobs studied a means of automatic traffic regulation incorporating the basic aspects of train path management. He presented a computer aided procedure to generate conflict free rescheduled plans for disturbed trains with an asynchronous simulation approach based on blocking times (Jacobs 2004).

Luethi et al proposed an approach for real-time train rescheduling that could enable buffer times to be reduced without impacting schedule reliability. A microscopic simulation was completed to show the effectiveness of the approach, which highlighted the importance of accurate train operations for recovery from disturbances (Luethi, Medeossi et al. 2009).

2.2.3 Heuristic Approach

Heuristic methods including the Greedy Algorithm (Krasemann 2010), Evolutionary Algorithms (Chen, Schmid et al. 2010), Genetic Algorithms (Takagi, Weston et al. 2006), Tabu Search (Corman, D'Ariano et al. 2010) and Simulated Annealing (Tomii, Tashiro et al. 2005) have been studied and widely used for railway train rescheduling due to their good performance on computation time. However most of the heuristic methods do not guarantee that global optimal solutions can be found, rather, near optimal solutions are expected to be found in reasonable computation time.

Chiu et al applied two heuristics to speed up and direct the search towards the optimal solution of a constraint satisfaction problem for train rescheduling. Two optimality criteria for rescheduling that correspond to minimising the number of station visits affected and passenger delay respectively. The feasibility of the proposed algorithms and heuristics were confirmed with experimentation using real-life data (Chiu, Chou et al. 2002).

To improve passenger satisfaction for railway services, Tomii et al formulated the problem of train rescheduling as a constraint optimisation problem in which the degree of passenger dissatisfaction should be minimised. They introduced an algorithm combining program evaluation and review technique (PERT) and meta-heuristics for solving train rescheduling problems. The experimental results showed good performance of the algorithm (Tomii, Tashiro et al. 2005).

In heavily used railway networks with heterogeneous train flows, such as the railway network in the Netherlands, the margin time in the timetable is very short. Thus even short train delays could cause knock on delays. D'Ariano and Albrecht modelled the train rescheduling problems as an alternative graph for conflict solution systems and they proposed a constructive heuristic algorithm for the dynamic modification of train running times and to satisfy the timetable constraints of train orders and routes, as well as the feasibility of the running profile. The benefits of the proposed methodology were demonstrated with a real example in the Netherlands (D'Ariano and Albrecht 2006).

As for the similar objective across the European railway network, Tornquist presented a heuristic approach for railway traffic rescheduling in the event of disturbances. With a comprehensive performance evaluation, he found that a minimisation of accumulated delays has a tendency to delay more trains than a minimisation of total final delay or total delay costs. He also did an experimental study of how the choice of planning horizon in the rescheduling process affects the network in the longer term (Tornquist 2007).

Corman et al proposed new Tabu Search algorithms for solving real time train traffic rescheduling problems with a short computation time. They applied the new heuristic algorithms into a real time traffic management system ROMA (Railway traffic Optimization by Means of Alternative graphs) and compared the optimised solutions with the solutions from previous Branch and Bound algorithms in ROMA. Their computational experiments show good performance of the new heuristic algorithms in terms of computation time and goodness of the optimised solutions (Corman, D'Ariano et al. 2010).

Chen and Schmid et al proposed an improved Differential Evolution algorithm for solving train rescheduling problems in junction areas in the event of disturbances. Compared with FCFS (First Come, First Served), the weighted average delays were significantly decreased and the algorithm also shows a good performance on computation time, which can be regarded as suitable for real time train rescheduling applications (Chen, Schmid et al. 2010).

Heuristic methods significantly improve the speed of feasible solution searching for train traffic rescheduling problems. The application of heuristics methods makes the decision support systems possible and practical for real time train rescheduling in a railway network. However, although global optimal solutions are not guaranteed to be found with heuristic methods, near optimal solutions can normally be found in reasonable computation time.

2.2.4 Other Approaches

Hirai et al proposed an integrated algorithm framework based on a pattern description language for automatic train rescheduling, especially for severe train traffic disruptions caused by an accident requiring more than about an hour of suspended train operations. They believe that their algorithms and framework are helpful for preparing adequate rescheduling plans for practical use (Hirai, Tomii et al. 2007).

Based on alternative graph modeling methods, Branch and Bound methods have also been widely used for solving train and traffic rescheduling problems (D'Ariano, Pacclarelli et al. 2007). By contrast with heuristic methods, Branch and Bound methods are deterministic methods, which mean the algorithms provide the same outputs with the same inputs all the time without randomness.

Tornquist and Persson formulated the train rescheduling problems with a MIP (Mixed Integer Programming) in an n-track network, and solve the mathematical programming problems with mixed heuristic algorithms and commercial solvers like CPLEX. They also presented the theoretical and practical strengths and limitations of the approaches (Tornquist and Persson 2007).

Distributed approaches have also been applied for train traffic rescheduling for many years. The basic idea of distributed approaches is to decompose the large scale problems into several sub-problems with reasonable scale, and solve the sub-problems locally with sorts of collaborative mechanism to achieve a global optimum.

Nobuyuki et al proposed a rescheduling method based on a distributed cooperative problem solving model for train rescheduling. Each planner in the system can search local solutions independently and simultaneously with a constraints relaxation approach (Nobuyuki, Akatsu et al. 1996).

Chou et al introduced a collaborative rescheduling method to optimise the train passing sequences in junctions of a railway network. They decompose the rescheduling problem for an entire railway network into rescheduling problems for each junction area, and try a greedy local search for optimised solutions for each junction until local optimised solutions for each junction do not have conflict between each other (Chou, Weston et al. 2009).

There have been some real world train traffic rescheduling systems applied into operation for recovery from disturbances (Mazzarello and Ottaviani 2007; Luethi 2008; Mannino and Mascis 2009; Mehta, Rößiger et al. 2010). As the development of theoretical and practical applications of real time train rescheduling continues, integrated real time train traffic management and control systems for a large scale railway network could come into practise in the not far future.

2.3 Conclusions

In this chapter, the main research on railway traffic management and control, including static railway timetabling and real time train rescheduling has been reviewed. In railway timetabling, the relevant research was catalogued into two main aspects, which are optimality of timetables and robustness of timetables. Different objective functions have been defined for optimality of timetables, and different definitions and measurements of timetable robustness were also reviewed. The research on train traffic rescheduling was then reviewed in terms of the applied approaches.

In earlier studies, most of the traffic management strategies for conflicts resolution in train rescheduling focused on solving combinatorial optimisation problems like train sequence changes, train connections combination, trains re-routing, while disregarding the train running time optimisation issue. With consideration of train re-sequencing and train re-timing for real time train rescheduling, new efficient algorithms are required for solving these large scale hybrid optimisation problems with discrete and continuous variables, and with computation time constraints for real time train rescheduling applications. In this thesis, a train rescheduling model which focuses on the re-sequencing and re-timing of perturbed train services approaching junction points and bottleneck sections is proposed, and an efficient innovative algorithm based on Differential Evolution (DE) algorithms is developed, with performance evaluations compared with two conventional strategies.

Chapter 3. Formulation of Real Time Train Rescheduling Problems in Junction Areas

In the previous chapters, the main problems of real time train rescheduling in junction areas in the event of disturbances have been described. The main purpose of the rescheduling methodology presented in this thesis is to provide real time optimised rescheduled timetables for the trains approaching junction areas and bottleneck sections, with optimised train sequences and train arrival times at the junction points.

In this chapter, a mathematical train rescheduling model for junction areas, which is named as Junction Rescheduling Model (JRM), is presented for the formulation of the proposed junction rescheduling problems using a Mixed Integer Programming (MIP). In the first section, the general concepts of mathematical programming are introduced including linear programming, non-linear programming and integer programming etc. Then the formulation of Junction Rescheduling Model (JRM) using a MIP is described in detail with a case explanation.

3.1 Introduction to Mathematical Programming

3.1.1 General Definitions of Mathematical Programming

The concept of mathematical programming can be traced back to 1947, when the Simplex Method was discovered by the American mathematician George Dantzig for numerically solving linear programming problems (Dantzig 1948). Mathematical programming is concerned with the determination of a maximum or a minimum of an objective function with several variables, which have to satisfy a number of constraints. Generally a mathematical programming can be described as an optimisation problem subject to constraints in the following form:

$$(P)\begin{cases} \quad Minimise\ f(x) \\ \quad subject\ to: \\ g_i(x) \leq 0 \ \ i = 1,\cdots,m \\ h_j(x) = 0 \ \ j = 1,\cdots,n \\ \quad x \in S \subset \mathbb{R}^n \end{cases}$$

Equation 3-1

where $x \subset \mathbb{R}^n$is a vector which has elements$x_1, \cdots x_m$, and is the unknown vector variable of the optimisation problem.

The function $f(\cdot)$ is called the objective function which decides the optimisation objective. In some of the cases, it also can be called the "cost function". The set of conditions $g_i(x) \leq 0 \ \ i = 1,\cdots,m$ and $h_j(x) = 0 \ \ j = 1,\cdots,n$ are called inequality constraints and equality constraints of the optimisation problem respectively. $x \in S \subset \mathbb{R}^n$ defines the value range of the unknown vector x in the optimisation problem.

For some of the optimisation problems the objectives are to a maximum of a function$f(\cdot)$. They can also be converted to the problem of minimisation of$f'(\cdot) = -f(\cdot)$.

In the problem(P) shown in Equation 3-1, every vector x which satisfies the constraints, $g_i(x) \leq 0 \ \ i = 1,\cdots,m$, $h_j(x) = 0 \ \ j = 1,\cdots,n$ and $x \in S \subset \mathbb{R}^n$ can be called a solution of the problem(P). The solution which minimises $f(\cdot)$ is an optimal solution of the problem (P), which can also be called a global optimum solution.

Relative to global optimum, a vector x^0 is called a local optimum of the problem(P), if and only if there exists a neighbourhood $N(x^0)$ of x^0, and x^0 is the global optimum of the problem as shown in Equation 3-2:

$$
(P') \begin{cases} Minimise\ f(x) \\ subject\ to: \\ g_i(x) \leq 0 \ \ i = 1, \cdots, m \\ h_j(x) = 0 \ \ j = 1, \cdots, n \\ x \in S \cap N(x^0) \end{cases}
$$

Equation 3-2

Figure 3-1 shows a simple example for the concepts of global and local optimum.

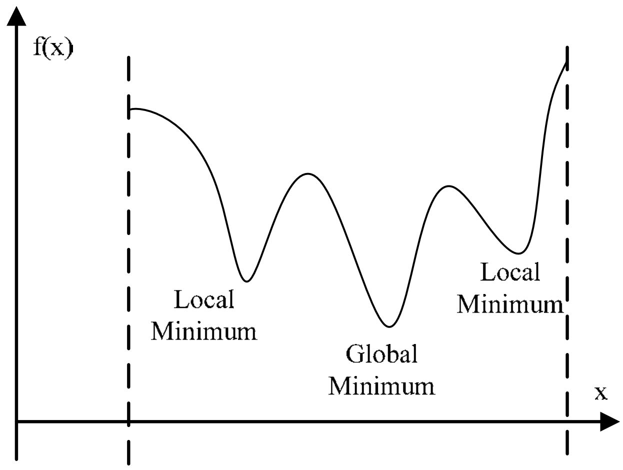

Figure 3-1 Global optimum and local optimum

In mathematical programming, as for the set S, if and only if

$$
\begin{cases} \forall x^1 \in S \\ \forall x^2 \in S \\ \forall \lambda \in [0,1] \end{cases} \Rightarrow \lambda x^1 + (1-\lambda)x^2 \in S
$$

Equation 3-3

Set S is regarded as convex.

The convexity of the set S is an important property in a mathematical programming problem. Generally it is much more difficult to solve non-convex mathematical programming problems because it is quite difficult to characterise the global optima of an optimisation problem.

3.1.2 Classification of Problems in Mathematical Programming

According to the properties of the function $f(\cdot)$, the constraints functions $g_i(\cdot)$ and $h_j(\cdot)$, and the definition of the subset S of $\mathbb{R}^n$, the problems in mathematical programming can be classified into several categories. In terms of the linearity of the objective function$f(\cdot)$, constraints function $g_i(\cdot)$ and $h_j(\cdot)$, the programming problems can be categorised into linear programming and non-linear programming. According to the convexity and continuity of subset S, they can also be divided into convex programming and non-convex programming, continuous programming and discrete programming. In some of the optimisation problems, there are no constraints functions; these problems can be called unconstrained programming, and the others are constrained programming. Some of the subset S only consists of integral values, which can be regarded as integer programming. In terms of other different properties of the optimisation problems, they can also be classified as stochastic and deterministic programming, dynamic programming, quadratic programming, conic programming etc in mathematical programming. Most of the optimisation problems in mathematical programming have multi-properties, for example the objective function of an optimisation problem is non-linear and the problem has constraints for unknown variables, this kind of optimisation problem can be called non-linear constrained programming in mathematical programming. Similarly there are terminologies like linear constrained programming and mixed integer programming etc. Table 3-1 shows the main classes of the various types of optimisation problems in mathematical programming. More details are given in the following sections.

Objective Function $f(\cdot)$	Constraints Function $g_i(\cdot)$ / $h_j(\cdot)$	Set S	Terminology
Continuous linear/non-linear		Continuous	Continuous programming
Linear/non-linear		Discrete	Discrete programming
Linear/non-linear	$m = 0$ And $n = 0$	$S = \mathbb{R}^n$	Unconstrained programming
Linear/non-linear	$m \neq 0$ Or $n \neq 0$	$S \subset \mathbb{R}^n$	Constrained programming
Convex	Convex	$S \subset \mathbb{R}^n$ Convex	Convex programming
Linear		$S \subset \mathbb{R}^n$	Linear programming
Non-linear		$S \subset \mathbb{R}^n$	Non-linear programming
Linear		$S \subset \mathbb{Z}^n$	Linear integer programming

Table 3-1 Classes of problems in mathematical programming

3.1.3 Linear Programming

Linear programming is the earliest research branch of mathematical programming. In 1939, a Russian mathematician developed the linear programming problems first, and until 1947 when George B. Dantzig published the Simplex Method (Dantzig 1948), which can be used for solving linear programming problems, more and more industries started to apply linear programming into their daily planning. Linear programming deals with the optimisation problems which consist of linear objective functions, subject to linear constraints. Thus a linear programming problem can be presented in the following form:

$$\begin{cases} Minimise\ f(x) = c^T x \\ subject\ to: \\ g_i(x) = a_i^T x - b_i \leq 0 \ \ i = 1, \cdots, m \\ h_j(x) = a_j^T x - b_j = 0 \ \ j = 1, \cdots, n \\ x \in S \subset \mathbb{R}^n \\ c, a_i, a_j \in \mathbb{R}^n \\ b_i, b_j \in \mathbb{R} \end{cases}$$ **Equation 3-4**

All the linear programming problems in the form of Equation 3-4 can be put into standard form by introducing additional slack variables. The standard form of linear programming is shown in Equation 3-5.

$$\begin{cases} Minimise\ z = c^T x \\ subject\ to: \\ Ax = b \\ x \geq 0 \end{cases}$$ **Equation 3-5**

where A is a real $m \times n$ matrix, m and n are the number of constraints and variables respectively; c is the vector of cost coefficients. $b = (b_1, \cdots, b_m)^T$ is the right-hand sides vector; z is the objective function to be minimised. Most of the methods for solving linear programming problems consider the problems in the standard form of Equation 3-5.

The earliest method which people applied to solving the linear programming problem is the Simplex Method, the detailed explanation of the Simplex Method can be found in (Minoux 1986). Another important method was introduced by Narendra Karmarkar for solving linear programming problem in 1984, which is a new interior point method (Karmarkar 1984; Adler, Resende et al. 1989). The method can be applied to solve linear and nonlinear convex optimisation problems.

3.1.4 Nonlinear Programming

Compared with linear programming, nonlinear programming refers to the process of solving the optimisation problems in the form of P with nonlinear objective functions or constrains.

In this section, two kinds of nonlinear programming problems are introduced, which are unconstrained nonlinear programming and constrained nonlinear programming.

3.1.4.1 Unconstrained Nonlinear Programming

Equation 3-6 is the mathematical definition of unconstrained nonlinear programming.

$$\begin{cases} Minimise\ f(x) \\ \quad x \in \mathbb{R}^n \end{cases}$$ **Equation 3-6**

If the objective function $f(\cdot)$ is continuous and has continuous partial first derivatives and the second derivatives for all $x \in \mathbb{R}^n$, then a necessary condition for x^* to be a local or global optimum of $f(\cdot)$ is shown in Equation 3-7.

$$\begin{cases} \nabla f(x^*) = 0 \\ \nabla^2 f(x^*) = \frac{\partial^2 f}{\partial x_i \partial x_j}(x^*)\ is\ a\ positive\ semi-definite\ matrix. \end{cases}$$ **Equation 3-7**

If the objective function $f(\cdot)$ in $\mathbb{R}^n$ is a continuously differentiable convex function, then a necessary and sufficient condition for x^* to be a global optimum of $f(\cdot)$ is $\nabla f(x^*) = 0$.

Most of the algorithms for solving unconstrained nonlinear programming problems refer to the process of searching the optimal solutions x^* for the objective function from a starting point x_0. The commonly used algorithms are Steepest Descent method, Newton's method, Quasi-Newton's method and some of the other methods without the calculation of derivatives (Powell 1964). The selection of starting point x_0 significantly affects the performance of the algorithms like searching speed, goodness of the solution.

In many mathematical programming problems there are some unconstrained nonlinear programming problems, the objective function of which is a concave or convex, but not everywhere differentiable. Generally, we can deal with this kind of mathematical programming problem using decomposition methods (Tai and Espedal 1998).

3.1.4.2 Constrained Nonlinear Programming

In practical optimisation applications, many optimisation problems have inequality constraints and/or equality constraints which need to be satisfied. Generally the problems can be defined as shown in Equation 3-8.

$$\begin{cases} \quad Minimise\ f(x) \\ \quad subject\ to: \\ g_i(x) \leq 0 \ \ i = 1, \cdots, m \\ h_j(x) = 0 \ \ j = 1, \cdots, n \\ \quad x \in \mathbb{R}^n \end{cases}$$

Equation 3-8

The necessary conditions for a solution in nonlinear programming to be optimal refer to Karush–Kuhn–Tucker conditions (Kuhn–Tucker or KKT conditions). KKT conditions were first stated by William Karush in his master's thesis in 1939 as the necessary conditions for constrained nonlinear programming problems and originally named after Harold W. Kuhn, and Albert W. Tucker in 1951 (Karush 1939; Kuhn 1951).

If x^* is a local minimum, supposing that the objective function $f(\cdot)$, constraints function $g_i(\cdot)$ and $h_j(\cdot)$ are continuous differentiable at point x^*, there exist constants $u_i(i = 1, \cdots, m)$ and $\lambda_j(j = 1, \cdots, n)$, which are called KKT multipliers that satisfy the conditions as shown in Equation 3-9:

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^{m} u_i \nabla g_i(x^*) + \sum_{j=1}^{n} \lambda_j \nabla h_j(x^*) = 0 \\ g_i(x^*) \le 0 \quad i = 1, \cdots, m \\ h_j(x^*) = 0 \quad j = 1, \cdots, n \\ u_i \ge 0 \quad i = 1, \cdots, m \\ u_i g_i(x^*) = 0 \quad i = 1, \cdots, m \end{cases}$$ **Equation 3-9**

To solve the condition functions above, we can get the solutions of all local minimums. The minimum of all local minimums is the optimum of the optimisation problems.

In general, the necessary conditions are only sufficient for global optimality for certain types of constrained nonlinear programming problems (Martin 1985). Those conditions are not sufficient for general constrained nonlinear programming problems unless some additional conditions are satisfied, such as Saddle-point condition, Second Order Sufficient Conditions (SOSC) (Neumaier 1996).

So far, many methods have been developed for solving constrained nonlinear programming problems. The methods can be divided into two large groups: direct methods and methods using the concept of duality.

Direct methods operate the given optimisation problems directly by generating a sequence of solutions which satisfy the constraints to minimise the value of the objective function step by step. The common direct methods are Method of Changing the Variables, Method of Feasible Directions, the Reduced Gradient Method and Newton's Method etc (Schenk 1998).

The methods using the concept of duality are mainly Penalty Function Methods and Classical Lagrange Methods. The common principle is to convert the given constrained programming problems into a sequence of unconstrained programming problems, and then to solve them with methods for unconstrained programming problems.

In general, it is difficult to get optimum solutions from direct methods, because of the limit of iterative processes. If the iterative process is interrupted because of computation time limits etc, they can still offer an approximate solution. Contrary to direct methods, the methods using the concept of duality are more robust and easy to obtain global convergence, but only upon the termination of iterative processes.

3.1.5 Integer Programming

In mathematical programming, there is an important area which refers to optimisation problems in which the variables are constrained to only integers. Because of the difficulty and the wide applications of this class of problems, there are large amounts of research work devoted to this area. Generally integer programming can be categorised into pure integer programming and mixed integer programming in terms of the property of the variables in optimisation problems. Pure integer programming deals with the integer programming problems in which all of the variables need to be taken integer values. In some cases, not all the variables need to be taken as integers, for example, they also include some other continuous variables. This kind of optimisation problem is referred to as Mixed Integer Programming. Within integer programming, there is a special class of problems in which the variables are restricted to take values 0 or 1. The optimisation of this class of problems is called 0-1 programming or binary integer programming.

Equation 3-10 is the mathematical definition of integer programming problems.

$$\begin{cases} Minimise\ f(x) \\ subject\ to: \\ g_i(x) \leq 0\ \ i = 1, \cdots, m \\ h_j(x) = 0\ \ j = 1, \cdots, n \\ x_k integral\ (\forall k = 1, \cdots, l) \\ x \subset \mathbb{R}^n \end{cases}$$

Equation 3-10

Compared with constrained programming problems, integer programming problems restrict the values of variables to be integers. To solve integer programming problems, there are three main families of methods which can be applied, and they are Branch and Bound methods, cutting-plane methods and meta-heuristics methods.

Branch and Bound methods

The principle of Branch and Bound methods was firstly introduced in 1960 (Land and Doig 1960). The methods have since been widely used for many applications and improved by many authors. Generally Branch and Bound methods mainly include these steps:

First, split the whole value range set S (search space or feasible region) into two or more smaller sets $\{S_1, S_2, \cdots, S_k\}$, where $\bigcup_1^k S_i = S$. This procedure is called *branching*. A tree structure can be defined whose nodes are the subset S_i.

Second, calculate the lower bound and upper bound for the minimum value of $f(x)$over a given subset S_i, which is called *bounding*. If the lower bound over some subset S_A is greater than the upper bound over some subset S_B, then S_A can be safely discarded during the search procedures, this step is called *pruning*.

The recursive search and calculation continues in terms of these three procedures to gain the minimum value of $f(x)$ over set S, and stops when set S is reduced to a single element.

In practice, the Branch and Bound method is a systematic method for solving programming problems, especially for discrete programming including integer programming etc. The searching complexity depends on the selection of criterions for branching, bounding and pruning, as well as the stopping criterions. In the worst case they may lead to exponential time complexities.

Cutting-plane methods

There is another class of methods which are popularly used for solving integer programming problems, which are cutting-plane methods introduced by Ralph E. Gomory in the 1950s. They are a class of methods which iteratively refine a feasible set or objective function by means of linear inequalities. The basic idea of cutting-plane methods is: firstly, the method relaxes the integer restriction of unknown variables, and solves the associated problems to obtain a basic feasible solution, so this solution is thought to be a vertex of the convex polytope in geometry that consists all of feasible points. If the vertex is not an integer, then find a hyperplane with the vertex and all feasible integer points on each side and add it as a additional constraint to create a modified linear programming problem. Solve the new problem and iteratively execute this process until an integer solution can be found. Cutting-plane methods can also be extended to solving nonlinear programming problems (Konno, Kawadai et al. 2003).

Meta-heuristics methods

Most practical mathematical programming problems are quite difficult to solve by classic methods, like the travelling salesman problem, as the size of the problem grows, the solution search space will grow significantly, which makes it very hard to solve in a feasible computation time.

From the 1950s, there is a family of methods, named meta-heuristic methods, which have been widely used for solving integer programming problems. The meta-heuristic method was formally defined by Osman and Laporte as *"an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions"* (Osman and Laporte 1996).

Meta-heuristic methods are usually not deterministic optimisation methods, they implement some form of stochastic optimisation. Normally meta-heuristic methods deal with the combinatorial optimisation problems in which the solutions space is discrete. The widely used meta-heuristic methods include genetic algorithms by Holland et al. (Hooker 1995), simulated annealing by Kirkpatrick et al. (Kirkpatrick, Gelatt et al. 1983), tabu search by Glover (Glover 1986), scatter search (Glover 1977) and ant colony optimisation by Dorigo (Dorigo 1992).

Not only used for discrete programming problems, meta-heuristic methods can also be applied for solving optimisation problems in which the solution space has real-valued points. The popular methods include differential evolution by Storn and Price (Storn and Price 1997), particle swarm optimisation by Eberhart and Kennedy (Kennedy and Eberhart 1995) and evolution strategies by Rechenberg (Rechenberg 1971).

3.1.6 Conclusions

In this section, the main concepts of mathematical programming and related popular methods are reviewed in terms of the main divisions, linear programming, nonlinear programming and also a specific mathematical programming area integer programming. Generally, most practical optimisation problems can be formulated into mathematical programming problems, and normally it is very hard to solve the practical problems in an acceptable time with classic methods because of the exponential increase in the search space as the size of the problem grows. Especially for some real time applications, these problems must be solved with much more efficient and fast methods and algorithms. In the next section, a mathematical programming model for real time train rescheduling problems in junction areas is proposed based on a MIP.

3.2 Formulation of Junction Rescheduling Model (JRM) with a MIP

For better understanding and solving real time train rescheduling problems in junction areas, it is necessary to model the problems formally in mathematics. A formal Junction Rescheduling Model (JRM) is proposed in this section.

As is the case for any rescheduling problem, the objectives for the rescheduling optimisation must first be defined. In this thesis, a definition of Weighted Average Delay (WAD) is used as the objective function. The objective is to find the optimal arrival times of approaching trains at junctions that result in the lowest WAD. The mathematical representation of the WAD is shown in Equation 3-11.

$$WAD = \sum_{i,j} \omega_i \left| t_{i,j} - t_{i,j}^0 \right| / \sum_i \omega_i$$ **Equation 3-11**

ω_i denotes the weighting of train i. The value of ω_i depends on the class of the trains, and reflects the priorities of the network operator. Generally, the fast intercity trains are assigned bigger values of weighting than local trains; the passenger trains will also be assigned bigger values of weighting than freight trains. The detailed ratios of weighting for different classes of trains are determined by different railway network operators in different countries in terms of different railway operation requirements. $t_{i,j}^{0}$ is the arrival time of train i at junction/stop j in the Nominal Timetable and $t_{i,j}$ is the rescheduled arrival time of train i at junction/stop$j$.

In this thesis, early running of trains is regarded as bad as late because early running would bring more track occupation time by trains at stations. So the absolute value $\left|t_{i,j} - t_{i,j}^{0}\right|$ is applied in the objective function as shown in Equation 3-11, which means the objective is to find the optimal rescheduled arrival time $t_{i,j}$ for all the trains in the control region during a time window to reschedule the trains as close to nominal timetable as possible.

Let T be the set of trains approaching junction K in one control time window, E the set of events, where an event e denotes one train on approaching routes passing the junction area. Let R be the set of routes for approaching trains leading to the junction area.

The index r is associated with a route, and i_r denotes a train on route r. The index e denotes an event, and event $e-1$ is the preceding event of the event e.

$|R|$ is the number of routes for approaching trains leading to the junction area and the number of trains on route r in one control window is N_r.

$r \in R = \{1,2,\cdots,|R|\}$ **Equation 3-12**

$i_r \in \{1,2,\cdots,N_r\}, r \in R$ **Equation 3-13**

$e \in E = \left\{1,2,\cdots,\sum_{r=1}^{|R|} N_r\right\}$ **Equation 3-14**

$i_{r,e}$ denotes the train number on route r passes the junction area in the event e.

$i_{r,e} \in \{0,1,2,\cdots,N_r\}, r \in R, e \in E$ **Equation 3-15** where $i_{r,e} = 0$ means no train on route r passes the junction area in event e.

r_e denotes the route which is set for the approaching train to pass the junction area in event e.

$$r_e = r \quad s.t.\ i_{r,e} \neq 0,\ \ e \in E \qquad \textbf{Equation 3-16}$$

Train arrival time at boundaries b of control region is ordered with the set

$$\left\{t_{i_r}^{b} \middle| i_r = 1,2,\cdots,N_r;\ r = 1,2,\cdots,|R|\right\} \qquad \textbf{Equation 3-17}$$

Where

$$t_{i_r+1}^{b} > t_{i_r}^{b} \quad i_r = 1,2,\cdots,N_r - 1;\ r = 1,2,\cdots,|R| \qquad \textbf{Equation 3-18}$$

In the beginning of event e, for each route r, the number of trains that have passed junction area is $n_{r,e}$.

$$n_{r,1} = 0,\ r \in R \qquad \textbf{Equation 3-19}$$

Equation 3-19 means that no train has passed the junction area on each route r at the beginning of the first event.

For each event e,

$$i_{r,e} \in \left\{0, n_{r,e} + 1 \middle| r \in R; e \in E\right\} \qquad \textbf{Equation 3-20}$$

Binary route setting decision variable $d_{r,e}$ is defined as:

$$d_{r,e} = \begin{cases} 1, & if\ the\ route\ r\ is\ set\ for\ the\ train\ to\ pass\ in\ event\ e, \\ & where\ r \in R, e \in E. \\ 0, & otherwise. \end{cases} \qquad \textbf{Equation 3-21}$$

and

$$\sum_{r \in R} d_{r,e} = 1 \qquad r \in R, e \in E \qquad \textbf{Equation 3-22}$$

$\sum_{e \in E} d_{r,e} = N_r \qquad r \in R, e \in E$ **Equation 3-23**

Equation 3-22 means only one route can be set for the approaching trains in each event e.

Equation 3-23 ensures that the route on which all of the approaching trains have passed the junction area will not be set.

Then

$r_e = \sum_{r \in R}(r * d_{r,e}) \qquad e \in E$ **Equation 3-24**

$i_{r,e} = d_{r,e} * \sum_{r \in R}((n_{r,e} + 1) * d_{r,e}) \qquad r \in R, e \in E$ **Equation 3-25**

Binary headway variable $h_{r,r'}$ is defined as:

$$h_{r,r'} = \begin{cases} 1, & if\ headway\ must\ be\ kept\ for\ trains\ on\ route\ r\ with\ the\ trains \\ & on\ route\ r'\ in\ consecutive\ events, where\ r, r' \in R. \\ 0, & otherwise. \end{cases}$$

Equation 3-26

The operational train headway on route r and r' in consecutive events is denoted by $t^H_{r,r'}$.

We define that $t^p_{i_{r,e}}$ is the rescheduled arrival time of the $i_{r,e}$th train on route r at junction point p in the event e, $t^{b,p}_{i_r}$ is the minimum running time of the train i_r on route r from boundary b to junction point p, t_{init} is the time constraint for the first train passing through the junction in the rescheduling control window.

Then

$t^p_{i_{r,e}} \geq t^b_{i_r} + t^{b,p}_{i_r} \qquad i_{r,e} \neq 0, r \in R, e \in E$ **Equation 3-27**

Equation 3-27 ensures that the minimum running time of trains from control boundary to junction points can be kept, and

$$\begin{cases} t^{p}_{i_{r,e}} \geq t_{init} & e = 1 \\ t^{p}_{i_{r,e}} \geq \left(t^{p}_{i_{r',e-1}} + t^{H}_{r_e,r_{e-1}}\right) * h_{r_e,r_{e-1}} & e > 1 \end{cases} \quad i_{r,e}, i_{r',e-1} \neq 0;\ e \in E;\ r, r' \in R$$

Equation 3-28

Equation 3-28 ensures the minimum headway of trains can be kept in terms of the train operation and control constraints.

For event $e > 1$,

$$n_{r,e} = n_{r,e-1} + d_{r,e-1} \qquad r \in R, e > 1 \;\&\; e \in E$$ **Equation 3-29**

The objective function presented in this thesis is to minimise the Weighted Average Delay (WAD) shown in Equation 3-11. For the JRM, the detailed form of Equation 3-11 can be transformed into the form of Equation 3-30.

$$WAD = \sum_{e \in E} \sum_{r \in R} \sum_{i \in N_r} (\omega_{i_r} * \left| t^{p}_{i_{r,e}} - t^{p^0}_{i_r} \right| * d_{r,e}) / \sum_{r \in R} \sum_{i \in N_r} \omega_{i_r}$$ **Equation 3-30**

where ω_{i_r} is the weight of the train i_r on route r, the value of which indicates the priority of the train. $t^{p^0}_{i_r}$ is the arrival time of the train i_r on route r at junction point p in the nominal timetable. The nominal timetable is the given timetable for practical daily train operation. The value of WAD defined in Equation 3-30 reflects the closeness of the rescheduled timetable to the nominal timetable for the trains passing junction areas.

The aim of train rescheduling in junction areas is to find the optimal route setting decision $d^{*}_{r,e}$ and train arrival time at junction point $t^{p^*}_{i_{r,e}}$ to minimise the WAD presented in Equation 3-30.

The presented problem above is a Mixed Integer Programming (MIP) problem, the variables need to be optimised are: $d_{r,e}$ and $t^{p}_{i_{r,e}}$. The values of all the other variables are given or can be calculated in practical railway operations. One issue which needs to be clarified here is that the constraints shown in the equations above are the basic constraints which need to be complied with in practical railway operations. For some of the particular circumstances of railway operations, some more constraints need to be added in the model JRM, according to actual operation and control requirements.

As presented above, the presented problem of train rescheduling in the junction area can be formulated with a MIP as follows:

Objective:

Minimise

$$WAD = \sum_{e\in E}\sum_{r\in R}\sum_{i\in N_r}(\omega_{i_r} * \left|t^{p}_{i_{r,e}} - t^{p^0}_{i_r}\right| * d_{r,e}) / \sum_{r\in R}\sum_{i\in N_r}\omega_{i_r}$$

Subject to:

$$\textstyle\sum_{r\in R} d_{r,e} = 1 \qquad r \in R, e \in E$$

$$\textstyle\sum_{e\in E} d_{r,e} = N_r \qquad r \in R, e \in E$$

$$t^{p}_{i_{r,e}} \geq t^{b}_{i_r} + t^{b,p}_{i_r} \qquad i_{r,e} \neq 0, r \in R, e \in E$$

$$\begin{cases} t^{p}_{i_{r,e}} \geq t_{init} & e = 1 \\ t^{p}_{i_{r,e}} \geq \left(t^{p}_{i_{r',e-1}} + t^{H}_{r_e,r_{e-1}}\right) * h_{r_e,r_{e-1}} & e > 1 \end{cases} \qquad i_{r,e}, i_{r',e-1} \neq 0;\ e \in E;\ r, r' \in R$$

and

$$r_e = \textstyle\sum_{r\in R}(r * d_{r,e}) \qquad e \in E$$

$$i_{r,e} = d_{r,e} * \textstyle\sum_{r\in R}((n_{r,e} + 1) * d_{r,e}) \qquad r \in R, e \in E$$

$$n_{r,1} = 0,\ r \in R$$

$n_{r,e} = n_{r,e-1} + d_{r,e-1} \qquad r \in R, e > 1 \,\&\, e \in E$

The presented MIP problem can be divided into two levels: the upper level is a Combinatorial Programming problem of optimising the binary route setting decision $d^*_{r,e}$, the lower level is a Constraint Nonlinear Programming problem of finding the optimal train arrival time at junction point $t^{p^*}_{i_{r,e}}$.

3.3 Explanation of Junction Rescheduling Model (JRM) with a Typical Case

In the last section the formulation of JRM is presented with a MIP. To understand the JRM better, the model is explained in detail with a typical case in this section.

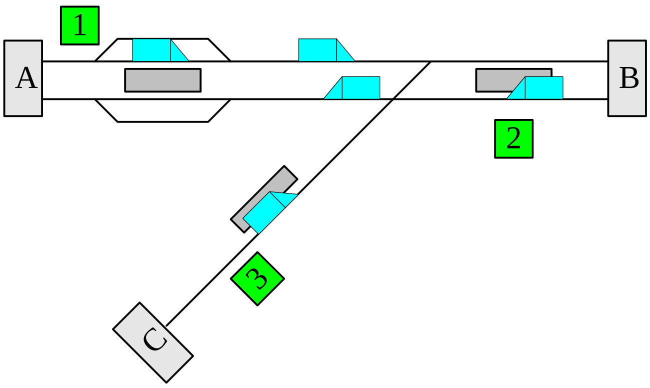

Figure 3-2 Layout of a typical railway junction

Figure 3-2 shows the layout of a typical railway junction, where there are trains on a single track section approaching the junction which converge with the trains on the double track section. As shown in Figure 3-2, the junction has three approaching routes: route 1 (A->B), route 2 (B->A) and route 3 (C->B). The approaching routes can be denoted in the set as shown in Equation 3-31:

$r \in R = \{1,2,3\}, \ \ |R| = 3$ **Equation 3-31**

It is assumed that five trains are approaching the junction area on different routes per hour, here one hour is chosen as a control window.

$$T_1 = \{1,2\},\ \ N_1 = 2$$ **Equation 3-32**

$$T_2 = \{1,2\},\ \ N_2 = 2$$ **Equation 3-33**

$$T_3 = \{1\},\ \ N_3 = 1$$ **Equation 3-34**

In practical railway operation, each train is assigned a unique train description number like 20022, 11033, 30221 etc. Here the number in set T denotes the train approaching the junction in terms of the boundary arrival time of the trains, as shown in Equation 3-17 and Equation 3-18.

Then

$$e \in E = \{1,2,\cdots,|E|\},\ \ |E| = 5$$ **Equation 3-35**

The optimal binary route setting decision $d^*_{r,e}$ must satisfy the constraints of Equation 3-22 and Equation 3-23. A possible solution of binary route setting decision $d^*_{r,e}$ is shown as follows:

$$d_{r,e} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$ **Equation 3-36**

A table, as below, is used to explain $d_{r,e}$ matrix. The value of r_e is calculated according to Equation 3-24 and listed in Table 3-2.

$$r_e = [1 \quad 3 \quad 2 \quad 1 \quad 2]$$ **Equation 3-37**

	$r \backslash e$	1	2	3	4	5
$d_{r,e}$	1	1	0	0	1	0
	2	0	0	1	0	1
	3	0	1	0	0	0
r_e		1	3	2	1	2

Table 3-2 Representation of $d_{r,e}$ and r_e

Based on Equation 3-19 and Equation 3-29, the value of $n_{r,e}$ can be reached, as shown in Table 3-3.

$$n_{r,e} = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$ **Equation 3-38**

	$r \backslash e$	1	2	3	4	5
$n_{r,e}$	1	0	1	1	1	2
	2	0	0	0	1	1
	3	0	0	1	1	1
$\sum_{r \in R} n_{r,e}$		0	1	2	3	4

Table 3-3 Representation of $n_{r,e}$

According to Equation 3-25, the value of $i_{r,e}$ is calculated and listed in Table 3-4.

$$i_{r,e} = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$ **Equation 3-39**

$$r_e = [1 \quad 3 \quad 2 \quad 1 \quad 2]$$ **Equation 3-40**

	$r \backslash e$	1	2	3	4	5
$i_{r,e}$	1	1	0	0	2	0
	2	0	0	1	0	2
	3	0	1	0	0	0
r_e		1	3	2	1	2

Table 3-4 Representation of $i_{r,e}$

According to the layout of the demo railway junction shown in Figure 3-2, there is no conflict between the trains on route 1 and 2. So here $h_{r,r'}$ is defined as follows.

$$h_{r,r'} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \textbf{Equation 3-41}$$

And assuming that

$$t^H_{r,r'} = \begin{bmatrix} 100 & 0 & 120 \\ 0 & 100 & 110 \\ 120 & 110 & 100 \end{bmatrix} \qquad \textbf{Equation 3-42}$$

$t^b_{i_r}$, $t^{b,p}_{i_r}$ and t_{init} are given values in the JRM.

Besides the optimal binary route setting decision variable $d^*_{r,e}$, for all $i_{r,e} \neq 0$, the optimal continuous trains arrival time variable $t^{p^*}_{i_{r,e}}$ for the objective function must be found under the constraints in Equation 3-27 and Equation 3-28. For every possible route setting decision variable $d_{r,e}$, the lower level of the presented MIP problem for optimising trains arrival time variable $t^p_{i_{r,e}}$ is a Constrained Nonlinear Programming problem.

Regarding the presented example, there are five trains going through the junction area in each control window. The number of possible route setting decisions is $5!/(2! * 2! * 1!) = 30$. Accordingly in the presented example, the possible r_e which denote the route which is set for the approaching train to pass the junction area in event e are listed in Table 3-5.

30 Possible r_e		
[1 1 2 2 3]	[2 1 1 2 3]	[3 1 1 2 2]
[1 1 2 3 2]	[2 1 1 3 2]	[3 1 2 1 2]
[1 1 3 2 2]	[2 1 2 1 3]	[3 1 2 2 1]
[1 2 1 2 3]	[2 1 2 3 1]	[3 2 1 1 2]
[1 2 1 3 2]	[2 1 3 1 2]	[3 2 1 2 1]
[1 2 2 1 3]	[2 1 3 2 1]	[3 2 2 1 1]
[1 2 2 3 1]	[2 2 1 1 3]	
[1 2 3 1 2]	[2 2 1 3 1]	
[1 2 3 2 1]	[2 2 3 1 1]	
[1 3 1 2 2]	[2 3 1 1 2]	
[1 3 2 1 2]	[2 3 1 2 1]	
[1 3 2 2 1]	[2 3 2 1 1]	

Table 3-5 Possible route setting r_e for presented example

The process of route setting for the 5 approaching trains can be represented by the decision tree shown in Figure 3-3.

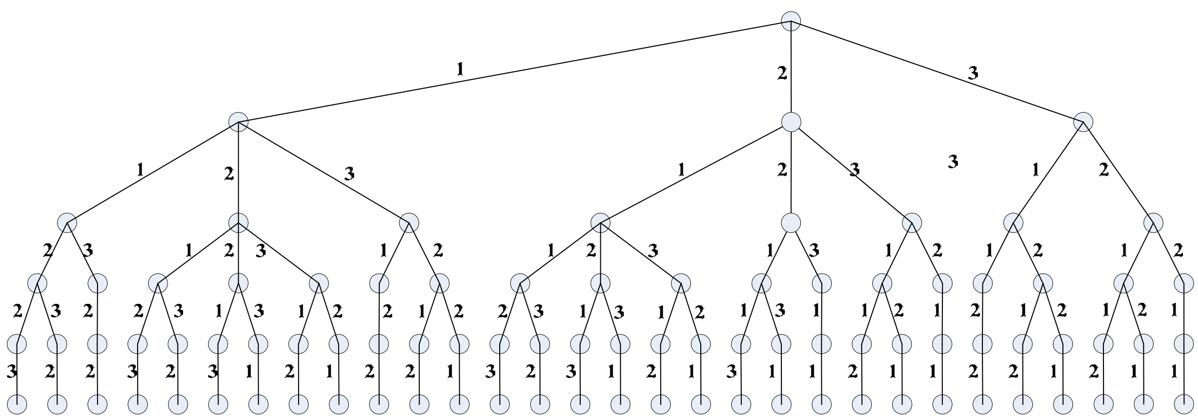

Figure 3-3 Decision tree representing the process of route setting for 5 approaching trains

From the decision tree, we can see that as the number of approaching trains in a control window increases, the number of possible route setting decisions will rise sharply. The upper level of the JRM problem has been proven to be a NP-hard problem, which means it is quite hard to find the optimal solution for the problem in Polynomial time (Garey and Johnson 1979). In addition, for every possible route setting decision $d_{r,e}$, the lower level of presented MIP problem for optimising train arrival times variable $t_{i_{r,e}}^{p}$ is a Constraint Nonlinear Programming problem. It also increases the difficulty of solving the JRM problem. Efficient algorithms for solving JRM problems are required.

3.4 Conclusions

To better understand train rescheduling problems in junction areas, it is necessary to build mathematical models for the proposed problems. In this chapter, the general concepts and definitions of mathematical programming were introduced. Then the formulation of the train rescheduling problems in junction areas with mathematical programming technologies were presented in the form of a proposed mathematical model, named as JRM by the author. The example of train rescheduling in a simple junction area demonstrates the use of the proposed JRM.

In the next chapter, the problem solving of JRM is studied. Approaches which can be used for solving the JRM problem are introduced and an innovative algorithm based on Differential Evolution algorithm, named as DE_JRM, is presented, as well as the performance evaluation of the algorithm DE_JRM.

Chapter 4. An Innovative Algorithm DE_JRM for Solving JRM Problems

The formulation of JRM problems has been presented in detail in the previous chapter. The proposed JRM problem is an MIP problem, which has integral route setting decision and continuous train arrival time variables. It is challenging to find an efficient algorithm to solve the proposed JRM problems. An innovative algorithm named DE_JRM is presented in this chapter. The algorithm DE_JRM is derived from a general Differential Evolution (DE) algorithm, and improved to be an efficient tool for solving proposed JRM problems. Firstly, the general concepts and definitions of DE algorithms are introduced. Then the improvements of the DE_JRM are described in detail. To validate the efficiency of the proposed algorithm, the performance of DE_JRM is evaluated with a case study chosen from Thameslink routes, and the evaluation results are analysed.

4.1 Innovative Algorithm DE_JRM

4.1.1 New Algorithm Requirement Consideration

The JRM problem presented in the previous chapter is a mixed integer programming (MIP) problem which can be divided into two levels: the upper level is a Combinatorial Programming problem of optimising the binary route setting decision $d_{r,e}^{*}$; the lower level is a Constrained Nonlinear Programming problem of finding the optimal train arrival times $t_{i_{r,e}}^{p^*}$ at junction point. Generally, to find the optimal solutions of presented JRM problems, the mathematical method as follows can be applied:

First, for upper level combinatorial programming problems of optimising the binary route setting decision, enumerate all of the possibilities of the binary route setting decisions.

In terms of every possible binary route setting decision, apply methods such as Sequential Quadratic Programming (SQP) etc. to solve the lower level constrained nonlinear programming problem, and get the optimal train arrival times at junction point under every possible binary route setting decision. Compare the optimal train arrival times and acquire the optimal solution, then the optimal route setting decision $d^*_{r,e}$, and the optimal train arrival time $t^{p^*}_{i_{r,e}}$ will be found. The main shortcoming of this method is that the computational time increases exponentially with the size of the problem. It is not possible to apply this method to large real time applications, for example, real time junction rescheduling problems here.

As seen from the JRM, the problem has the following features:

1. The search space is very large and increases exponentially as the size of problem increases.

2. The search space is a mix of discrete and continuous spaces, and is known to be not smooth or even not well understood.

3. The problem presented refers to a real time application, and the solution must be found under required time limits.

4. Nearly optimised solutions can be accepted in the application.

Considering all of the features above, an innovative algorithm based on a Differential Evolution (DE) algorithm for solving JRM problems, named DE_JRM, is presented. In the next sections, the general Differential Evolution algorithm is introduced, and the details of the algorithm DE_JRM are described. The performance of DE_JRM will also be evaluated at the end of this chapter.

4.1.2 Introduction of Differential Evolution

Direct search methods are widely used for solving most optimisation problems, the objective functions of which are nonlinear or non-differential. The general principle of these methods is to generate variations of the parameter vectors. Once a variation is generated, a decision needs to be made as whether or not to accept the newly derived parameters. Some of the methods use the greedy criteria for the decision making process, which makes the algorithms convergence sufficiently fast, but could lead the algorithms to be trapped in a local optimum. Others introduce probabilities for the decision making process and permit algorithms to search along other directions with certain probabilities, and this helps to escape from a local optimum.

Differential Evolution is also classified as a direct search method, and was first introduced by Rainer Storn and Kenneth Price in 1997 (Storn and Price 1997). It has been regarded as a simple but efficient optimisation approach which can reliably converge to the global optimum of optimisation problems with sufficiently fast convergence speed (Mayer, Kinghorn et al. 2005; Zhang and Sanderson 2007; Takahama and Sakai 2009).

DE is also classified as an evolutionary algorithm, which includes three operations: Mutation, Crossover and Selection. Considering the problem presented in Equation 3-8 $\begin{cases} Minimise\ f(x) \\ subject\ to: \\ g_i(x) \leq 0 \ \ i = 1, \cdots, m \\ h_j(x) = 0 \ \ j = 1, \cdots, n \\ x \in \mathbb{R}^n \end{cases}$, which is a typical constrained nonlinear programming problem, we can let all the unknown variables, which are to be optimised, depend on the real-valued parameter shown in

$$\{x_j | j = 1,2, \dots, D\} \qquad \textbf{Equation 4-1}$$

The optimisation of the problem is to vary the D-dimensional parameter $X = (x_1, x_2, \cdots, x_D)^T$ until the objective value is minimised and all the inequality and equality constraints are met.

As a parallel direct search method, we also define parameter vectors $\{X_{j,g} | j = 1,2, \dots, NP\}$ as a population for each generation g, and NP is the constant population size. Every $X_{j,g}$ can also be called an individual of the population in generation g. First an initial population needs to be chosen randomly if nothing is known of the problem. In practice, the unknown variables of most of the problems have an upper and lower bound $x_j \in [x_j^L, x_j^U]$. The basic principle of generating the initial population is to maximise the variety of the parameter vectors in all searching spaces, and generally the initial population can be derived according to a uniform distribution or normal distribution between the upper and lower bounds. The main idea of DE is a scheme to generate trial parameter vectors until satisfied solutions can be found in terms of the objective functions. Similar to other evolutionary algorithms, DE includes three operations: Mutation, Crossover and Selection. The general flow chart of the DE method is shown in Figure 4-1. The variant work of parameter vectors normally starts from Mutation.

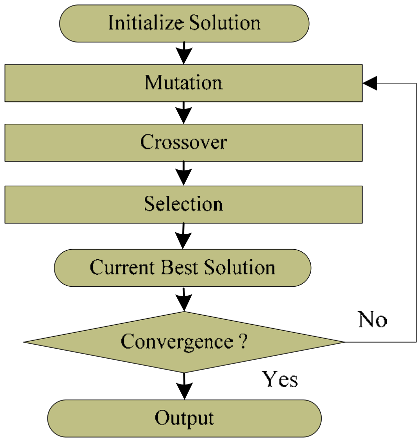

Figure 4-1 Flow Chart of Classic DE method

1) Mutation

Firstly, trial parameter vectors $\{V_{j,g} | j = 1,2, \dots, NP\}$ are generated with the Mutation operation according to the frequently used strategies as follows,

(1) Mutation Strategy 1:

$$V_{j,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}) \qquad \textbf{Equation 4-2}$$

where

$r1, r2, r3 \in \{1,2, \cdots, NP\}, r1, r2, r3 \neq j$, and $F > 0$.

$r1, r2, r3$ are integers and randomly chosen from the set $\{1,2, \cdots, NP\}$. F is called the mutation factor to control the amplification of the differential variation $(X_{r2,g} - X_{r3,g})$. Usually the value of F is chosen from the interval $(0,1+)$. In classic DE, F is set to be a constant. In some modified adaptive DE algorithms (Abbass 2002; Liu and Lampinen 2005; Qin and Suganthan 2005; Teo 2006; Zhang and Sanderson 2007; Wen, Lu et al. 2008), F is varied associated with index j in the optimisation process. In theory, there is no upper limit on F, but in practice effective values of F are seldom chosen greater than

1. The selection of values of Mutation factors could have an influence on the performance of DE algorithms (Price, Storn et al. 2005).

(2) Mutation Strategy 2:

Mutation strategy 2 generates trial parameter vectors $V_{j,g}$ in the following way:

$$V_{j,g} = X_{j,g} + F_1 \cdot \left(X_{best,g} - X_{j,g}\right) + F_2 \cdot (X_{r2,g} - X_{r3,g}) \qquad \textbf{Equation 4-3}$$

where $X_{best,g}$ is the best vector in the current generation. In Mutation Strategy 2, there is an additional mutation factor to control the variation $\left(X_{best,g} - X_{j,g}\right)$, and it is used to enhance the greediness of the Mutation operation with the introduction of the current best vector $X_{best,g}$. This can make the convergence faster compared with Mutation Strategy 1 for most applications.

(3) Mutation Strategy 3:

In some of DE, the trial parameter vectors are generated by mutation operation using Equation 4-4.

$$V_{j,g} = X_{best,g} + F \cdot (X_{r2,g} - X_{r3,g}) \qquad \textbf{Equation 4-4}$$

It can be seen that Mutation Strategy 3 is greedier than Mutation Strategy 1 and 2. Normally this can be used for the optimisation problems where the global optimum is relatively easy to find.

Small scale tests have been trialled by the author and, Mutation strategy 2 is selected in the Mutation operation in order to balance the convergence speed and the diversity of the population distribution in this thesis. The basic principle for value selection of mutation factor F can be found in (Price, Storn et al. 2005).

2) *Crossover*

In order to increase the diversity of trial parameter vectors, there is usually a crossover operation shown in Equation 4-5 after mutation to generate new trial parameter vectors $U_{j,g}$ in DE.

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & if\ rand_i(0,1) \leq CR\ or\ i = randint_j(1,D) \\ x_{i,j,g} & otherwise \end{cases}$$ **Equation 4-5**

where $rand_i(0,1)$ is a uniform random number in the interval $(0,1)$, $randint_j(1,D)$ is a integer randomly chosen from $(1,D)$ for each j. $CR \in [0,1]$ is the crossover probability, which is a constant in classic DE algorithms. The value selection principle of CR can refer to (Price, Storn et al. 2005). In many adaptive DE algorithms, CR is varied associated with index j.

3) *Selection*

The last operation of classic DE algorithms is Selection, which is used to select the better solutions from all the trial parameter vectors $U_{j,g}$ generated through Mutation and Crossover. These solutions become the new parent individuals $X_{j,g+1}$ that then form a new generation according to Equation 4-6. $f(\cdot)$ is the objective function, which is defined in mathematical programming problems.

$$X_{j,g+1} = \begin{cases} U_{j,g} & if\ \ f(U_{j,g}) < f(X_{j,g}) \\ X_{j,g} & otherwise \end{cases}$$ **Equation 4-6**

The process of DE will stop until the algorithms "converge", which is normally defined as the $X_{best,g}$ satisfying the optimal requirements of users or no further better solutions can be found in a number of generations. In real time applications, the algorithms need to stop when the upper time limit is met.

4.1.3 Improved DE Algorithm DE_JRM for Solving JRM Problems

In the last section, the basic procedure and operations of classic DE are introduced. Regarding the JRM presented in Chapter 3.2 , which is a Mixed Integer Programming problem. Because of the features of JRM, the classic DE algorithms introduced in the previous section cannot be used directly for solving the JRM problem. There are two variable vectors which need to be optimised in the JRM; they are the binary route setting decisions $d^*_{r,e}$ and train arrival times at junction point $t^{p^*}_{i_{r,e}}$, and the constraints presented in Equation 3-22, Equation 3-23, Equation 3-27 and Equation 3-28 must be satisfied.

In the JRM, the binary route setting decisions $d_{r,e}$ is determined after the train arrival times at junction point $t^{p}_{i_{r,e}}$ are decided. For solving the JRM, the parameter $\{X = t^{p}_{i_{r,e}} | r \in R; e \in E\}$ is defined, which need to be varied until the objective function presented in Equation 3-30 is minimised and all the constraints in JRM are met. If applying the classic DE directly to solve the JRM problem, the stochastic Mutation and Crossover processes cannot ensure that the generated trial parameter vectors comply with the constraints that are typical for JRM, so that most of the trial parameter vectors generated are invalid. Due to the lack of valid solution trial parameter vectors, it is hard to evolve better solutions for JRM using classic DE algorithms. There has been much research on developing improved DE algorithms (Das, Abraham et al. 2008; Sayah and Zehar 2008; Yang, Dong et al. 2008), but they are not suitable for solving the proposed JRM problems.

In order to create an algorithm that is suited to solving JRM problems, an additional process/operation into the DE algorithm, named ‘Modification’ operation is developed in this thesis, thus creating DE_JRM. The main function of Modification is to modify invalid solution trial parameter vectors so that they become valid in terms of the constraint rules of JRM. The modification operation is based on the greedy rules with stochastic process presented in the next section. Normally greedy rules in Modification could decrease the diversity of trial parameter vectors, but integrated with stochastic Mutation and Crossover operations, these rules increase the number of valid trial parameter vectors largely and enhance the variety of trial parameter vectors during the process of DE. On the basis of large numbers of valid trial parameter vectors in every generation, DE_JRM can evolve improved solutions from generation to generation which converge after numbers of generations. The addition of 'Modification' creates an effective tool for solving JRM problems. The flow chart of DE_JRM is shown in Figure 4-2. The modified trial parameter vectors $M_{j,g}$ will be generated by Modification operation based on the input parameter vectors $U_{j,g}$, which is the output of the Crossover operation. The Selection operation is carried out according to Equation 4-7. The details of the Modification operation are as follows.

$$X_{j,g+1} = \begin{cases} M_{j,g} & if \;\; f(M_{j,g}) < f(X_{j,g}) \\ X_{j,g} & otherwise \end{cases}$$ **Equation 4-7**

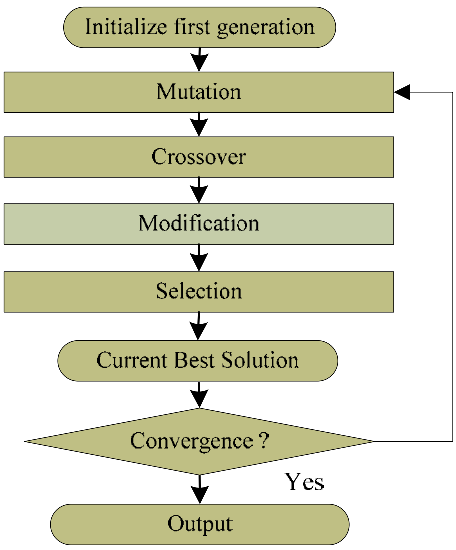

Figure 4-2 Flow chart of DE_JRM

In Modification operations, three processes are defined based on specific greedy rules to modify any invalid trial parameter vectors so as to meet the constraints in JRM. The processes defined in the Modification operation are shown in Figure 4-3.

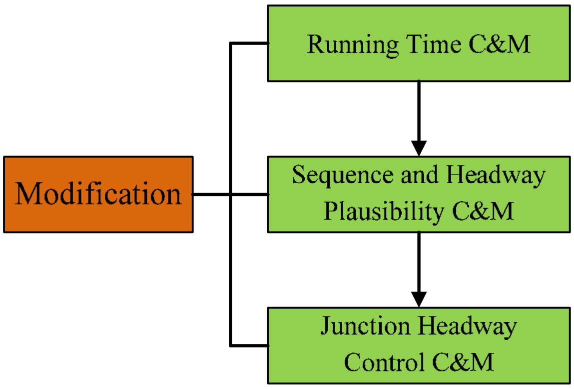

Figure 4-3 Processes in Modification operation

1) Running Time Check and Modification:

The main task of running time check and modification is to check whether the trial parameter vectors comply with the minimum running time constraints in JRM shown in Equation 3-27, and to modify the invalid parameter vectors according to the rules as shown in Equation 4-8.

$$IF\ t^{p}_{i_{r,e}} < t^{b}_{i_r} + t^{b,p}_{i_r},\ Then\ t^{p}_{i_{r,e}} = t^{b}_{i_r} + t^{b,p}_{i_r} \qquad i_{r,e} \neq 0, r \in R, e \in E \qquad \textbf{Equation 4-8}$$

2) Sequence and Headway Validity Check and Modification:

Sequence and headway validity check and modification is used to ensure that the trial parameter vectors generated by the stochastic Mutation and Crossover processes are valid in terms of the train sequence restrictions at control boundaries. For example, if there is only one route for trains from point A to point B, then the train arrival sequence at point B must be the same as that at point A. The modification rule for sequence and headway validity checking is presented in Equation 4-9. The train headway on the converging routes is also checked and modified. The minimum train headway must be assured in terms of the signalling systems and operation modes. This check is applied for all trains on the same converging route.

$$\begin{cases} IF \quad t^{b}_{i_r} > t^{b}_{(i-1)_r} \; and \; t^{p}_{i_{r,e}} < t^{p}_{(i-1)_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ THEN \quad t^{p}_{i_{r,e}} = t^{p}_{(i-1)_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ \qquad r_e = r_{e-1};\; e > 1,\; e \in E;\; r \in R \end{cases} \quad \textbf{Equation 4-9}$$

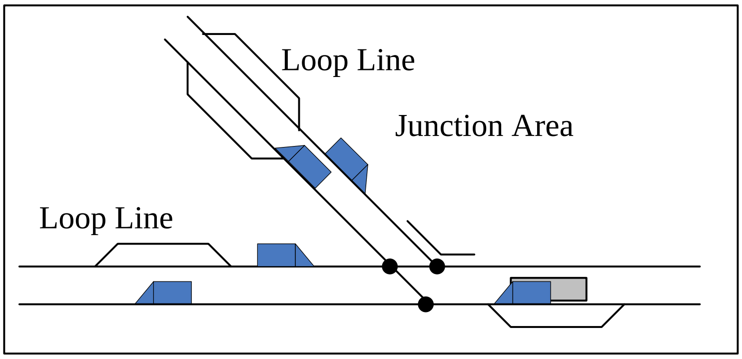

Figure 4-4 Loop Lines on Approaching Route to Junction Area

As for the rule of sequence headway validity check and modification, if there is loop as shown in Figure 4-4, which means the overtaking is possible for the trains approaching the junction, then the rule needs to be changed to the rule shown in Equation 4-10.

$$\begin{cases} IF \quad t^{p}_{(i-1)_{r,e-1}} < t^{p}_{i_{r,e}} < t^{p}_{(i-1)_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ THEN \quad t^{p}_{i_{r,e}} = t^{p}_{(i-1)_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ \qquad r_e = r_{e-1};\; e > 1,\; e \in E;\; r \in R \end{cases} \quad \textbf{Equation 4-10}$$

3) Junction Headway Control Check and Modification:

Junction control headway checking is used to check whether the headway of the trains approaching the junction point from different origins is kept or not. The rules for junction headway control check and modification are vital to traffic management in junction areas;

they will affect which train from which approaching route will pass the junction first, as well as the train sequence after the junction areas. In the junction headway control check and modification process, three different strategies are applied to ensure that the train headway at junction point is kept.

Strategy 1: First Come First Served (FCFS)

In Strategy 1, a First Come First Served (FCFS) based rule presented in Equation 4-11 is used to adapt the invalid parameter vectors to be valid in terms of the headway constraints in JRM. This strategy is to check the headway of trains passing the junction point from different approaching routes, whether against train headway constraints or not, and modify the invalid individual parameter vectors in terms of FCFS.

$$\begin{cases} IF\ t^{p}_{i_{r,e-1}} < t^{p}_{i_{r,e}} < t^{p}_{i_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} and\ h_{r_e,r_{e-1}} \neq 0 \\ THEN \quad t^{p}_{i_{r,e}} = t^{p}_{i_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ \qquad\qquad r_e \neq r_{e-1};\ e > 1,\ e \in E;\ r \in R \end{cases} \quad \textbf{Equation 4-11}$$

Strategy 2: Priority based Modification

In Strategy 2, the rule used for junction headway control check and modification is based on the priority of the approaching trains, which is denoted by train weighting ω_{i_r}. When the trains approaching to the junction point from different routes have potential conflicts, the higher weighting the train is, the more priority of passing the junction the train has. The details of Priority based Modification are presented in Equation 4-12.

$$\begin{cases} IF\ t^{p}_{i_{r,e-1}} < t^{p}_{i_{r,e}} < t^{p}_{i_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} and\ h_{r_e,r_{e-1}} \neq 0\ and\ \omega_{i_{r,e}} > \omega_{i_{r,e-1}} \\ THEN\ t^{p}_{i_{r,e-1}} = t^{p}_{i_{r,e}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ OTHERWISE \quad t^{p}_{i_{r,e}} = t^{p}_{i_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ \qquad\qquad\qquad\qquad r_e \neq r_{e-1};\ e > 1,\ e \in E;\ r \in R \end{cases}$$

Equation 4-12

Strategy 3: Weighted Delay based Modification

Strategy 3 introduces a Weighted Delay based Modification. The rule of junction headway control check and modification in this strategy depends on the potential weighted delay at the junction point due to the junction headway control. In terms of the different train sequence for passing junctions, the potential weighted delay of trains will be calculated and compared, and the train sequence which causes the lower weighted delay will be chosen. The details of the rule are presented in Equation 4-13.

We define variable $\tau_{i_{r,e}}$ to denote the delay time of train $i_{r,e}$ at the junction point if the train $i_{r,e}$ was rescheduled to wait before the junction point until the train $i_{r,e+1}$ from other approaching routes passes the junction point.

$$\begin{cases} IF\ t^{p}_{i_{r,e-1}} < t^{p}_{i_{r,e}} < t^{p}_{i_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} and\ h_{r_e,r_{e-1}} \neq 0 \\ \qquad\qquad and\ \omega_{i_{r,e}} * \tau_{i_{r,e}} > \omega_{i_{r,e-1}} * \tau_{i_{r,e-1}} \\ THEN\ t^{p}_{i_{r,e-1}} = t^{p}_{i_{r,e}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ OTHERWISE \quad t^{p}_{i_{r,e}} = t^{p}_{i_{r,e-1}} + t^{H}_{r_e,r_{e-1}} * h_{r_e,r_{e-1}} \\ \qquad\qquad r_e \neq r_{e-1};\ e > 1,\ e \in E;\ r \in R \end{cases} \qquad \textbf{Equation 4-13}$$

In the junction headway control check and modification process, three different strategies have been presented. Every strategy has different greedy rules for headway control modification, which may be more effective for certain types of JRM problems. Because of the complexity of railway operation, it is very hard to decide which strategy is better for different scenarios in JRM. Thus a probability based hybrid method for selecting the junction headway control check and modification strategies in the Modification operation is presented.

Let $p_j^G(h)$ denote the selecting probability of the junction headway control check and modification strategy h for parameter vector j in generation G. At every generation G, for every individual parameter vector j, one strategy h will be chosen for headway control check and modification according to the selecting probability $\{p_j^G(h)|h = 1,2,3\}$. Here $p_j^G(1), p_j^G(2), p_j^G(3)$ denote the selecting probability of Strategy 1 (FCFS), Strategy 2 (Priority based Modification) and Strategy 3 (Weighted Delay based Modification) respectively. The selecting probabilities $\{p_j^G(h)|h = 1,2,3\}$ will vary in the process of DE_JRM.

In the initialisation, for every individual parameter vector j, we assign the initial select probability of the junction headway control check and modification strategies $\{p_j^0(h) = 1/3|h = 1,2,3\}$. During the process of DE_JRM, the selecting probabilities will be adjusted according to the rules as follows.

In the Selection operation, for each individual j in generation G, if the trial parameter vector $M_{j,g}$ generated after Modification is chosen in the Selection operation to be an new individual in $X_{j,g+1}$, then the selecting probability of the selected strategy h for junction headway control check and modification will be increased in generation $G + 1$ according to Equation 4-14 as follows, while the selecting probability of other strategies will be decreased to keep the probability distribution constraints $\sum_k p_j(k) = 1 \; and \; p_j(k) > 0$.

$$\begin{cases} p_j^{G+1}(h) = p_j^G(h) + \left(1 - p_j^G(h)\right) * \gamma \\ p_j^{G+1}(l) = p_j^G(l) - p_j^G(l) * \gamma \quad \forall l \neq h \end{cases}$$

Equation 4-14

where $\gamma \in (0,1)$, which is a control parameter defined to regulate the probability increment. The effect of control parameter γ can be seen from Equation 4-14. When $\gamma \to 0$, then $p_j^{G+1}(h) \to p_j^G(h)$, which means the selecting probability of strategy h for individual j will stay nearly the same in the next generation. When $\gamma \to 1$, then $p_j^{G+1}(h) \to 1$, that means in the next generation the selecting probability of strategy h will be nearly 1. In this thesis, the control parameter γ is set to 0.2.

If the trial parameter vector $M_{j,g}$ generated after Modification is not chosen in the Selection operation to be an new individual in $X_{j,g+1}$, then the selecting probability of strategies for junction headway control check and modification will be adjusted in generation $G+1$ according to Equation 4-15.

$$\begin{cases} p_j^{G+1}(h) = p_j^G(h) - p_j^G(h) * \gamma \\ p_j^{G+1}(l) = p_j^G(l) + p_j^G(l) * \gamma \quad \forall l \neq h \end{cases} \qquad \textbf{Equation 4-15}$$

In terms of the stochastic method presented above, these three strategies are applied in the Modification operation for junction headway control check and modification.

The Modification operation mainly includes the rules presented as above. However, some specific rules can also be added into the Modification operation for the specific railway operation requirements. The solution individuals generated by Mutation and Crossover can be regarded to be feasible in terms of real-life railway control and operation rules, once the Modification operation has been applied.

```
Initialize first generation;
Initialize selecting probabilities;
Calculate the initial cost (WAD);
While ( ! current best solution converge)
        Mutation;
        Crossover;
        Running time C&M;
        Sequence and Headway Validity C&M;
        Select junction headway control strategy;
        Junction headway control C&M;
        Calculate the current cost;
        Selection;
        Update selecting probabilities;
        Calculate the current best solution;
End.
Output the best solution;
```

Figure 4-5 Pseudo-code of DE_JRM

Integrating the mixed method in the Modification operation, the pseudo-code of presented DE_JRM algorithm is shown in Figure 4-5. In the next sections, the DE_JRM algorithm is evaluated in terms of two criteria: goodness of output solution and computational time. In this thesis, a systematic approach for the performance evaluation of DE_JRM is proposed.

4.2 Performance Evaluation of DE_JRM

4.2.1 Systematic Approach

In the last section, an improved DE algorithm, DE_JRM for solving JRM problems was presented in detail. In this section, the performance of DE_JRM is evaluated using a systematic approach described in the following sections.

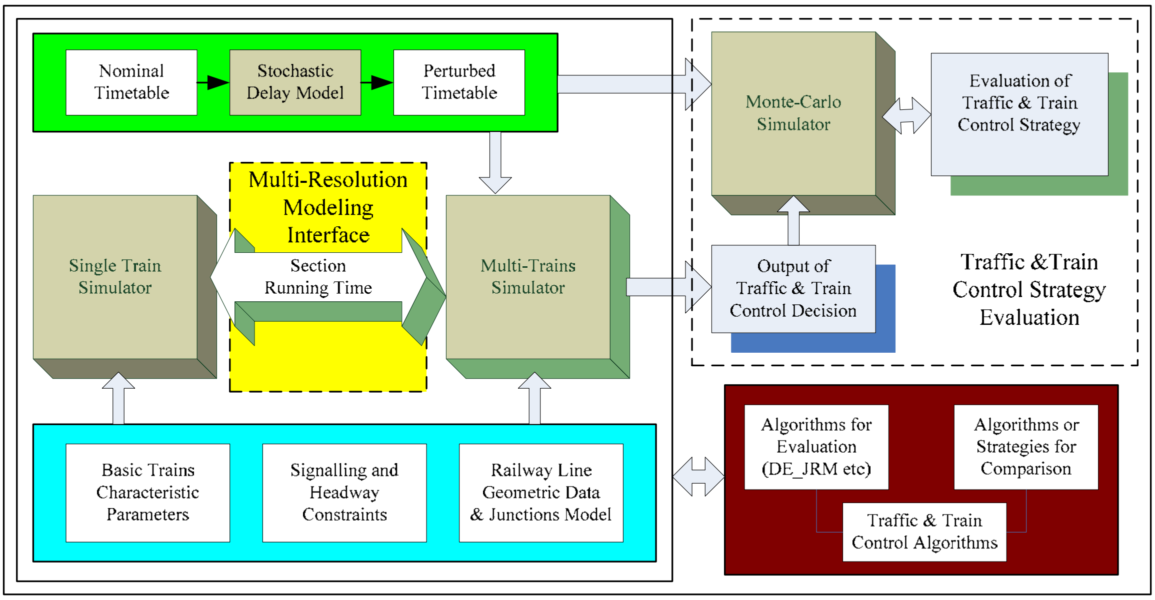

Figure 4-6 Systematic Approach for Evaluation of Algorithms for JRM

The diagram in Figure 4-6 is used to provide an overview of the proposed systematic approach. The different elements of the system that together provide the required functionality are represented here. There are 5 main parts to the system architecture, namely:

(1) Basic infrastructure and rolling stock data, including line geometry, line speed limits, train mass, maximum power, static friction coefficient, parameters for the train resistance equation, service braking deceleration rate etc;

(2) Timetable repository, holding both nominal timetables and perturbed timetables, with the latter based on a stochastic delay model;

(3) Single train simulator and a multi-train simulator with a interface based on Multi-Resolution Modelling (MRM) method;

(4) Monte-Carlo Simulator, used for driving statistical evaluation based on the Monte-Carlo methodology;

(5) Algorithm repository, storing traffic and train control strategies and algorithms for JRM including DE_JRM and other algorithms or strategies for comparison.

All of the modules were established as M-files in a Matlab environment. The data flow between the modules is shown in Figure 4-6.

The main function of the single train simulator is to carry out running time calculations for the trains in junction areas. On the basis of the data provided by the infrastructure and rolling stock data module, the train running times in each section of the junction areas can be calculated. These running times include minimum running times and operational running times which are used in a multi-train simulator. The interface between the single train simulator and the multi-train simulator transmits the section running times. A Multi-Resolution Modelling concept (Davis and Bigelow 1998) is applied between the single train simulator and multi-train simulator, the macroscopic simulation is applied in the multi-train simulator based on the headways of trains in junction areas.

Perturbed scenarios are generated from the nominal timetable using a stochastic delay model which can be created from real railway operations data and sophisticated empirical models. These are then used in a Monte-Carlo simulation for the statistical evaluation of the performance of the rescheduling algorithms and strategies.

The performance evaluation of the algorithm DE_JRM is presented with a typical case study using the systematic approach described in this section.

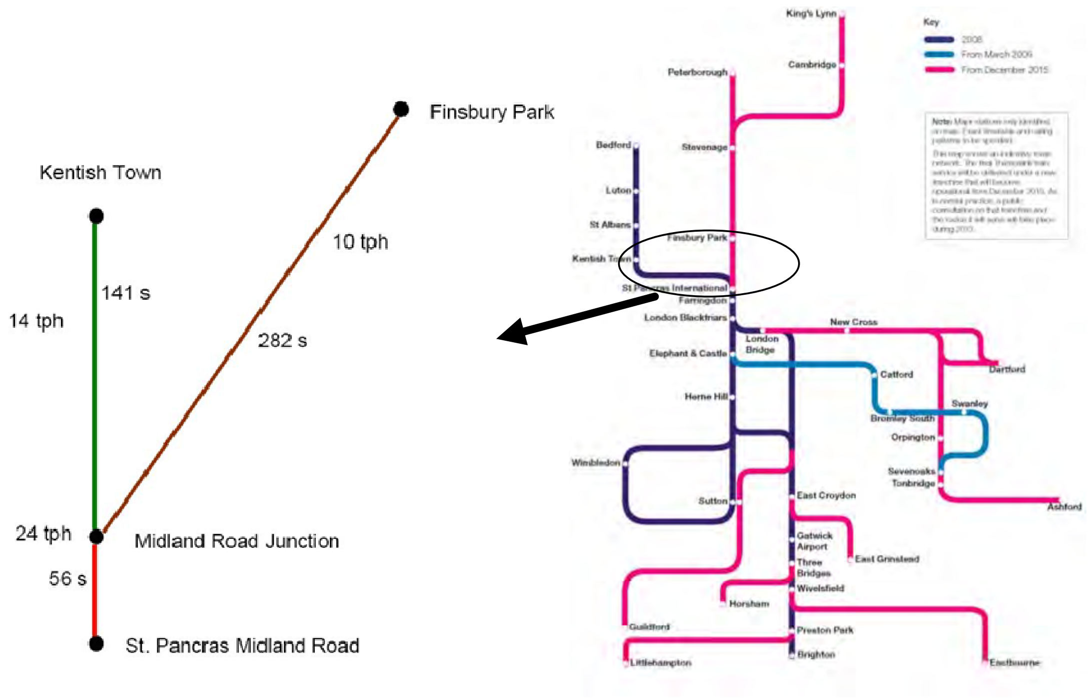

Figure 4-7 Layout of Thameslink Scenario for Case Study

4.2.2 Case Study

A sketch map of the layout for the case study is shown in Figure 4-7. It shows a simplified version of part of the track arrangements to be created by the Thameslink infrastructure investment programme. Some of the tracks shown are in tunnel and relatively steeply graded, with fairly tight curves. The Midland Road Junction will be constructed into a fly-over junction and only services travelling respectively from Kentish Town and Finsbury Park to St. Pancras International (the underground station previously known as St. Pancras Midland Road) are discussed in this thesis. These are approaching a bottleneck section, having previously travelled on two different parts of an intensively used suburban network, from as far as Bedford, Cambridge and Kings Lynn. Services travelling north from St. Pancras Midland Road are not of interest in this chapter because they are leaving the bottleneck section without conflicts with the trains travelling south in Midland Road Junction, which is a fly-over junction.

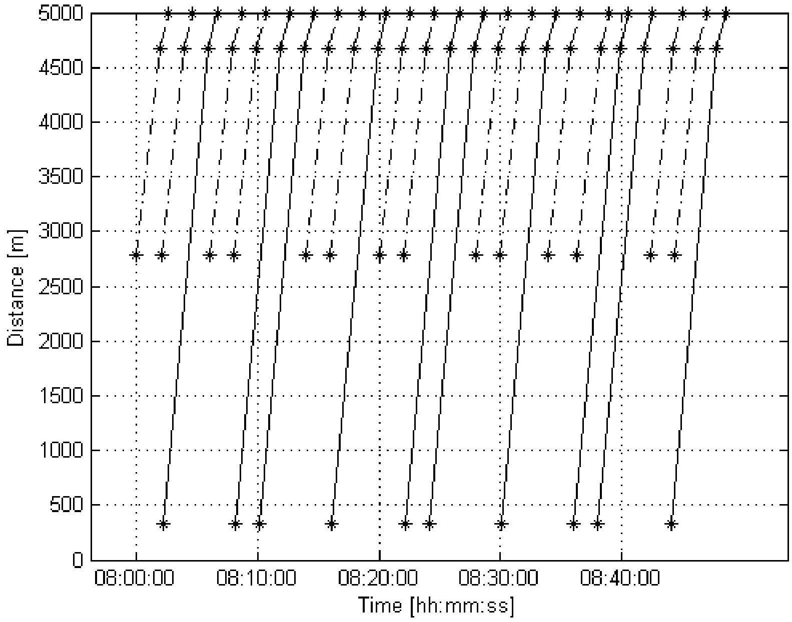

Figure 4-8 Nominal Timetable used for the Simulation, in Time-Distance Format

According to the Thameslink Programme (Thameslink-Programme), there will be an objective of the Key Output 2 (KO2), planned to be achieved in 2018 in the Thameslink project. In KO2, it is planned that a service of 24 trains per hour will run in each direction going through the bottleneck section of the Thameslink line during peak hours. In terms of the perspective of the Thameslink project, it is assumed that there are 14 trains per hour (one timetable period) from Kentish Town to St. Pancras Midland Road and 10 trains per hour from Finsbury Park to St. Pancras Midland Road in the peak time, as required in the Thameslink programme plan. The times shown in Figure 4-7 are the running times for trains on each section. The nominal timetable used in this scenario is shown in Figure 4-8 as a time-distance graph, produced on the basis of section running times. The dash-dot lines denote trains from Kentish Town to St. Pancras Midland Road. The solid lines denote trains from Finsbury Park to St. Pancras Midland Road. In the nominal timetable, the train service interval over the section from Midland Road Junction to St. Pancras Midland Road is 150 s. All trains continue from St. Pancras Midland Road to Blackfriars, thus making this section a bottleneck with a metro-type service.

Class Model	Class 377/1
Power Collection	750V DC/ 25kV AC
Vehicle Formulation	DMS(A)+ PTS+MOS+DMS(B)
Total Length	80.78m (DMS(A), DMS(B) - 20.4m, PTS, MOS - 19.99m)
Total Weight	173.6 tonnes DMS(A) - 46.2 tonnes PTS - 40.7 tonnes MOS - 40.5 tonnes DMS(B) - 46.2 tonnes
Max Speed	100mph (161km/h)
Traction Output	1,500 KW

Table 4-1 Configuration of Class 377 Trainsets

Figure 4-9 Class 377 Running on Thameslink Line (Wikimedia Commons, 2011)

In this thesis, Class 377 train-set is chosen as the rolling stock for simulation, which is a typical suburban EMU running on existing Thameslink line. The basic characteristics and image of Class 377 train-sets are shown in Table 4-1 and Figure 4-9. All of the trains will be formed of 3*4-car Class 377 sets, at 24 trains per hour in each direction through the bottleneck section.

According to the rescheduling decision generated by DE_JRM, some of the approaching trains, which have suffered a delay earlier on in the journey, need to speed up to arrive at the junction point (4677.76 m) closer to the timetabled time, by using the recovery time. This is assumed to be 20% of the operational running time in this case, that is, when running at 80% of line speed. Conversely, some of the trains need to slow down to avoid potential conflicts at the junction point.

According to the JRM, if an hour's service was to be considered in this Thameslink scenario, the number of possible route setting decisions will be $24!/(10! * 14!) = 1961256$. In the common computing environment used by the author (CPU: Intel Pentium Dual CPU 2.4 G Hz, 2 G of RAM, Computing software: Matlab R2008a), for every route setting decision, it takes on average about 0.3 s using the solver provided by Matlab to solve the lower level Constrained Nonlinear Programming problem of finding the optimal train arrival times $t_{i_{r,e}}^{p^*}$ at junction point. That means if we use this computer programme to enumerate all the possible route setting decisions, and then solve the lower level Constrained Nonlinear Programming problem, it will take about $1961256 * 0.3/3600 = 163.438$ hours to get the optimal solution, which is fully unacceptable for practical real time junction rescheduling applications.

With the presented Thameslink scenario as a case study, the performance of DE_JRM is evaluated in terms of goodness of output solution and computational time. For the evaluation of the algorithm, two kinds of evaluation work was undertaken to evaluate the algorithm performance of DE_JRM using the systematic simulation approach; evaluation with typical delay scenarios and statistical evaluation based on Monte-Carlo methodology.

4.2.3 Evaluation with Typical Delay Scenarios

To evaluate the performance of DE_JRM with typical delay scenarios, 26 typical delay scenarios within the case study were chosen, including single train delays and multi-train delay events. These delays are introduced at the boundaries of the junction areas, which are Kentish Town station and Finsbury Park station, shown in Figure 4-7. For each delay scenario, the DE_JRM algorithm is applied to generate a train rescheduling decision for the junction area shown in Figure 4-7 and the weighted average delay (WAD) of the trains passing through junction area is calculated, meanwhile the WAD derived by the First Come First Served (FCFS) strategy, which is a common junction control strategy widely used in British's railways, is also calculated. The WAD of the trains through the junction area, rescheduled with a control strategy implemented in Automatic Route Setting (ARS) systems in Britain's railways, is calculated. The principle of the ARS strategy is described in Appendix A. The results of WAD derived from DE_JRM, FCFS and ARS strategy are compared.

Scenario ID	Delay Type	Number of Delayed Trains	Delay Time (second)	Delayed Trains
1	Single Train Delay	1	60	2/K
2	Single Train Delay	1	60	5/K
3	Single Train Delay	1	60	8/K
4	Single Train Delay	1	300	2/K
5	Single Train Delay	1	300	5/K
6	Single Train Delay	1	300	8/K
7	Single Train Delay	1	60	2/F
8	Single Train Delay	1	60	3/F
9	Single Train Delay	1	60	4/F
10	Single Train Delay	1	300	2/F
11	Single Train Delay	1	300	3/F
12	Single Train Delay	1	300	4/F
13	Multi-train Delay	2	60	2/K, 2/F
14	Multi-train Delay	2	60	5/K, 3/F
15	Multi-train Delay	2	60	5/K, 4/F
16	Multi-train Delay	2	60	2/K, 4/F
17	Multi-train Delay	2	300	2/K, 2/F
18	Multi-train Delay	2	300	5/K, 3/F
19	Multi-train Delay	2	300	5/K, 4/F
20	Multi-train Delay	2	300	2/K, 4/F
21	Multi-train Delay	6	60	2/K, 8/K, 11/K, 1/F, 3/F, 8/F

22	Multi-train Delay	6	60	3/K, 5/K, 10/K, 3/F, 4/F, 7/F
23	Multi-train Delay	6	300	2/K, 8/K, 11/K, 1/F, 3/F, 8/F
24	Multi-train Delay	6	300	3/K, 5/K, 10/K, 3/F, 4/F, 7/F
25	Multi-train Delay	10	60	2/K, 5/K, 8/K, 11/K, 13/K, 1/F, 3/F, 6/F, 7/F, 9/F
26	Multi-train Delay	10	300	2/K, 5/K, 8/K, 11/K, 13/K, 1/F, 3/F, 6/F, 7/F, 9/F

Table 4-2 Delay Scenarios used for Evaluation

Table 4-2 shows the delay scenarios which were used for the evaluation. In the simulation, one period of a repeating timetable was chosen as the time window, (shown in Figure 4-8) commonly one hour for British railways. In the column “Delayed Trains”, “2/K” and “2/F” denote the “2nd train from Kentish Town station in a time window” and “2nd train from Finsbury Park station in a time window” respectively, the delayed trains are denoted in format “Number/Station” in Table 4-2.

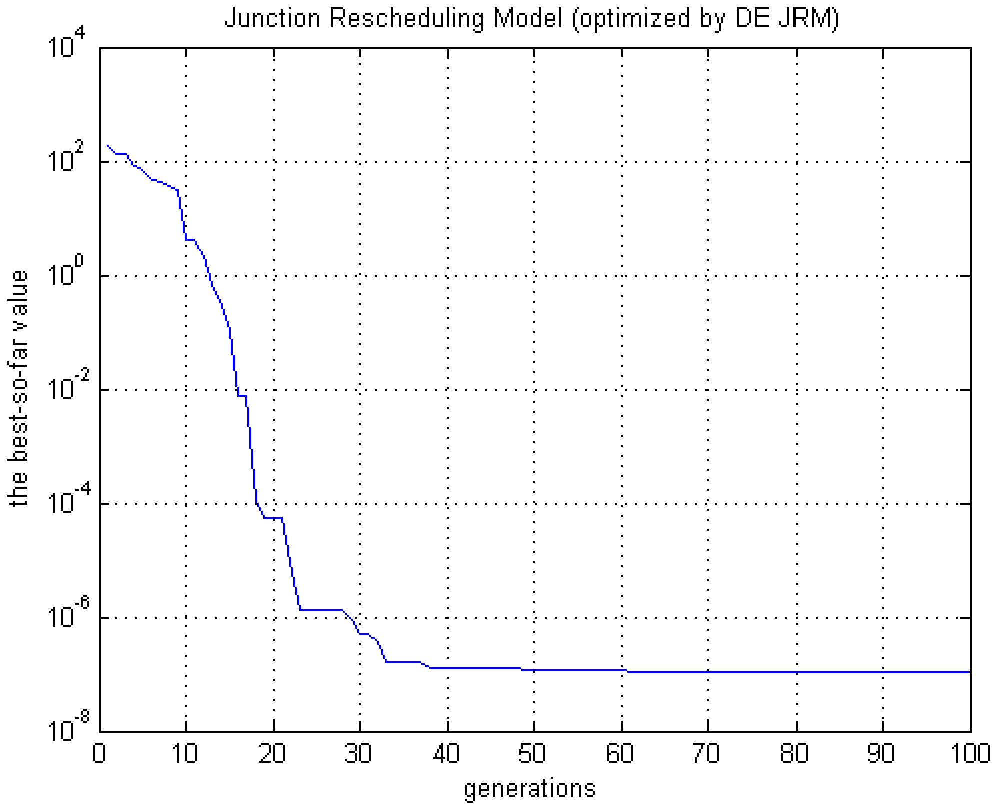

Figure 4-10 Convergence Graph of DE_JRM

The basic parameters for algorithm DE_JRM and simulation scenarios are listed as follows:

- ❖ The ratio of weights assigned for the trains from Kentish Town station and the trains from Finsbury Park station is set to 7:5.
- ❖ Number of population in DE_JRM is set to 200.
- ❖ Mutation factors are set to $F_1 = F_2 = 0.9$.
- ❖ Crossover factor CR is set to 0.95.

Scenario ID	DE_JRM		FCFS		ARS	
	WAD (s)	Time (s)	WAD (s)	Time (s)	WAD (s)	Time (s)
1	1.0524	2.6827	3.0922	N/A	3.0922	N/A
2	1.1269	2.8262	3.2236	N/A	3.2236	N/A
3	1.0889	2.8307	3.0922	N/A	3.0922	N/A
4	8.2644	2.6617	20.7236	N/A	20.7236	N/A
5	7.9984	2.3507	24.0842	N/A	22.8874	N/A
6	8.0457	2.9664	20.7236	N/A	20.7236	N/A
7	0.9864	2.6651	2.3026	N/A	2.3026	N/A
8	1.0227	2.9004	2.4342	N/A	2.4342	N/A
9	1.0778	2.5521	2.4342	N/A	2.4342	N/A
10	8.3684	2.8820	19.0780	N/A	18.3715	N/A
11	8.9632	3.0018	18.4210	N/A	18.4210	N/A
12	7.9647	2.7554	16.1809	N/A	16.1809	N/A
13	1.5689	3.0267	5.3847	N/A	5.3847	N/A
14	1.6247	2.6510	5.1894	N/A	5.1894	N/A
15	1.6022	2.9448	5.6578	N/A	5.6578	N/A
16	1.6224	2.8551	5.5264	N/A	5.5264	N/A
17	10.2325	2.6779	40.0448	N/A	40.0448	N/A
18	20.6877	2.7502	41.0442	N/A	38.2358	N/A
19	11.1654	2.9310	46.0526	N/A	42.2611	N/A
20	9.9885	2.4788	36.8422	N/A	36.8422	N/A
21	2.0221	3.0558	16.2511	N/A	16.2511	N/A
22	2.3815	3.2411	16.5032	N/A	16.5032	N/A
23	74.3365	2.8997	155.5832	N/A	140.6642	N/A
24	76.3365	3.1284	149.6713	N/A	138.0969	N/A
25	2.6618	2.8667	27.2367	N/A	27.2367	N/A
26	102.3683	3.1504	233.4526	N/A	210.3657	N/A

Table 4-3 WAD of DE_JRM for 26 Typical Delay Scenarios

Table 4-3 shows the Weighted Average Delay (WAD) values of the listed 26 delay scenarios after rescheduling with DE_JRM, FCFS and ARS strategy.

The computational time of DE_JRM is also listed in the table, and one of the typical convergence graphs is shown in Figure 4-10. The calculation of computational time of DE_JRM is undertaken in a common computing environment used by the author (CPU: Intel Pentium Dual CPU 2.4 G Hz, 2 G of RAM, Computing software: Matlab R2008a). The computational time of FCFS and rescheduling strategy of ARS can be ignored because these two strategies directly generate decisions for train rescheduling in junction areas according to the trains' movement without searching processes for optimisation. As seen from Table 4-3, in terms of computational time, DE_JRM can satisfy the time restriction for real time train rescheduling applications in the junction area, and there is not much difference in computational time between single train delay scenarios and multi-train delay scenarios, as well as short delay scenarios and long delay scenarios.

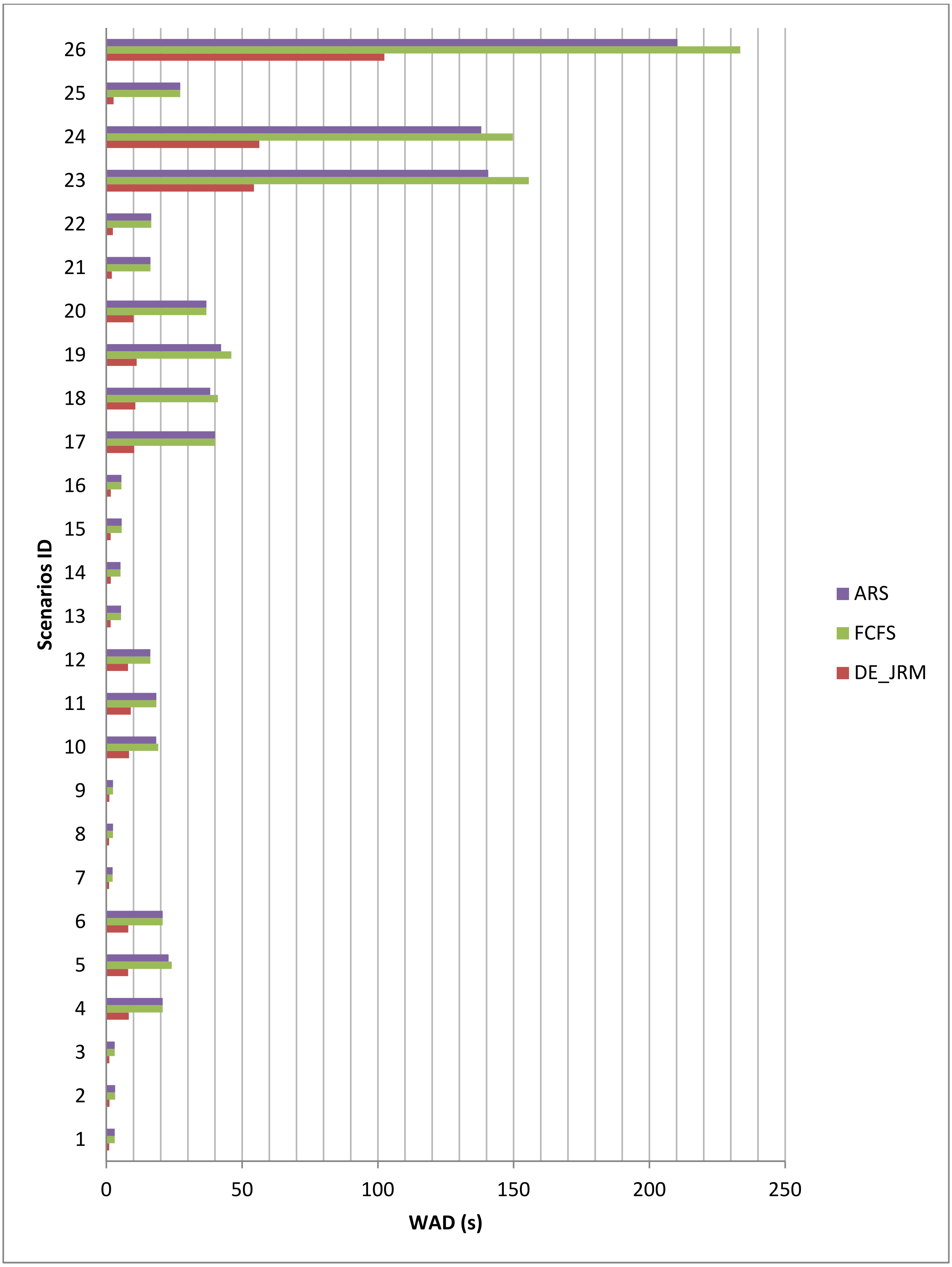

Figure 4-11 Comparison of WAD with DE_JRM, FCFS and ARS

Figure 4-11 shows the comparison of WAD with DE_JRM, FCFS and the ARS strategy. As can be seen, for both short delay scenarios and long delay scenarios, algorithm DE_JRM decreases WAD significantly compared with FCFS and the ARS strategy in terms of the definition of WAD in this thesis. In most of the short delay scenarios, WAD after rescheduling with DE_JRM is quite low and close to 0, which means that the perturbed trains can be rescheduled quite close to the nominal timetable due to the train re-timings in junction area that allow trains to be able to speed up when approaching to junctions. DE_JRM also shows good performance in long delay scenarios. Compared with short delay scenarios, it is not possible to decrease the WAD very much and reschedule the perturbed trains quite close to the nominal timetable because of the train control and operation constraints like headway constraints, train speed restriction etc.

In addition, FCFS and the ARS strategy show similar performance in both short and long delay scenarios. The ARS strategy cannot improve WAD in this case study, because the weights assigned to the trains on the Thamelink line have no significant difference compared with highly mixed railway lines with different classes of trains, such as passenger-freight mixed railway lines. Another reason is that the density of bottleneck sections of the Thameslink line is very high, The ARS strategy also only considers the one most front train approaching the junction point on each route rather than searching the optimal solutions for all the trains in a time window (set as one hour in case study) on each approaching route.

Because of the complexity of railway operation, the selected 26 delay scenarios are far less than enough to cover all the typical delay scenarios for evaluation. In order to further evaluate the performance of the algorithm DE_JRM, a statistical evaluation method based on Monte-Carlo methodology is presented in the next section, to evaluate the performance of the algorithm DE_JRM, in terms of a Statistical WAD (SWAD).

4.2.4 Statistical Evaluation with Monte-Carlo Methodology

Because of the complexity of railway systems and operations, it is very difficult to evaluate whether an algorithm can handle all the scenarios that occur in railway operations. It is not possible to configure and simulate all the scenarios of railway operations, even only for a relatively small area of railway lines.

In this section, a statistical evaluation method based on Monte-Carlo simulation methodology (Raeside 1974; Milchev 2003; Raychaudhuri 2008) is introduced to evaluate the performance of the algorithm DE_JRM, in terms of a Statistical WAD (SWAD). The basic procedure flow chart of the proposed statistical evaluation method is shown in Figure 4-12.

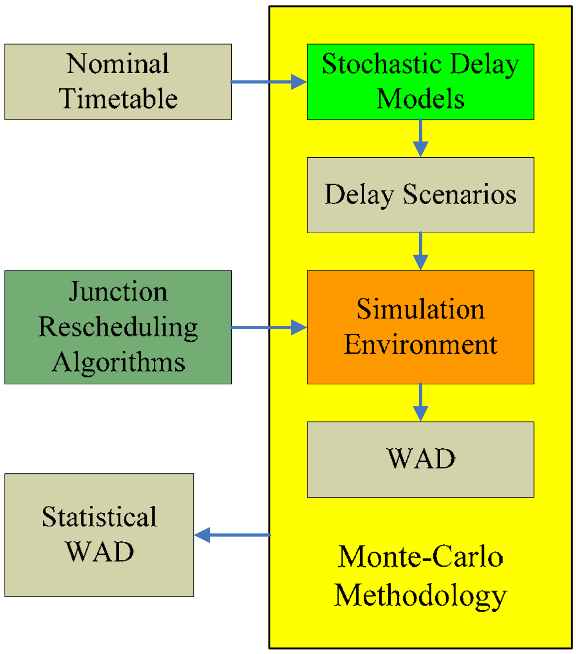

Figure 4-12 Procedure of the Proposed Statistical Evaluation

According to Monte-Carlo simulation methodology, one of the important procedures is to generate large numbers of stochastic delay scenarios derived from certain types of stochastic delay model. The stochastic model includes the delay probability distribution functions which can be fitted in terms of empirical train operation data or from the classic delay models like normal distribution, negative exponential distribution etc. In the scenarios where the railway lines have been in operation, it is possible to collect the real data from the railway operation fields with the existing data acquisition systems used by infrastructure management companies. Nevertheless, for the scenarios where the railway lines have not been in operation yet, there is no empirical data available, thus some classic delay probability distribution functions can be used to generate the delay scenarios stochastically. For every delay scenario, the junction rescheduling algorithms and strategies are applied with the support of the simulation environment shown in Figure 4-6, and the corresponding WAD is calculated. For evaluation of every junction rescheduling algorithm, with the large numbers of stochastic delay scenarios, the statistical average value of the individual WADs is calculated and named as Statistical WAD (SWAD), which is regarded as the performance indicator in this thesis for the statistical evaluation of train rescheduling algorithms. SWAD of each junction rescheduling algorithms is compared. The lower SWAD value shows better performance of the rescheduling algorithm in terms of goodness of the optimised solutions.

4.2.4.1 Stochastic Delay Model

The main purpose of performance evaluation of train rescheduling algorithms in this thesis is to check whether the algorithms can handle most of the delay scenarios in the event of disturbance. Large numbers of delay scenarios need to be generated for the statistical evaluation based on Monte-Carlo simulation methodology. The generation of the delay scenarios is regarded as a key issue in the process of statistical evaluation.

Many different train delay models have been studied (Higgins and Kozan 1998; Hansen 2004). One of the commonly used delay models for positive arrival delays is exponential distribution (Schwanhauber 1974; Yuan, Goverde et al. 2002). The exponential distribution can also be applied to model departure delays and original delays (Ferreira and Higgins 1996). To model the variability of train delay better, some other more flexible distribution models have also been applied, for example, the normal distribution, the gamma, Weibull and lognormal distributions etc (Carey and Kwiecinski 1994; Yuan, Goverde et al. 2006).

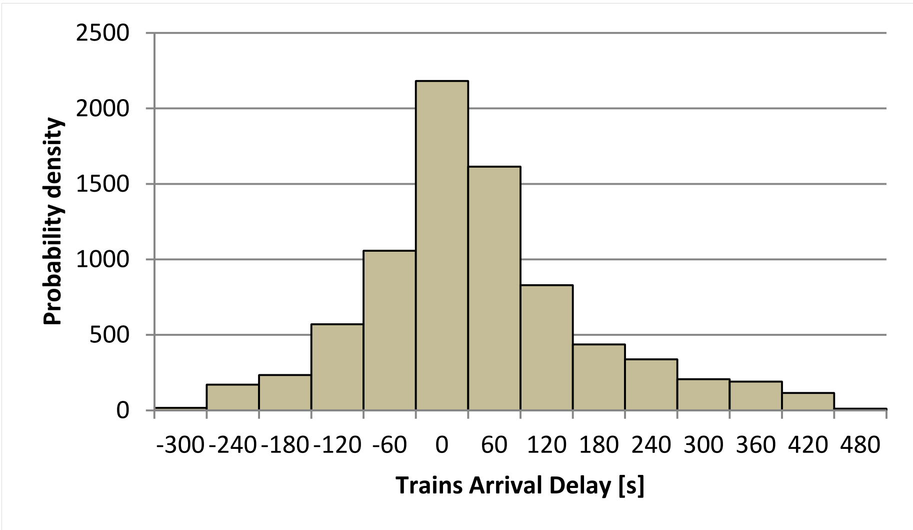

Figure 4-13 Train Arrival Delays Density at the Southbound Platform Track of the existing Kentish Town Station (see Table 4-4)

As for the case study of the Thameslink scenario in this thesis, currently, there is no operational data for the configuration of the Thameslink route under investigation since construction of the Midland Road junction has not been completed and the operational timetable of 24 trains per hour through the core area is expected to apply on the

Thameslink route from 2018. Therefore, data from the existing Thameslink route, recorded by Network Rail over a period of one month, has been analysed. In the current timetable, some slow trains dwell at the boundary stations while other trains do not. For the former, the arrival time is an actual arrival time while, for the non-stopping trains, the passing time is taken as the ‘arrival’ time. A probability density distribution histogram for the train arrival delays at the southbound platform track of the existing Kentish Town Station on the Thameslink route is shown in Figure 4-13. In this thesis, some very abnormal incidents which cause very long train delays are not considered. The data can only be regarded as a reference train delays data for the statistical evaluation of the rescheduling algorithm DE_JRM.

It is very difficult to forecast which train delay probability distribution will be closest to reality before KO2 has been implemented on the Thameslink route. Four stochastic delay probability distributions for boundary arrival time have been assumed in this thesis to generate large numbers of delay scenarios for the purpose of statistical evaluation of DE_JRM. They are:

(1) Empirical distribution over [-300, 480] based on existing operational train delay data;
(2) Normal distribution over [-30, 120] for short train delays;
(3) Normal distribution over [-60, 300] for long train delays;
(4) Negative exponential distribution over [0, 480].

These four train delay distributions were used for generating large numbers of delay scenarios in the simulation environment, and with the presented systematic approach, large numbers of computer simulation experiments were undertaken for the statistical evaluation of rescheduling algorithms based on the Monte-Carlo simulation methodology.

The details of the stochastic delay probability distributions are as follows.

1) *Empirical distribution over [-300, 480] based on existing operational train delay data*

In operational railway networks, the daily train delays data at specific locations is usually recorded by field management systems for purposes such as operational analysis, incident recording and other short term or long term strategy regulations. With large daily operational train delays data, the train delays probability distribution curves can be estimated and fitted empirically. A common method used for fitting the train delay probability curves is introduced with the presented case study, as well as the computer generation method of large numbers of delay scenarios.

With the operational train delay data at the southbound platform track of the existing Kentish Town Station on Thameslink, the frequency of the delays in certain delay intervals is listed in Table 4-4. The negative train delays in the table mean the trains arrive at the recorded location earlier than the nominal timetable. Some of the very early and very late delays have not been counted in the table. Sixty seconds was chosen as the interval for the bins of delay time.

Bin	Delay Time (s)	Frequency
-300	DT<=-300	16
-240	-300<DT<=-240	170
-180	-240<DT<=-180	234
-120	-180<DT<=-120	570
-60	-120<DT<=-60	1057
0	-60<DT<=0	2182
60	0<DT<=60	1614
120	60<DT<=120	829
180	120<DT<=180	437
240	180<DT<=240	338
300	240<DT<=300	206
360	300<DT<=360	190
420	360<DT<=420	115
480	420<DT<=480	11
	480<DT	0
Total		7969

Table 4-4 Recorded Train Delays Frequency in Different Intervals

Here, a definition of Empirical Distribution Function is introduced. Let $m_n(x)$ denote the number of elements of the sample that are smaller than x, which is a given real number and n is the total number of the sample.

Definition: *The function* $F_n(x) = m_n(x)/n$ *is called the empirical distribution function of the sample* $\{x_1, x_2, \cdots, x_n\}$.

Glivenko's Theorem:

Let $F(x)$ is the true distribution function, we put

$D_n = \sup_{x \in \mathbb{R}^1} |F_n(x) - F(x)|$ **Equation 4-16**

Then

$P(lim_{n \to \infty} D_n = 0) = 1$ **Equation 4-17**

As seen from *Glivenko's Theorem*, the empirical distribution function $F_n(x)$ can be regarded as an approximation of the true distribution as long as the total number of sample n is big enough for analysis purposes. In practical applications, we normally count the number of samples in terms of several equal intervals, as Table 4-4 shows.

With the train delay frequency data shown in Table 4-4, the discrete values of $F_n(x)$ can be calculated as shown in Table 4-5. The fitted curve of the empirical distribution function of the sample train delay data from the existing Kentish Town station is drawn in Figure 4-14. A linear interpolating method was applied to fit the curve between two successive discrete value points.

x	$m_n(x)$	$\mathrm{m_n(x)/n}$
-300	16	0.002008
-240	186	0.02334
-180	420	0.052704
-120	990	0.124231
-60	2047	0.25687
0	4229	0.530681
60	5843	0.733216
120	6672	0.837244
180	7109	0.892082
240	7447	0.934496
300	7653	0.960346
360	7843	0.984189
420	7958	0.99862
480	7969	1

Table 4-5 Discrete Values of Empirical Distribution Function of Train Delays Data

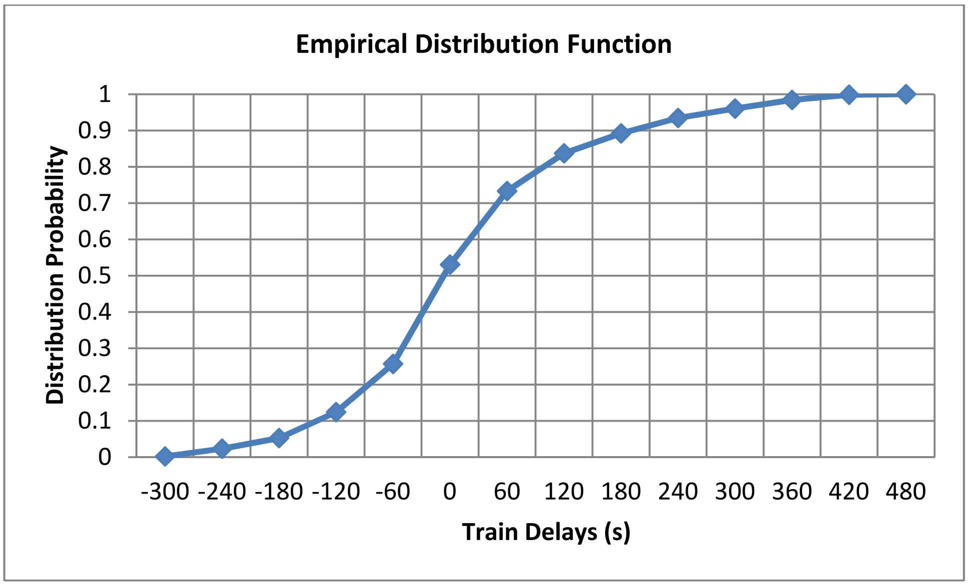

Figure 4-14 Empirical Distribution Function of Train Delay Data

Based on the empirical distribution curve shown in Figure 4-14, large numbers of delay scenarios can be generated in simulation environments for statistical evaluation of the rescheduling algorithms. We can generate large numbers of random numbers between [0, 1] with uniform distribution. For every generated random number, in terms of the empirical distribution curve shown in Figure 4-14, we can find the corresponding train delay number on x-axis as shown in Figure 4-15. The generated train delays were added on the train boundary arrival time to simulate the delay scenarios for statistical evaluation. Meanwhile, headways of trains at boundaries also need to be kept.

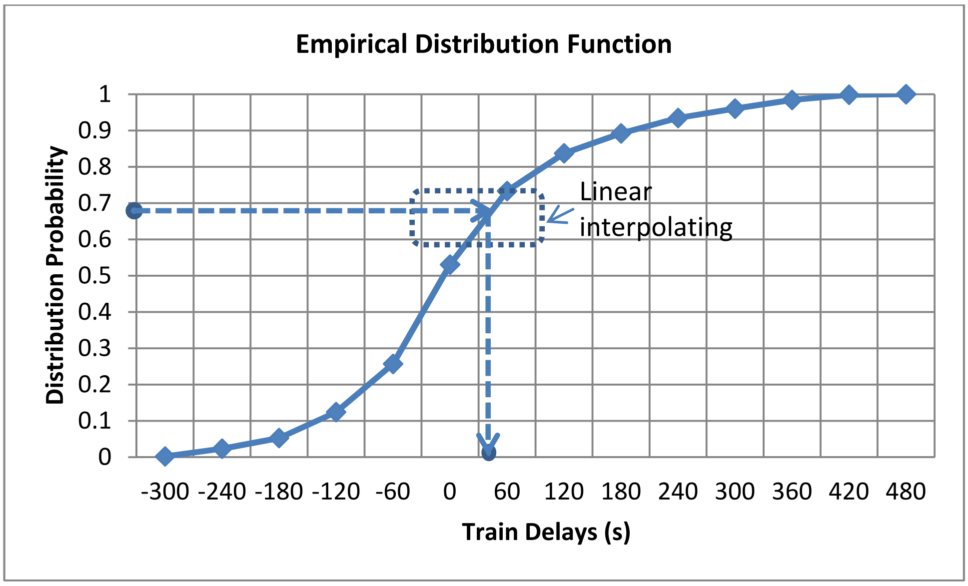

Figure 4-15 Generation of Train Delay with Random Probability Numbers

2) Normal distribution over [-30, 120] for short train delays

In this thesis, a train delay distribution based on normal distribution is also assumed to apply for simulation of short train delay scenarios over delay time [-30, 120]. The purpose of this assumption is to evaluate the performance of the rescheduling algorithms under short train delay scenarios. The basic probability density function of normal distributions is introduced as follows.

Definition: The probability density function of normal distribution is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \qquad \textbf{Equation 4-18}$$

where parameter μ is called the *mean* which indicates the peak location of probability density function of normal distributions and σ^2 is the *variance* which is regarded as the measure of the width of the distribution.

A normal distribution over [-30, 120] is assumed to generate short train delay scenarios in this thesis, where the parameter μ is set to 45 and σ is set to 37.5. The probability density function diagram and the cumulative probability density function diagram of the assumed normal distribution are shown in Figure 4-16 and Figure 4-17.

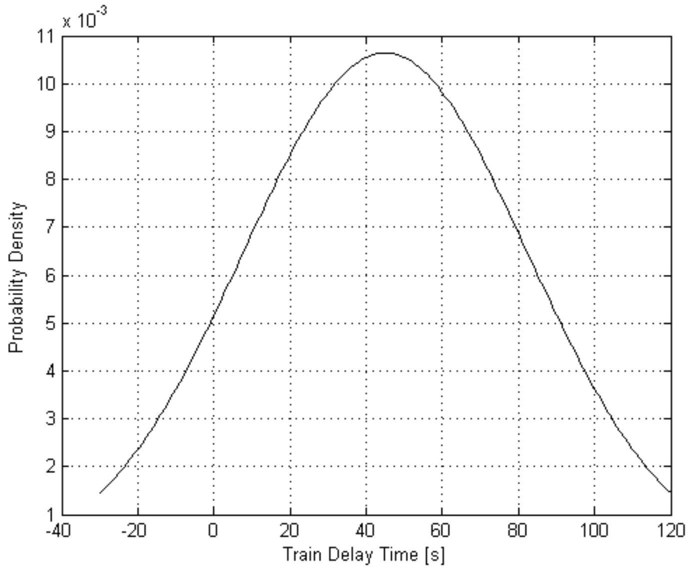

Figure 4-16 Probability Density Function of the Normal Distribution for Short Train Delays

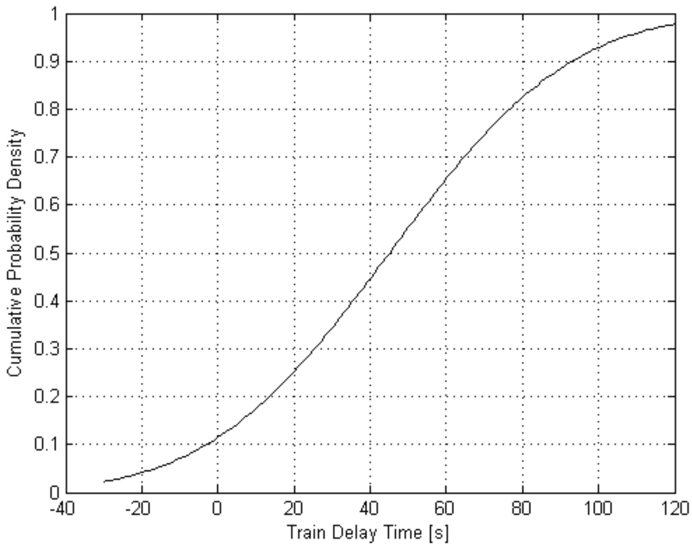

Figure 4-17 Cumulative Probability Density Function of the Normal Distribution for Short Train Delays

According to the assumed normal distribution over [-30, 120], large numbers of short train delays can be generated in the simulation environment as the short train delay scenarios. The value domain of stochastic numbers generated in Matlab in terms of the assumed normal distribution is [-Inf, +Inf]. With the assumed normal distribution over [-30, 120], there will be about 95% in probability to generate the data which are inside of the interval [-30, 120]. The generated train delay data which are out of the interval [-30, 120] will be abandoned, only the train delay data inside of the interval [-30, 120] are used for simulation of short train delay scenarios.

3) Normal distribution over [-60, 300] for long train delays;

For statistical evaluation of rescheduling algorithms under long train delays, a normal distribution over [-60, 300] is assumed to generate large numbers of long train delay scenarios. The parameter μ of the applied normal distribution is set to 120 and σ is set to 90. The probability density function diagram and the cumulative probability density function diagram of the assumed normal distribution for long train delays are shown in Figure 4-18 and Figure 4-19. The generation procedures of long train delay scenarios is the same with the generation procedures of short train delay scenarios.

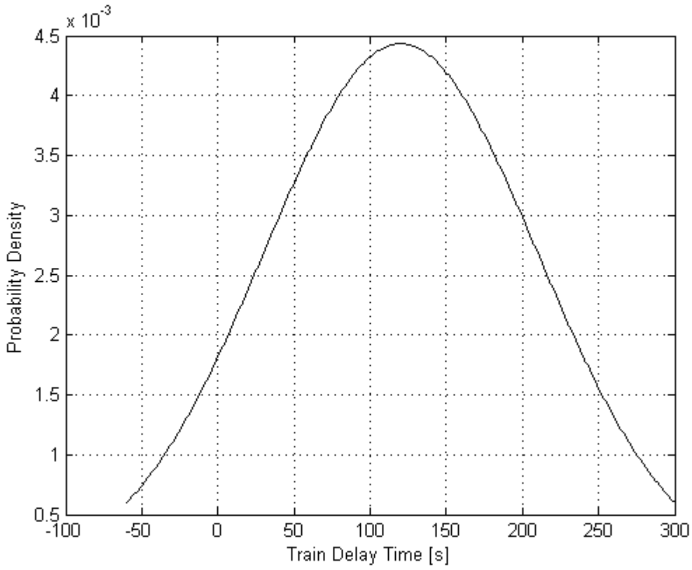

Figure 4-18 Probability Density Function of the Normal Distribution for Long Train Delays

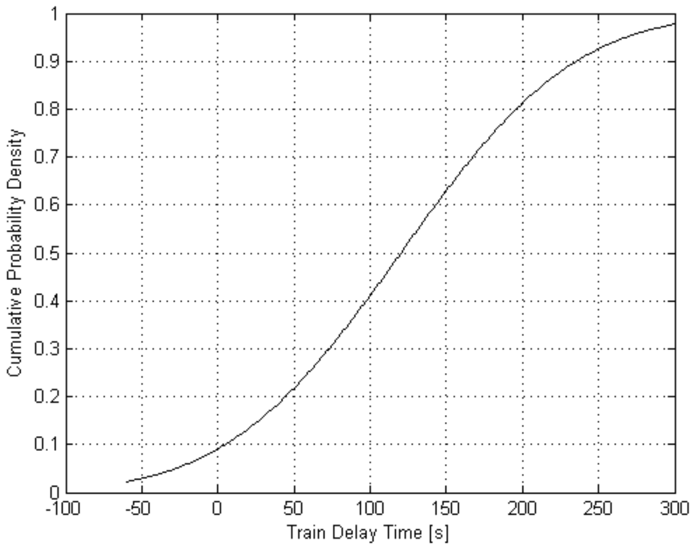

Figure 4-19 Cumulative Probability Density Function of the Normal Distribution for Long Train Delays

4) Negative exponential distribution over [0, 480].

The exponential distribution is often considered to be a valid model for train delays (Schwanhauber 1974; Ferreira and Higgins 1996; Yuan, Goverde et al. 2002; Goverde 2005). Besides the stochastic distributions listed above, a negative exponential distribution over [0, 480] is also assumed to generate the delay scenarios for statistical evaluation of rescheduling algorithms.

The general probability density function of an exponential distribution is shown in Equation 4-19.

$$f(x) = \begin{cases} \lambda * e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$ **Equation 4-19**

where $\lambda > 0$ is the parameter of exponential distributions. λ is set to 1/90 in this thesis to generate the train delays in the interval [0, 480]. The expected value of the generated train delays is $\frac{1}{\lambda} = 90$ s. The probability density function and cumulative distribution function are shown in Figure 4-20 and Figure 4-21. The generation of train delay scenarios is the same with other delay models.

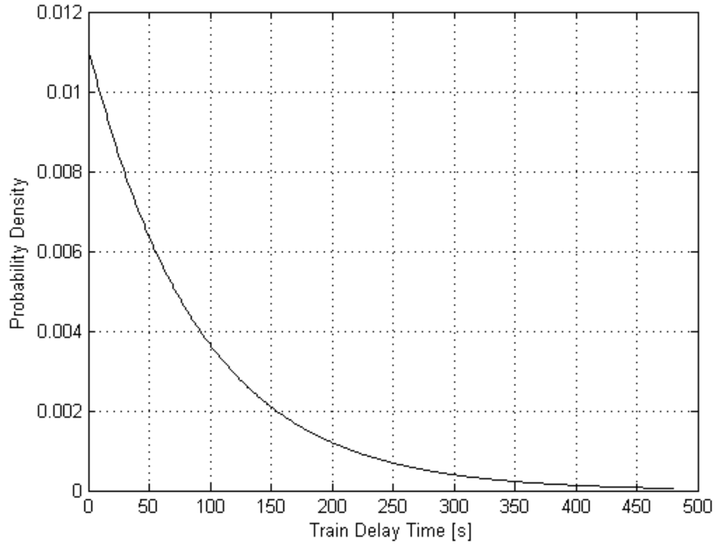

Figure 4-20 Probability Density Function of the Assumed Exponential Distribution

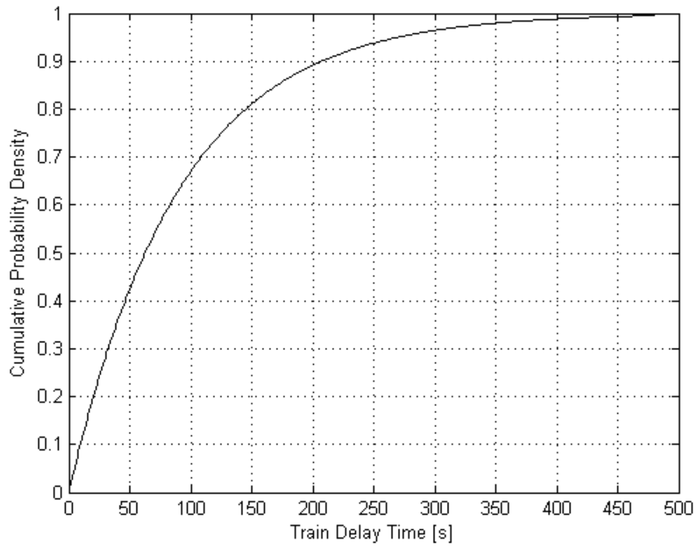

Figure 4-21 Cumulative Distribution Function of the Assumed Exponential Distribution

4.2.4.2 Simulation Experiments and Statistical Output Analysis

According to the presented four train delay probability distributions at control boundaries, large numbers of delay scenarios can be generated in the simulation environment. For each delay scenario, DE_JRM, FCFS and the ARS strategy have been applied in the simulation environment for rescheduling decision making respectively, and the WAD after rescheduling by each rescheduling algorithm were calculated. With large numbers of computer simulation experiments for evaluation, a Statistical WAD (SWAD) which is the statistical average value of the output WADs can be calculated for DE_JRM, FCFS and the ARS strategy. SWAD is regarded as the performance indicator of the rescheduling algorithms in this thesis. The lower SWAD of the rescheduling algorithms, means that the algorithm shows better performance on decreasing the WAD.

For each rescheduling algorithm, 10000 simulation experiments based on each train delay probability distribution have been undertaken for statistical evaluation. The SWAD of rescheduling algorithm DE_JRM, FCFS and the ARS strategy for each train delay distribution boundary is listed in Table 4-6.

Rescheduling Algorithm / Delay Distribution	DE_JRM	FCFS	ARS
Empirical distribution over [-300, 480]	48.6554	76.3221	74.3651
Normal distribution over [-30, 120]	25.3246	50.5617	50.1143
Normal distribution over [-60, 300]	110.4348	148.0294	145.3220
Negative exponential distribution over [0, 480]	67.1087	102.3629	100.1488

Table 4-6 SWAD of DE_JRM, FCFS and ARS under Different Delay Distributions for Flyover Junction Scenario

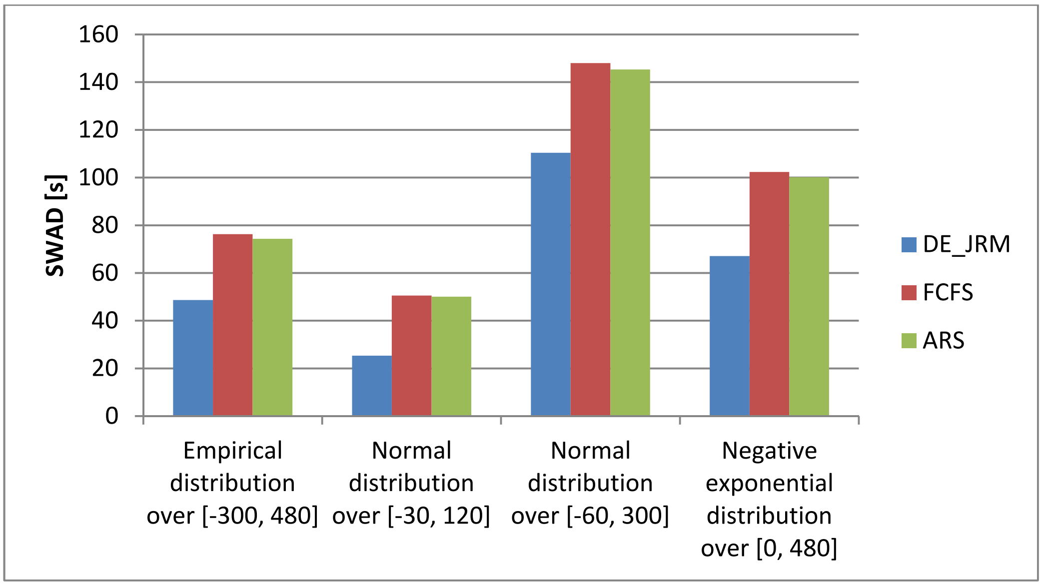

Figure 4-22 Comparison of SWAD with Rescheduling Strategies in Flyover Junction Scenario

Figure 4-22 shows the comparison of SWAD with rescheduling algorithm DE_JRM, FCFS and the ARS strategy under different delay probability distributions in flyover junction scenarios. It can be deduced that for all the four proposed train delay distributions, the WAD can be decreased significantly with the train rescheduling algorithm DE_JRM compared with FCFS or the ARS strategy. The statistical WAD with the ARS strategy cannot be decreased significantly compared with FCFS. It indicates that the application of the ARS strategy cannot bring many benefits to decreasing the WAD in this scenario.

Because of train operation and control constraints, even with DE_JRM, it is not possible to decrease the WAD to zero in these scenarios due to the limit recovery and margin times in the nominal timetable.

The average computation time of the algorithm DE_JRM is around 2-3 seconds, which is more than satisfactory for the real time applications of train rescheduling in junction areas.

4.2.5 Performance Evaluation of Rescheduling Algorithms for Flat Junctions

The case study presented in Chapter 4.2.2 refers to a train rescheduling problem of a fly-over junction where there are no conflicts between the trains approaching the junction from opposite directions. In this thesis, the performance of algorithm DE_JRM for solving real time train rescheduling problems in a flat junction is also studied. According to the case study described in Chapter 4.2.2, the Midland Road Junction which is expected to be implemented in 2015 is assumed to be a flat junction, shown in Figure 4-23 in the simulation environment for performance evaluation of rescheduling algorithms.

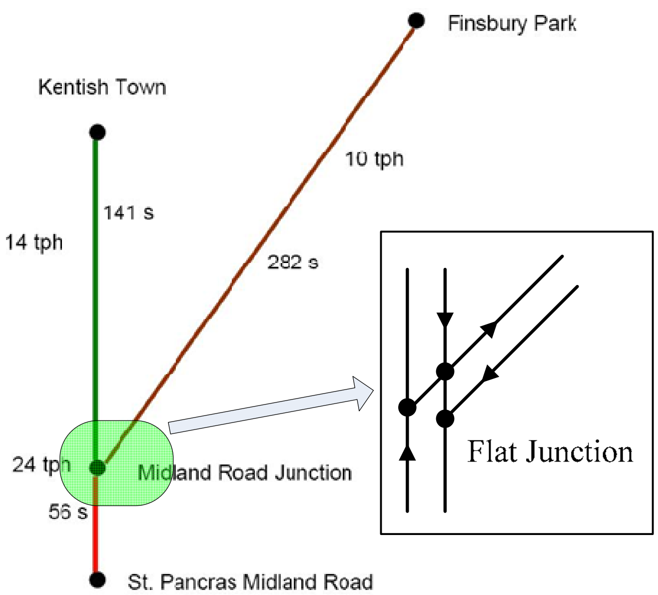

Figure 4-23 Layout of Assumed Flat Junction for Midland Road Junction

As shown in Figure 4-23, because of the flat junction, the trains approaching the Midland Road Junction from St. Pancras to Finsbury Park station would have potential conflicts with the trains from Kentish Town to St. Pancras. So the rescheduling decision making for the trains approaching Midland Road Junction also needs to consider the northbound trains from St. Pancras. For this scenario with a flat junction, large numbers of simulation experiments with delay scenarios are generated based on the presented delay distributions for the statistical evaluation of the rescheduling algorithms. The rescheduling algorithms DE_JRM, FCFS and the ARS strategy have been applied in the simulation environment for the generated large numbers of simulation experiments with delay scenarios. The SWAD of DE_JRM, FCFS and the ARS strategy were calculated after undertaking the simulation experiments.

In addition, for each rescheduling algorithm, 10,000 simulation experiments based on each train delay probability distribution at the TM boundary have been undertaken for statistical evaluation in this flat junction scenario. The SWAD of rescheduling algorithm DE_JRM, FCFS and ARS strategy for each train delay distribution is listed in Table 4-7.

Rescheduling Algorithm / Delay Distribution	DE_JRM	FCFS	ARS
Empirical distribution over [-300, 480]	59.4204	86.8941	86.7722
Normal distribution over [-30, 120]	30.4147	55.0482	54.7028
Normal distribution over [-60, 300]	141.4943	172.8642	166.7905
Negative exponential distribution over [0, 480]	87.7850	118.5137	117.4350

Table 4-7 SWAD of DE_JRM, FCFS and ARS under Different Delay Distributions in the Flat Junction Scenario

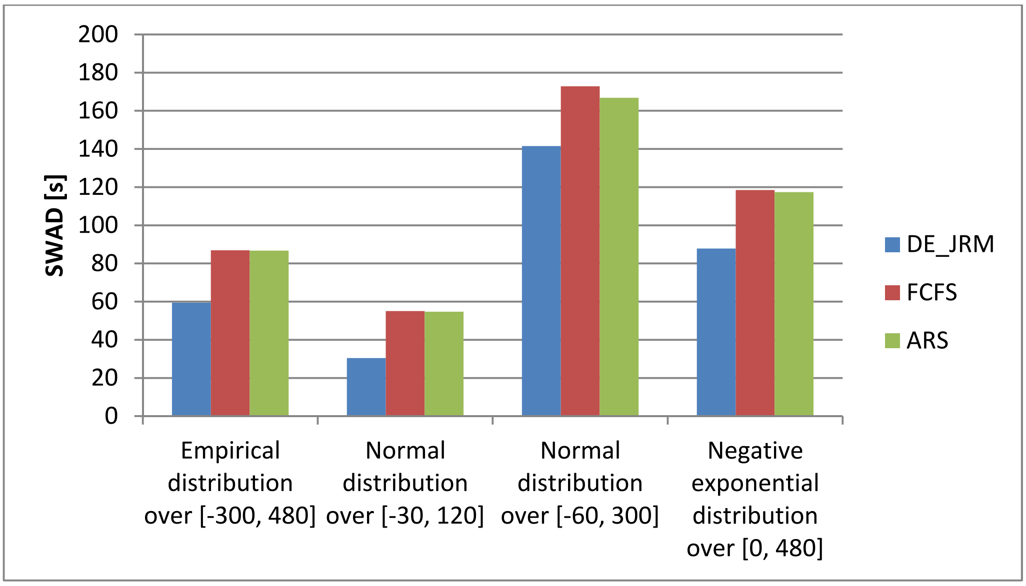

Figure 4-24 Comparison of SWAD with Rescheduling Strategies for Flat Junction Scenario

The comparison of SWAD of rescheduling algorithms DE_JRM, FCFS and the ARS strategy for flat junction scenario is presented in Figure 4-24. It can be seen that, for flat junctions, the proposed train rescheduling algorithm DE_JRM can decrease the WAD significantly compared with FCFS and the ARS strategy. The average computation time of the algorithm DE_JRM is still around 2-3 seconds, which satisfies the computation time constraints for real time train rescheduling in junction areas. In terms of decreasing the WAD, the performance of the ARS strategy is not much better than the most commonly used and simplest junction control strategy, FCFS.

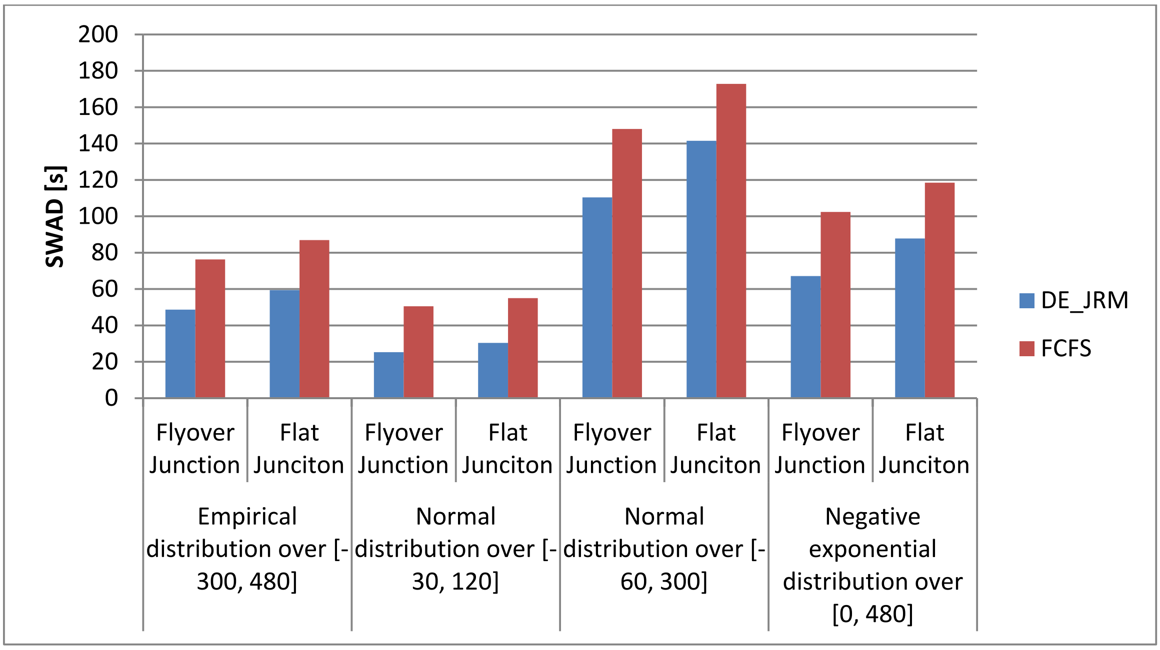

Figure 4-25 Comparison of SWAD with DE_JRM and FCFS in Flyover and Flat Junction Scenarios

Figure 4-25 shows the comparison of SWAD with the proposed train rescheduling algorithm DE_JRM and FCFS in flyover and flat junction scenarios. It is well known that the cost for infrastructure construction of upgrading flat junctions to flyover junctions is very high. It can be seen from Figure 4-25 that with the DE_JRM, the WAD in flat junction scenarios is even lower than the WAD with FCFS in flyover junction scenarios. This indicates that the application of advanced train traffic management and control systems will be an alternative approach compared with infrastructure upgrading from flat junctions to flyover junctions in terms of decreasing the WAD. The cost may also be much lower.

4.3 Conclusions

In this chapter, an innovative algorithm DE_JRM is proposed for solving real time train rescheduling problems formulated with JRM. Based on general Differential Evolution algorithms, the algorithm DE_JRM is derived with an addition of an innovative operation named “Modification”. The addition of the Modification operation makes DE_JRM an efficient tool for solving real time train rescheduling problems in junction areas. Following the proposal of the algorithm DE_JRM, a stochastic performance evaluation method based on Monte-Carlo simulation methodology was presented for performance evaluation of rescheduling algorithms. The performance of algorithm DE_JRM was evaluated with the presented stochastic performance evaluation method for a flyover junction scenario and a flat junction scenario, which are from a case study of the Thameslink Route, and the results were compared with a commonly used junction control strategy FCFS and an ARS strategy.

The performance evaluation results showed that, for both flyover and flat junctions, under all the four proposed train delay distributions the WAD can be decreased significantly with the train rescheduling algorithm DE_JRM compared with FCFS and the ARS strategy. The statistical WAD with the ARS strategy cannot be decreased significantly compared with FCFS. It indicates that the application of the ARS strategy cannot bring many benefits to decreasing the WAD in these scenarios.

With the application of algorithm DE_JRM, the WAD in flat junction scenarios is even lower than the WAD with FCFS and the ARS strategy in flyover junction scenarios. This shows that the application of advanced train traffic management and control systems will be an alternative approach compared with infrastructure upgrading from flat junctions to flyover junctions in terms of decreasing the WAD. This will bring the infrastructure management companies more benefits on cost decreasing.

Chapter 5. Real Time Train Rescheduling in Bottleneck Sections

In the last chapters, the formulation of the junction rescheduling problems and the proposed improved differential evolution algorithm DE_JRM were presented. The evaluation results have demonstrated the good performance of the DE_JRM for dealing with junction rescheduling problems. Compared with FCFS and a conventional ARS strategy, DE_JRM can decrease the WAD significantly.

On main line railways, bottleneck sections are often at the heart of networks, between junctions where different services converge from a range of origins or diverge to a variety of destinations. A typical urban railway configuration, with a bottleneck section and the associated approach tracks, is shown in Figure 1-1. Well known examples include Lines A and B/D of the RER network in Paris and the planned Thameslink and CrossRail networks in London, as well as the subsurface lines of London Underground. In some instances, services through the core sections also share tracks with services that do not travel through the core but influence the punctuality and reliability of trains approaching the junctions.

Bottleneck sections usually have the highest density traffic in a railway network. The margin time for service recovery from the event of a disturbance is very limited in bottleneck sections, which means there is no additional margin time which can be used for train rescheduling inside bottleneck sections. For the approaching routes leading to bottleneck sections, there will be more margins between trains due to the less dense traffic compared with bottleneck sections; this will provide more capacity to reschedule the trains approaching bottleneck sections.

In this chapter, the proposed methodology for solving real time train rescheduling problems in junction areas, including JRM and the algorithm DE_JRM is extended to model and solve real-time train rescheduling problems for bottleneck sections of railway networks. The Core Area of the Thameslink Route, which has a typical bottleneck section, was chosen as the case study, and the sketch map of the area layout is shown in Figure 5-1.

First, the train rescheduling problems in bottleneck sections in the event of disturbance is analysed and formulated with the JRM. Then the application of algorithm DE_JRM for solving the raised problems is introduced in detail. The last part of this chapter will focus on the case study of the Core Area of the Thameslink Route for the evaluation of the methodology.

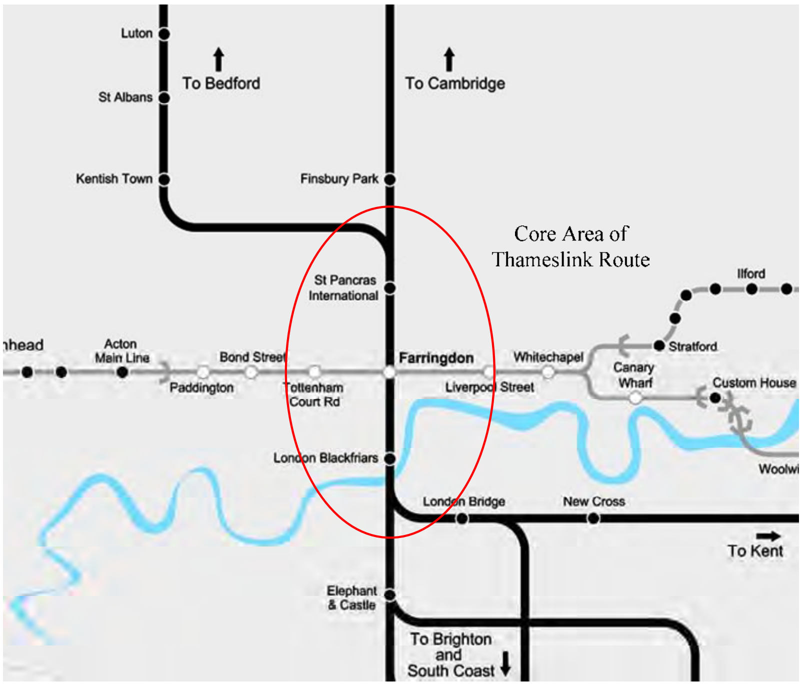

Figure 5-1 Sketch map of the Core Area of Thameslink Route (Websites of Network Rail)

5.1 Real Time Train Rescheduling Problems in Bottleneck Sections

Generally, as for the bottleneck sections shown in Figure 5-1, there are two junctions located at the both ends as the "portals" to bottleneck sections. The approaching trains from different origins converge to the bottleneck sections through the two portal junctions (arriving at one junction and leaving from another one). The real time train rescheduling problems of the portal junctions could be dealt with independently, or the real time train rescheduling problems of the portal junctions can be considered integrally. Whether the train rescheduling problems in portal junctions of bottleneck sections can be considered and solved independently or integrally depends on the coupling relationship of the portal junctions.

5.1.1 Coupling Relationship of Portal junctions of Bottleneck Sections

The coupling relationship of portal junctions describes the coupling level of train rescheduling of portal junctions of bottleneck sections. It depends on two factors of the railway operation circumstances, which are minimum operational running time of trains between two portal junctions, and the maximum time window width for train rescheduling. (Only the trains arriving at the junction point within a time window are considered for rescheduling.)

Generally if the two portal junctions are located remotely from each other, the real time train rescheduling problems of the two portal junctions may be treated independently, because the rescheduling decisions from one junction will not directly affect the rescheduling decisions for another portal junction. Otherwise for each junction, the rescheduling of approaching trains for one portal junction point needs to consider the movements of the trains approaching this portal junction from another portal junction, which is determined by the rescheduling decisions of the junction. A formal definition is given to determine whether real time train rescheduling of two portal junctions can be considered independently.

Let T_{min}^{R} denote the minimum operational running time of trains between the two portal junctions, T_{max}^{W} denotes the maximum time window width of train rescheduling.

(1) If $T_{min}^{R} > T_{max}^{W}$, then train rescheduling problems for two portal junctions are weakly coupled, as demonstrated in Figure 5-2, and can be solved independently.

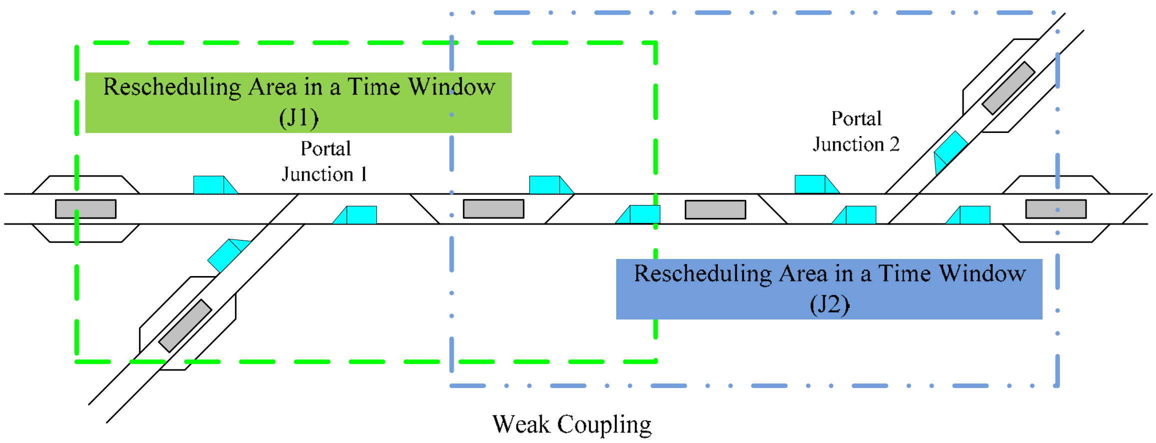

Figure 5-2 Weakly coupled train rescheduling for portal junctions

(2) If $T_{min}^{R} \leq T_{max}^{W}$, then train rescheduling problems for the two portal junctions are strongly coupled as demonstrated in Figure 5-3, and have to be solved integrally.

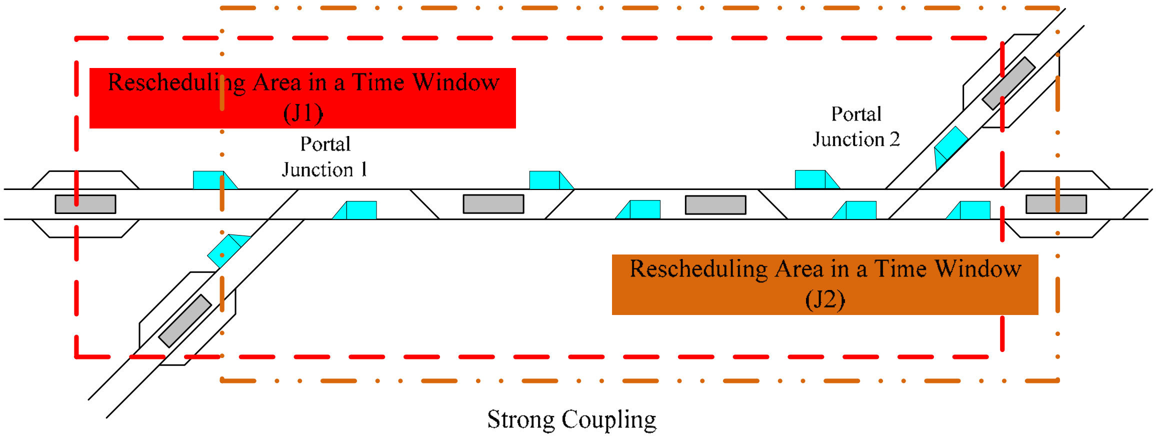

Figure 5-3 Strongly coupled train rescheduling for portal junctions

The train rescheduling problems of weakly coupled portal junctions of bottleneck sections can be modelled and solved independently with the JRM and DE_JRM. In this chapter, train rescheduling problems of strongly coupled junctions of bottleneck sections are studied.

5.1.2 Formulation of Train Rescheduling for Bottleneck Sections with JRM

As for the strongly coupled portal junctions, let p_1 and p_2 denote the two portal junction points. Event e_1 and e_2 denote the events of a train passing the portal junction points p_1 and p_2 in bottleneck section respectively.

According to JRM, the variables which need to be optimised in the event of disturbances are the rescheduled arrival times of the trains approaching the bottleneck section at the two junction points shown in Figure 5-4, which can be defined as $t_{i_r,e_1}^{p_1}$ and $t_{i_r,e_2}^{p_2}$.

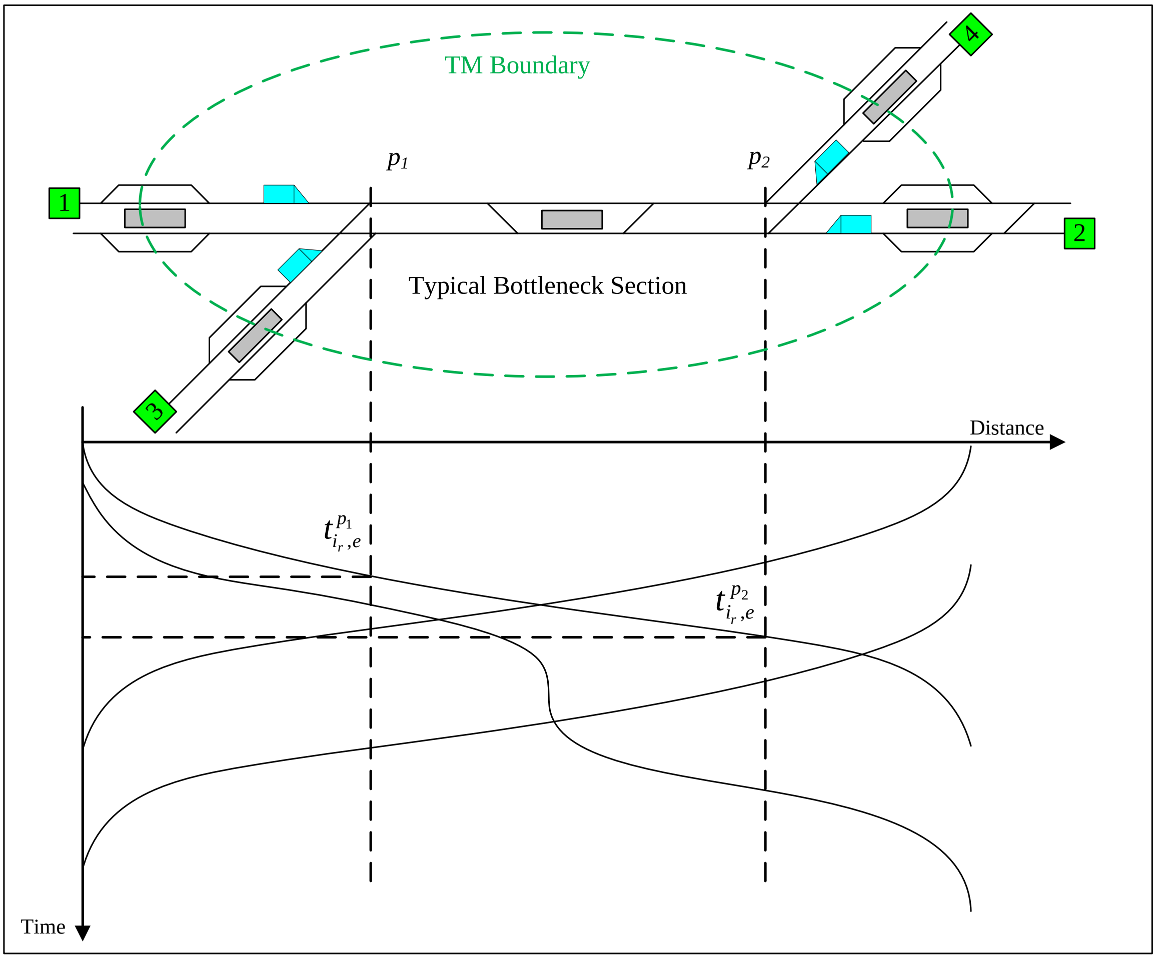

Figure 5-4 Train arrival times through a typical bottleneck section

As for the portal junctions located at the both ends of the bottleneck sections, these two junctions are strongly coupled. For each portal junction, the real time train rescheduling problems in the event of disturbances can be modeled with JRM with constraints which need to be complied with due to the train operation and control constraints in bottleneck sections.

As shown in Figure 5-4, for the train going through the bottleneck section from p_1 to p_2, the running time constraints in Equation 3-27 need to be written in the form of Equation 5-1 and Equation 5-2.

$$t^{p_1}_{i_r,e_1} \geq t^{b}_{i_r} + t^{b,p_1}_{i_r} \qquad i_{r,e_1} \neq 0, r \in R, e_1 \in E \qquad \textbf{Equation 5-1}$$

$$t_{i_{r,e_2}}^{p_2} \geq t_{i_{r,e_1}}^{p_1} + t_{i_r}^{p_1,p_2} \qquad i_{r,e_1}, i_{r,e_2} \neq 0, r \in R, e_1, e_2 \in E$$ **Equation 5-2**

where $t_{i_r}^{b,p_1}$ is the minimum running time of train i_r from control boundary b to junction point p_1, and $t_{i_r}^{p_1,p_2}$ is the minimum running time of train i_r from junction point p_1 to junction point p_2.

Accordingly, for the train going through the bottleneck section from p_2 to p_1, the running time constraints can be formulated as shown in Equation 5-3 and Equation 5-4.

$$t_{i_{r,e_2}}^{p_2} \geq t_{i_r}^{b} + t_{i_r}^{b,p_2} \qquad i_{r,e_1}, i_{r,e_2} \neq 0, r \in R, e_1, e_2 \in E$$ **Equation 5-3**

$$t_{i_{r,e_1}}^{p_1} \geq t_{i_{r,e_2}}^{p_2} + t_{i_r}^{p_2,p_1} \qquad i_{r,e_1}, i_{r,e_2} \neq 0, r \in R, e_1, e_2 \in E$$ **Equation 5-4**

where $t_{i_r}^{b,p_2}$ is the minimum running time of train i_r from control boundary b to junction point p_2, and $t_{i_r}^{p_2,p_1}$ is the minimum running time of train i_r from junction point p_2 to junction point p_1.

For each portal junction of the bottleneck sections, the train rescheduling problems in bottleneck sections can be modeled with JRM and the headway constraints shown in Equation 3-28 presented in Chapter 3.2 need to be modified with the constraints as shown in Equation 5-5 and Equation 5-6.

$$\begin{cases} t_{i_{r,e_1}}^{p_1} \geq t_{init}^{p_1} & e_1 = 1 \\ t_{i_{r,e_1}}^{p_1} \geq \left(t_{i_{r',e_1-1}}^{p_1} + t_{r_{e_1},r_{e_1-1}}^{H^{p_1}}\right) * h_{r_e,r_{e-1}}^{p_1} & e_1 > 1 \end{cases} \qquad i_{r,e_1}, i_{r',e_1-1} \neq 0;\ e_1 \in E;\ r, r' \in R$$

Equation 5-5

$$\begin{cases} t_{i_{r,e_2}}^{p_2} \geq t_{init}^{p_2} & e_2 = 1 \\ t_{i_{r,e_2}}^{p_2} \geq \left(t_{i_{r',e_2-1}}^{p_2} + t_{r_{e_2},r_{e_2-1}}^{H^{p_2}}\right) * h_{r_{e_2},r_{e_2-1}}^{p_2} & e_2 > 1 \end{cases} \qquad i_{r,e}, i_{r',e-1} \neq 0;\ e_2 \in E;\ r, r' \in R$$

Equation 5-6

Equation 5-5 and Equation 5-6 ensure that the headway of trains at junction point p_1 and p_2 must be kept due to railway operation and control constraints.

The other constraints of train rescheduling problems for the portal junction of bottleneck sections are the same with the constraints as shown in JRM.

The objective function used here is the sum of the WADs for two portal junctions, as presented in Equation 3-11. The objective of train rescheduling for bottleneck sections in the event of disturbances is to minimise the sum of the WADs for two portal junctions.

The constraints in the equations above show clearly that the train rescheduling problems for each portal junction of bottleneck sections are strongly coupled. The rescheduling decisions of one portal junction will form the constraints for the rescheduling decision making of another portal junction. Train rescheduling problems for bottleneck sections have to be solved with integral consideration of two strongly coupled portal junctions.

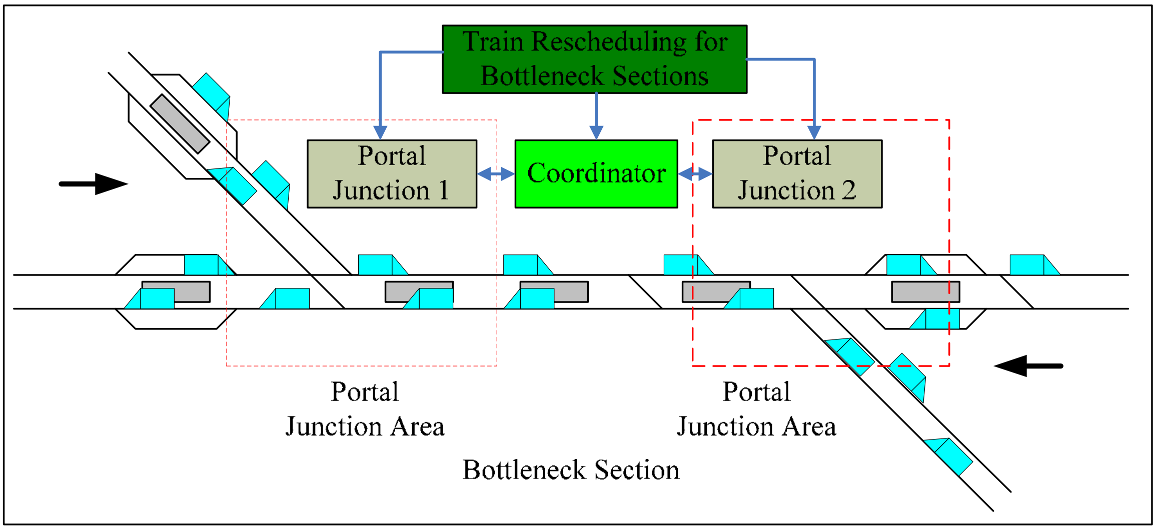

Figure 5-5 Framework of cooperative train rescheduling for portal junctions of bottleneck sections

In this chapter, a cooperative concept for solving train rescheduling problems for portal junctions of bottleneck sections is proposed, and the framework with cooperative concept is shown in Figure 5-5. The idea of the framework is to “coordinate” the train rescheduling decisions for two portal junctions, the detail of which is to regulate the train rescheduling decisions for two portal junctions to make sure that there is no conflict in the rescheduling decisions, which are the rescheduled train arrival times at two junction points. Here “no conflict” means that all the constraints in JRM can be kept during train rescheduling for two portal junctions.

5.2 Application of DE_JRM for Train Rescheduling in Bottleneck Sections

According to the framework shown in Figure 5-5, with the cooperative concept for train rescheduling for portal junctions of bottleneck sections, the solution algorithm for JRM, DE_JRM, can also be applied for solving rescheduling problems of bottleneck sections. In this chapter, the application of DE_JRM for train rescheduling in bottleneck sections is described.

As analysed in the previous section, the train rescheduling for the two portal junctions which needs to be handled as an integral problem can also be solved with the algorithm DE_JRM. The variables to be optimised in train rescheduling problems for portal junctions of bottleneck sections are $t^{p_1}_{i_r,e_1}$ and $t^{p_2}_{i_r,e_2}$. The vary parameter vector for solutions is defined as$\{X = [t^{p_1}_{i_r,e_1}, t^{p_2}_{i_r,e_2}] | r \in R; e_1, e_2 \in E\}$. The processing flow chart of DE_JRM is shown in Figure 4-2. There is no difference in the Mutation, Crossover and Selection operations, only the Modification operation needs to be modified in terms of the features of train rescheduling problems for bottleneck sections. As described in Chapter

4.1 , the stochastic operation of Mutation and Crossover cannot ensure that the generated solution parameters can comply with the constraints in JRM. Especially for the train rescheduling problems for two portal junctions of bottleneck sections, there are extra constraints which need to be complied with because of the coupling relationship of two portal junctions. The functions of Coordinator shown in Figure 5-5 are realised in the Modification operation.

There are three processes in the Modification operation, as shown in Figure 4-3. Because of the constraints in train rescheduling problems for bottleneck sections, more modification rules need to be added into the Modification operation. As for running time constraints, two more rules, shown in Equation 5-7 and Equation 5-8 need to be added for Running Time Check in the Modification operation.

$$IF\ t_{i_{r,e_2}}^{p_2} < t_{i_{r,e_1}}^{p_1} + t_{i_r}^{p_1,p_2}, THEN\ t_{i_{r,e_2}}^{p_2} = t_{i_{r,e_1}}^{p_1} + t_{i_r}^{p_1,p_2} \qquad i_{r,e_1}, i_{r,e_2} \neq 0, r \in R, e_1, e_2 \in E$$

Equation 5-7

$$IFt_{i_{r,e_1}}^{p_1} < t_{i_{r,e_2}}^{p_2} + t_{i_r}^{p_2,p_1}, THEN\ t_{i_{r,e_1}}^{p_1} = t_{i_{r,e_2}}^{p_2} + t_{i_r}^{p_2,p_1} \qquad i_{r,e_1}, i_{r,e_2} \neq 0, r \in R, e_1, e_2 \in E$$

Equation 5-8

The train rescheduling problems for bottleneck sections include the train rescheduling for two portal junctions, as for Sequence and Headway Validity Check and Modification and Junction Headway Control Check and Modification in the Modification operation, all the rules need to extend to apply for two portal junctions. In Junction Headway Control Check and Modification, the strategy selecting procedure is undertaken as described in Chapter 4.1.3, and same strategy chosen from the three strategies (FCFS, Priority based Modification, and Weighted Delay based Modification) in terms of the selecting probabilities is applied to modify the rescheduled train arrival times in the trial parameter vectors for both portal junctions. For example, if FCFS has been chosen as the modification strategy in Junction Headway Control Check and Modification for one trial parameter vector, then as for the train arrival times $t_{i_r,e_1}^{p_1}$ and $t_{i_r,e_2}^{p_2}$ at portal junction points p_1 and p_2 in the trial parameter vector, FCFS rule is applied for modification of both $t_{i_r,e_1}^{p_1}$ and $t_{i_r,e_2}^{p_2}$.

In Selection operation, according to the objective function of train rescheduling for bottleneck sections, which is the sum of WADs for two portal junctions, the better solutions will be selected from all the trial parameter vectors generated after Mutation, Crossover and Modification as the parent parameter vectors for evolution of the next generation until the algorithm converges.

5.3 Case Study on the Core Area of Thameslink Route for Evaluation

A case study on the Core Area of the Thameslink Route has been undertaken for the evaluation of the presented method and solution algorithms DE_JRM. The configuration sketch map of the Core Area of the Thameslink Route is shown in Figure 5-6.

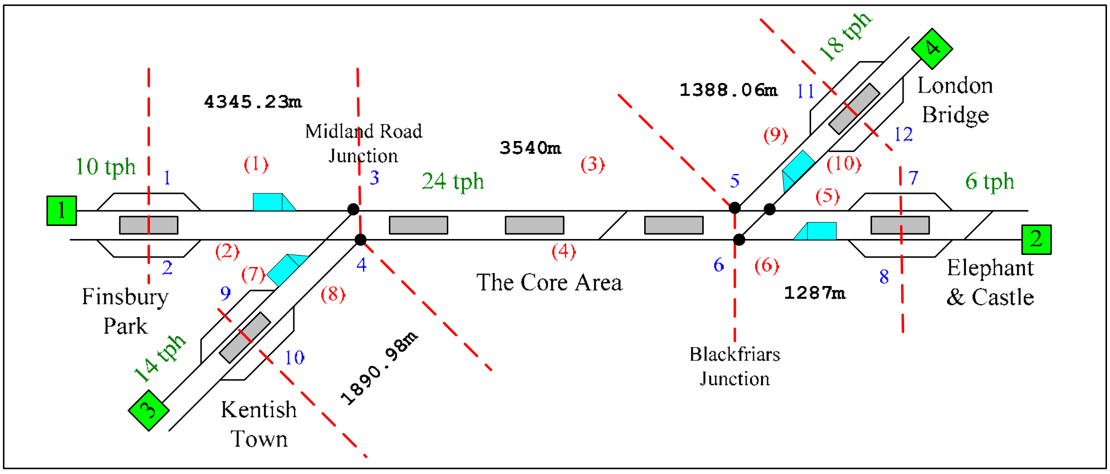

Figure 5-6 Configuration sketch map of the Core Area of Thameslink route

The scenario chosen for case study has typical bottleneck sections with trains from four different origins converging into the bottleneck sections. There are two junctions located at both ends of the Core Area as “portal” junctions, which are Midland Road Junction and Blackfriars Junction. As planned in the Thameslink Programme, the Midland Road Junction will be built as fly-over grade and Blackfriars Junction will continue to be a flat junction. The traffic density in each section for each direction at the programme stage KO2 is marked in the figure with the unit of tph (trains per hour). The traffic density through the bottleneck section will be very high. It is expected to run 24 trains per hour in each direction at peak hours through the bottleneck section in 2018, as shown in Figure 5-6. The service interval of the trains is only 150 s, and there are three stations inside the bottleneck section, which means the margin for train rescheduling inside of the bottleneck section will be very limited. There will be more margins on the approaching routes which can be used for train rescheduling for the bottleneck section in the event of disturbances.

The boundary of the traffic management area has been chosen at the departure points of trains at Kentish Town, Finsbury Park, London Bridge and Elephant & Castle stations.

The trains chosen for simulation are formed of 3*4-car Class 377 sets, as described in Table 4-1.

The basic parameters for algorithm DE_JRM and simulation scenarios are listed as follows:

- ❖ The ratio of weights assigned for the trains from Kentish Town station, the trains from Finsbury Park station, the trains from station London Bridge and the trains from Elephant & Castle is set to 7:5:9:3.
- ❖ Number of population in DE_JRM is set to be 200.
- ❖ Mutation factors are set to be $F_1 = F_2 = 0.9$.
- ❖ Crossover factor CR is set to be 0.95.

The statistical evaluation methodology presented in Chapter 4.2.4 has also been applied for the evaluation of DE_JRM. The four different delay distributions were applied to generate large numbers of delay scenarios for statistical evaluation. The probability distribution of the four delay distributions are presented in Chapter 4.2.4.1.

The algorithm DE_JRM has been applied for the generated delay scenarios to calculate the WADs with DE_JRM, and the Statistical WAD (SWAD) was calculated after large numbers of the simulation experiments. FCFS and the ARS strategy have also been applied in the simulation experiments, the calculated SWAD was compared with the results of DE_JRM.

For each rescheduling algorithm, 10,000 simulation experiments based on each train delay probability distribution at the TM boundary have been undertaken for statistical evaluation of the case study. The SWAD of the algorithm DE_JRM, FCFS and the ARS strategy for each train delay distribution is listed in Table 5-1.

Rescheduling Algorithm / Delay Distribution	DE_JRM	FCFS	ARS
Empirical distribution over [-300, 480]	57.5916	87.5209	86.3568
Normal distribution over [-30, 120]	31.3990	55.9281	55.0247
Normal distribution over [-60, 300]	142.5045	174.4756	167.3216
Negative exponential distribution over [0, 480]	88.6148	119.7492	117.2685

Table 5-1 SWAD of algorithm DE_JRM, FCFS and ARS

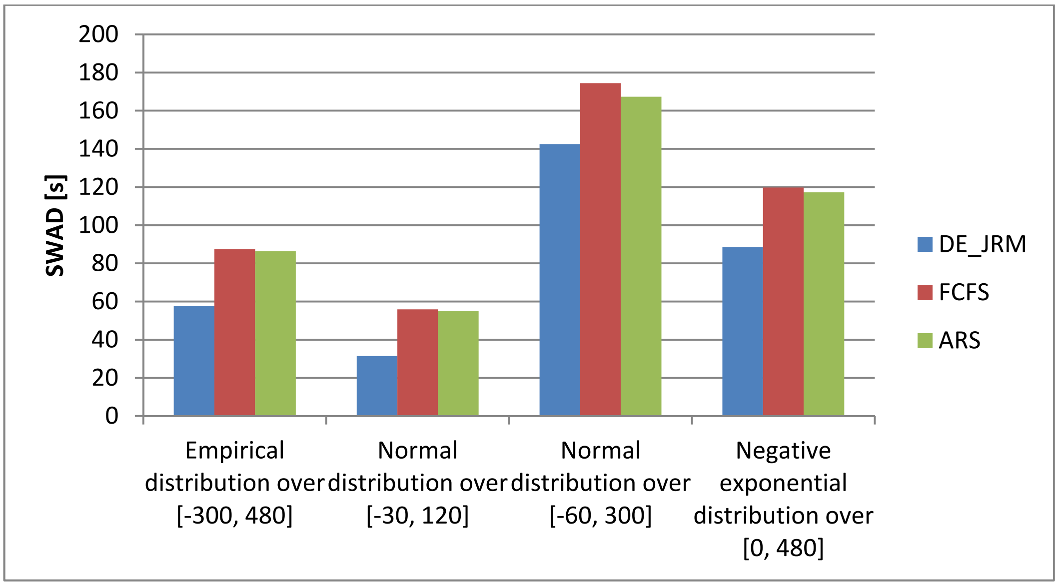

Figure 5-7 Comparison of SWAD with Rescheduling Strategies for Bottleneck Section Scenario

The comparison of Statistical WAD with rescheduling algorithms DE_JRM, FCFS and the ARS strategy in the bottleneck section scenarios is shown in Figure 5-7. As for all the four train delay distributions, the WAD is decreased significantly with the proposed train rescheduling algorithm DE_JRM compared with FCFS and the ARS strategy. The average computation time of the algorithm DE_JRM is around 2-3 seconds, which is more than satisfactory for the real time applications of train rescheduling in bottleneck sections. The proposed methodology including JRM and the train rescheduling algorithm DE_JRM has been proved to be an efficient approach for solving real time train rescheduling problems in bottleneck sections.

5.4 Conclusions

In this chapter, the proposed methodology for solving real time train rescheduling problems in junction areas including JRM and the algorithm DE_JRM was extended to model and solve real-time train rescheduling problems for bottleneck sections of railway networks. The details of the methodology extension of the JRM and the algorithm DE_JRM have been introduced. A case study on the Core Area of Thameslink Route has been undertaken for the evaluation of the method and the associated solution algorithm DE_JRM.

The evaluation results show a good performance of the proposed methodology for solving real time train rescheduling problems in bottleneck sections based on the JRM and the algorithm DE_JRM. As for all four train delay distributions, the WAD is decreased significantly with the proposed train rescheduling algorithm DE_JRM compared with FCFS and the ARS strategy. The average computation time of the algorithm DE_JRM is around 2-3 seconds, which is more than satisfactory for the real time applications of train rescheduling in bottleneck sections.

Chapter 6. Integrated System Architecture for Train Rescheduling and Control

In the previous chapters, a methodology for solving train rescheduling problems in junction areas as well as in bottleneck sections was presented. The methodology focuses on decision making in a traffic management system, which provides optimised rescheduled timetables when delays occur to trains in the territory controlled by the system. Feasible rescheduled timetables for trains must be generated and implemented by railway operational systems, such as traffic management systems, train control systems, and drivers. To provide a clear understanding on how to implement the proposed methodology in real railway operations, in this chapter, an integrated system architecture is proposed for train rescheduling and control in junction areas and bottleneck sections. The objective is to deliver an integrated traffic management and train control system which can provide optimised traffic control and operational decisions for bottleneck sections of mainline railways.

6.1 System Architecture

In mainline railway systems, the safe headway between following trains is ensured by the signalling system, which is a safety critical system that must comply with the safety requirements of Safety Integration Level 4 (SIL 4) (Charlwood, Turner et al. 2004). Under the supervision and control of railway control and signalling systems, drivers control the speed of trains to make the train arrive at specific locations on time according to the given nominal timetables. Inevitably, there are many disturbance inputs to railway systems, such as driving errors, passenger accidents, weather conditions etc. Because of the disturbances, conflicts between trains may occur. These conflicts must be detected as early as possible and resolved by traffic regulation and management in terms of the objective functions regulated in operational strategies. According to the different demands of railway operators, the main objectives of railway traffic management systems may be different. Some of the systems are required to minimise train delays weighted by train classes, some of the systems are applied to minimise the penalties paid by infrastructure managers to train operators to compensate for train delays or other incidents, and some of them are used to maximise the capacity for passengers and cargo deliveries.

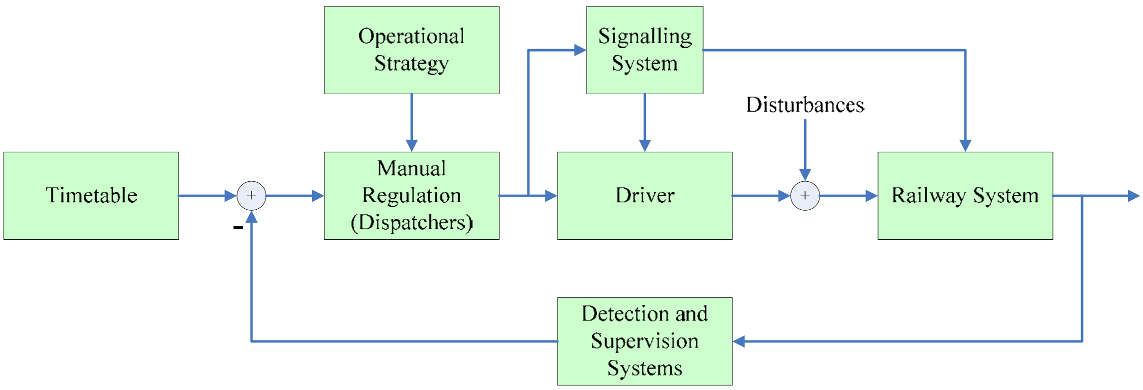

Figure 6-1 Conventional railway traffic management and train control loop

6.1.1 Traffic Control Loop

Figure 6-1 shows a conventional railway traffic management and train control loop that applies to most mainline railways. The nominal timetable feeds into the control loop as the input, which all the trains are expected to follow. Any deviations of the train movements from the nominal timetable will be detected by train detection and supervision systems such as track circuits, train positioning systems etc, and the deviations must be minimised by rescheduling the trains in the event of disturbances according to the objectives of the railway operators. In most conventional railway systems, manual regulation is generally applied to minimise the deviations of train movements from the timetable by coordinating signalling systems and drivers according to operational strategies. In the conventional traffic control loop shown in Figure 6-1, the dispatchers in the control centre have an important role for railway traffic management in the event of disturbances using their experience of dealing with incidents. The trains can be rescheduled through re-routing and signal controls etc. As can be seen, there is no optimisation considered in train rescheduling when disturbances occur and train speed control mainly relies on the experience of traffic dispatchers and train drivers. Experienced dispatchers and drivers can keep the timetable better than ones who are lack of operational and driving experiences.

For a railway network with high density traffic e.g., bottleneck sections, it will be very difficult for the dispatchers to make rescheduling decisions because of the large number of trains and the complex interactions between trains in control regions. In addition, dispatchers can hardly predict the late-on potential conflicts which could happen due to the regulated plans. All these require an advanced traffic management system which can provide real time optimised rescheduling decision support and can look ahead for potential conflicts, as far as possible.

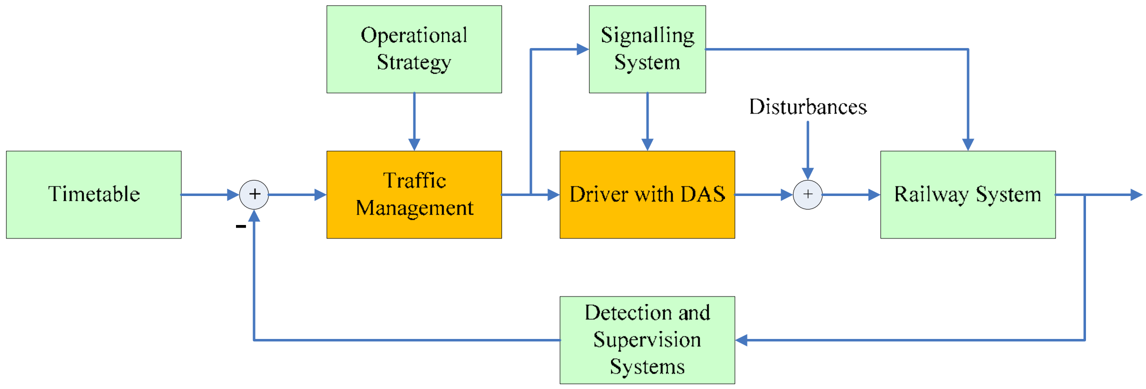

Figure 6-2 Advanced rail traffic management and train control loop

A systematic architecture for integration of advanced railway traffic management and train control for bottleneck sections with high density traffic is proposed in this thesis, the basic control loop of which is shown in Figure 6-2. Compared with conventional traffic management and the train control loop shown in Figure 6-1, the main upgrade is the implementation of an advanced Traffic Management (TM) system and a Driver Advisory System (DAS) to assist drivers with train speed control. The system architecture is shown in Figure 6-3. Manual regulation by dispatchers is replaced by advanced Traffic Management systems. The main functions of the traffic management system are deviation detection and adjustment, and conflict detection and resolution. Inside the traffic management systems, advanced algorithms like DE_JRM can be applied to generate rescheduled timetables for the trains in the control regions. The drivers will receive more information from the DAS, which obtain the rescheduled timetable from traffic management systems and, provide real time optimised advisory information (advisory speed, coasting points, traction prompts, braking prompts) to help train drivers achieve conflict free and energy saving train operations.

6.1.2 Data Flow of System Architecture

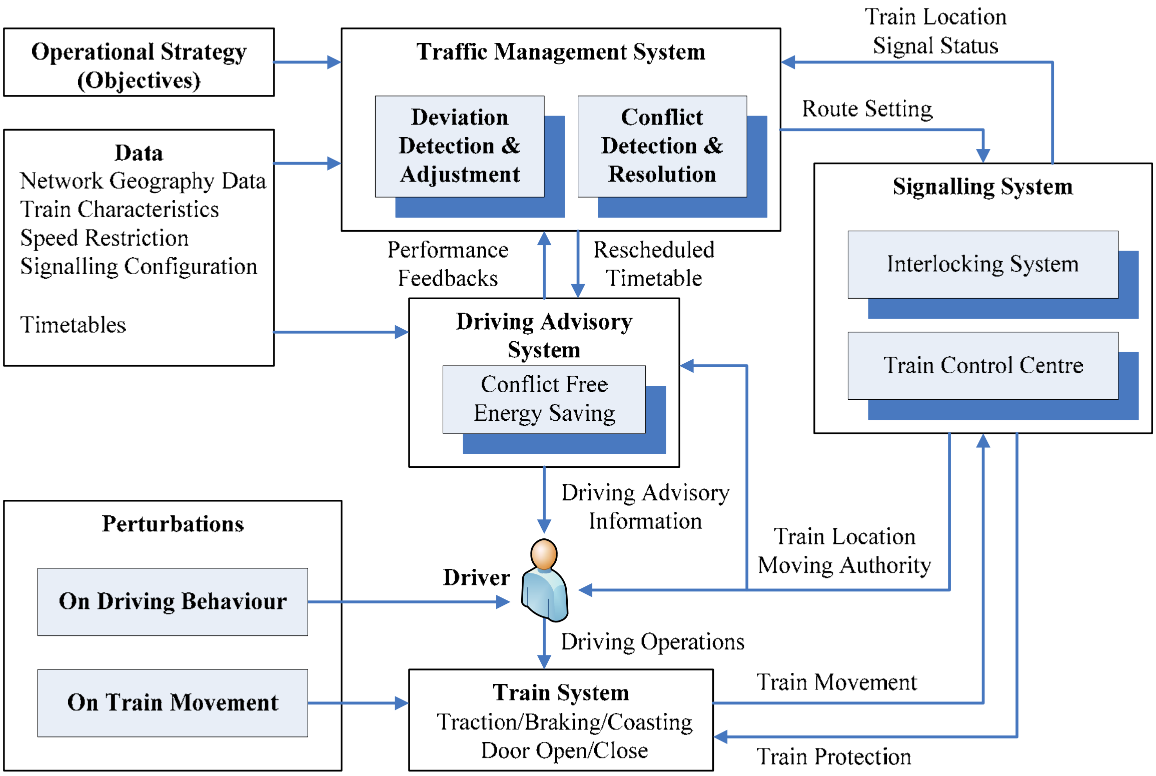

Figure 6-3 System Architecture for integration of traffic management and train control

The data flow of the system architecture for the integration of traffic management and train control for bottleneck sections is shown in Figure 6-3. The general functional requirements and data interfaces of the core systems in the system architecture are as follows.

1) Traffic Management System

Functional requirements:

- Acquire static and dynamic information from the operation
- Detect the deviation between train movement and nominal timetable
- Adjust timetable to deal with the deviation

- Detect potential conflicts
- Resolve conflicts with advanced algorithms
- Allow manual traffic management interface as a backup mode

Data interface:

To achieve the functions listed above for the traffic management system, the following data are required:

- Knowledge of railway network geography data, including network topology, gradients, curvature, line speed restriction
- Train characteristics data
- Signalling configuration data, including signal positions, route information
- Timetable information
- Real time train location information

Output data:

- Rescheduled timetables for trains
- Route setting request to signalling system

2) Driving Advisory System

Functional requirements:

- Determine whether the rescheduled timetable from the traffic management system can be achieved by the train
- Calculation of energy efficient train trajectory to achieve target times according to the rescheduled timetable

- Monitor the train movement and provide advisory information to help drivers follow the train trajectory which satisfy the rescheduled timetable

Data interface:

Data required:

- Route information during the train journey including geography data, gradients, curvature, line speed restriction
- Train characteristic data
- Signalling configuration data
- Rescheduled timetables from the traffic management system
- Real time train location information

Output data:

- Driving advisory information to drivers (advisory speed, coasting points, traction prompts, braking prompts)
- Performance feedbacks to traffic management system

3) Signalling System

Functional requirements:

- Ensure safe interval between trains in sections
- Ensure no conflicting routes are set for trains at stations
- Train speed protection
- Detection of train positions

Data interface:

- Route setting request from traffic management system

- Real time train locations and signal status to traffic management system
- Real time train locations and moving authority to DAS and driver
- Train protection actions to train system

4) Driver

Functional requirements:

- Manually control train speed
- Door control
- Monitor train movement

Data interface:

- Driving advisory information from DAS
- Real time train location and moving authority from signalling system
- Driving actions (traction/braking/coasting, door open/close) to train system

In the system architecture shown in Figure 6-3, an important factor which affects driver and train system is that of perturbations. Because of the perturbations on train drivers and the train system (as well as the track system, which is not shown in Figure 6-3), control deviations may occur, so that train movement cannot fully follow the rescheduled timetable. A control feedback loop shown in Figure 6-2 is applied in the system architecture to regulate and decrease the deviations.

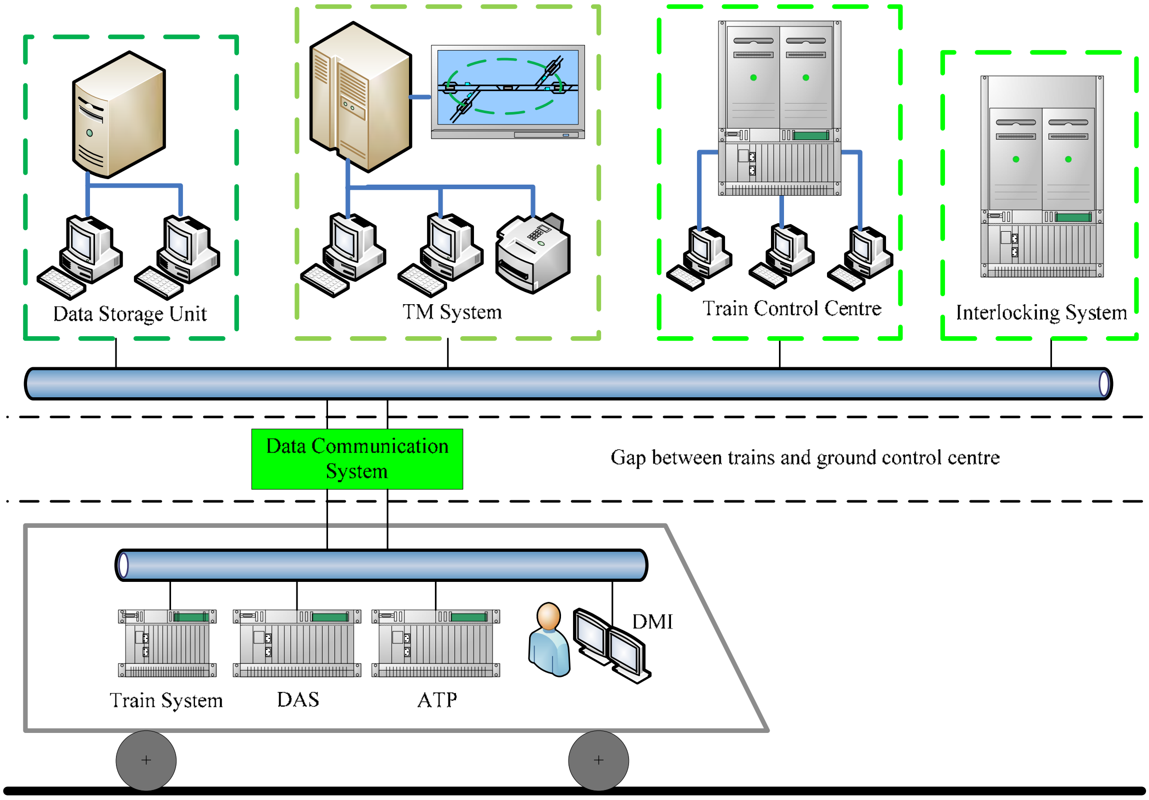

Figure 6-4 System configuration for traffic management and train control

Based on the system architecture shown in Figure 6-3, a system configuration for integration of traffic management and train control for bottleneck sections is presented in Figure 6-4. Generally, the sub-systems can be classified into three parts: Ground control systems, Vehicle on-board control systems and Data communication system for safe and reliable data transmission between vehicle on-board control systems and ground control systems.

6.2 Traffic Control Process and System Configuration

6.2.1 Traffic Control Process

For bottleneck sections with high traffic density, minor disturbances to one train can cause long consequential knock-on delays to following trains. Real time traffic management and train control in the event of disturbance will be expected to decrease the weighted average delays (WAD). A real time traffic management and train control process is described in this chapter in terms of the system configuration shown in Figure 6-4.

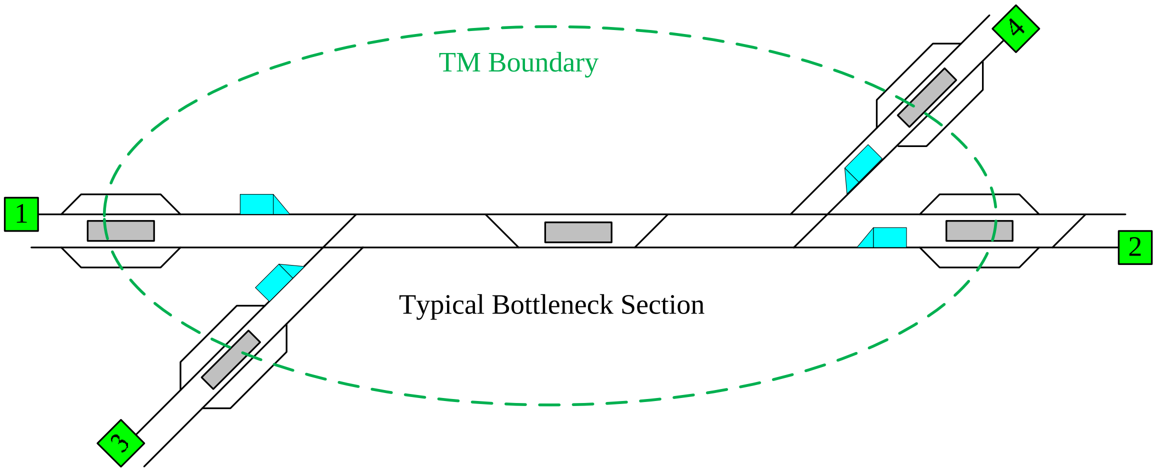

Figure 6-5 Traffic management boundary of typical bottleneck sections

Due to the capability limit of information processing, there must be a control boundary for railway traffic management systems. For example, Figure 6-5 shows a traffic management boundary definition for a typical bottleneck section. It is important that the boundary is defined sufficiently large for effective control and is kept small enough to allow efficient computation of better strategies. Once the train enters the boundary, the traffic management system will make train rescheduling decisions with consideration of the entering train, and the train driver is expected to receive rescheduled timetables via the driving advisory information. A general process of traffic management and train control for bottleneck sections or junction areas is described as follows.

1. When a train is detected by the signalling system that is due to enter into the traffic management boundary, the traffic management system checks the train arrival time at the boundary;
2. According to the states of all the trains in the traffic management territory, the algorithm DE_JRM in the traffic management system will generate an updated optimised rescheduled timetable for all the trains in the traffic management territory without conflicts;
3. The new updated rescheduled timetable will be sent to all the DAS on trains via a data communication system;
4. Once the DAS on each train has received an updated rescheduled timetable for each train respectively, the rescheduled timetables will be re-checked and used to generate the optimised train trajectories;
5. For each train, driving advisory information will be provided to the driver via an appropriate DMI according to the generated optimised train trajectory.
6. The drivers are expected to follow the driving advisory information provided by the DAS to take the driving actions including acceleration, braking and coasting.
7. Because of the ongoing perturbations that impact on drivers and railway operations, it is possible for the train movement not to follow exactly the optimised train trajectories, which satisfy the rescheduled timetables.
8. Once minor deviations occur between the train movements and optimised train trajectory, the DAS will cope with the deviation to recover from the deviation and

keep the train movement aligned with the latest received rescheduled timetable from the traffic management system.

9. When major deviations occur, and the DAS identifies that the rescheduled timetable cannot be performed by the train, the DAS will send feedback information to the traffic management system as a notification. The traffic management system will carry on from Step 2.

Step 1 to Step 9 describes the normal processes of traffic management and train control for bottleneck sections. As traffic management systems are not safety critical systems as signalling subsystems, any possible non-safe outputs from traffic management system will be identified and prohibited by the signalling subsystem. As computer based systems, it is also possible for traffic management systems to lose functionalities due to hardware or software logic errors. It will be necessary for traffic management systems to have additional interfaces to railway traffic dispatchers who can manually manage railway traffic in the event of specific situations.

6.2.2 System Configuration

Figure 6-4 provided a general system configuration of integrated systems of traffic management and train control for bottleneck sections. Compared with conventional railway control systems, the new feature is the introduction of the traffic management system and the driving advisory system. The key technologies applied in these systems are presented here.

- ❖ The key technology in the traffic management system is the train rescheduling algorithms, which are used for solving real time train rescheduling problems for bottleneck sections. The proposed DE_JRM will be one of the choices.

- The main requirement of the signalling system for integration into the system configuration shown in Figure 6-4 is that the signalling systems have the capability of providing date interfaces with the traffic management systems and the DAS. Existing signalling systems with ATP in Britain can be integrated into the system configuration. Communication based train control systems (CBTC) are also choices which can be applied. In Britain, the infrastructure management company Network Rail has committed to ERTMS/ETCS as the future basis signalling system for mainline railways, ETCS level 2 with GSM-R would be a good choice in terms of cost and interoperability. Actually, ETCS level 2 is also a kind of system realisation of CBTC systems.
- Safe, reliable and fast data transmission between trains and ground control centres is essential in the proposed system configuration. Without safe data transmission protocols, the common communication approaches cannot be used for transmission of train control data due to high safety, reliability and availability requirements for train control systems. There are several existing choices, for instance, Wi-Fi with safe data transmission protocols, and GSM-R with EuroRadio.
- The Driving Advisory System (DAS) can be implemented and integrated with other train onboard systems. An example of a DAS DMI is shown in Figure 6-6, which is a standard ETCS DMI with advisory speed information (marked as a red dot on the speed panel) to drivers. The system is implemented on trains on the Lotschberg Tunnel Line in Switzerland. Another option is for it to be implemented as a stand-alone system installed in the cab, or as a portable device carried by the drivers.

Figure 6-6 ETCS DMI with Advisory Speed

6.2.3 Application of ATO

As the development of computer and control technology, Automatic Train Operation (ATO) systems have been fully or partly applied to some of the urban transit lines and underground lines, including the Victoria Line on London Underground, Paris Metro Line 14, Circle MRT Line Singapore and Beijing Subway Line 10 etc. An ATO subsystem is an obvious candidate for integration of traffic management and train control for bottleneck sections. The benefits and shortcomings of the application of ATO are discussed in this section.

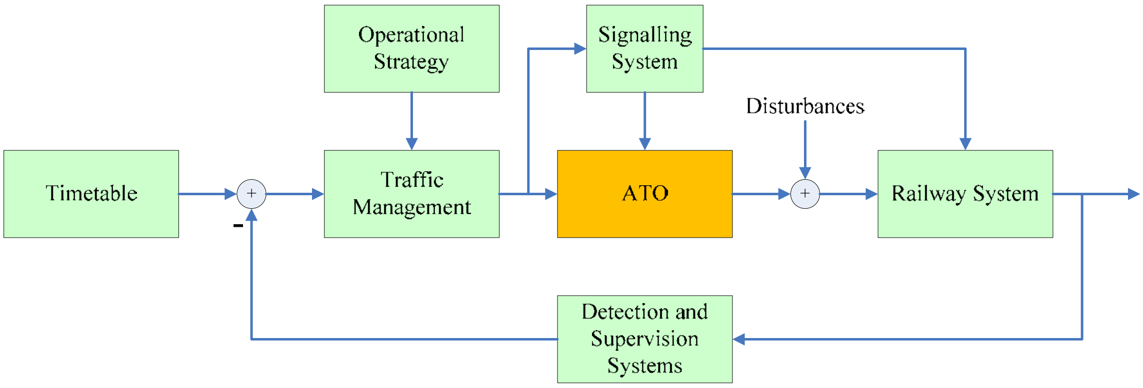

Figure 6-7 Control loop with an integrated ATO application

Figure 6-7 shows the control loop with an integrated ATO system for traffic management and train control for bottleneck sections. The ATO system in the control loop replaces the DAS and the train drivers, although many of today's systems still rely on a driver to initiate door closure. The rescheduled timetable generated in the event of disturbances by the traffic management system will be sent to the ATO system on trains instead of the DAS. The functions carried out by the DAS, train speed control and door control by drivers will be implemented as part of the ATO system.

The application of ATO would bring the following benefits:

- Minimum deviation of train speed control with optimised train trajectory
- Maximum use of available safe line speed profile
- More margin time available due to more consistent train performance without manual train speed control
- Accurate stopping at stations
- Reduce door opening and closing time
- Release driver from work of train speed control, this allows drivers to concentrate more on safety issues.

The application of ATO for train control could also have some shortcomings as follows:

- ❖ The control logic of ATO is pre-coded by system designers, and thus is not comparable with experienced train drivers in terms of the capability of dealing with complicated train operation scenarios, especially in the event of disturbances.
- ❖ The application of ATO in bottleneck sections and junction areas of the railway network may have consistence problems with train operations outside of these areas. For instance, system transitions need to be done by the drivers when trains are passing control boundaries. This could make the train operations more complicated for the whole of the railway network.
- ❖ It is necessary to have drivers as a backup mode for the ATO system on trains in case of system failures.

As discussed above, there are benefits and shortcomings with the application of ATO for train control in junction areas and bottleneck sections. For different railway scenarios, the choice of application of ATO needs to be balanced. From a technical perspective, the application of DAS with train drivers may be an easier, more reliable and robust approach compared with application of full ATO due to the complexity of train traffic operations on mainline railways. While the application of ATO systems could yield benefits in bottleneck sections, these have very high density traffic flow with very limit allowance times.

6.3 Conclusions

To implement the proposed methodology of train rescheduling for junction areas and bottleneck sections, an integrated system architecture of traffic management and train control for junction areas and bottleneck sections of mainline railways has been presented. The system architecture creates an integrated system with the structure of

Traffic Management system + DAS + Driver + Signalling system. Data flow among sub-systems is described in the system architecture, and system functional requirements and data interfaces have been specified briefly. The presented system architecture reflects the general traffic control process. As for practical application of the system architecture, the system configuration and technical realisation has also been discussed in this chapter. An extension to the approach was proposed with application of ATO to replace DAS and Driver. The benefits and shortcomings with application of ATO have also been discussed.

The author recommends the system architecture of Traffic Management system + DAS + Driver + Signalling system for integrated train rescheduling and control in junction areas and bottleneck sections, where the traffic flow density is not very high and allowance times in the nominal timetables are sufficient. The application of ATO systems could have more benefits in bottleneck sections, those have very high density traffic flow with very limit allowance times in the nominal timetables like peak hour traffic on metro lines. Advanced algorithms for solving train rescheduling problems such as DE_JRM can be applied in the traffic management system in the proposed system architecture.

Chapter 7. Conclusions and Future Work

7.1 Conclusions

To better manage railway traffic and control trains in highly utilised areas of railway networks for decreasing train delays in the event of disturbances, a systematic methodology has been proposed in this thesis for modelling and solving real time train rescheduling problems in junction areas and bottleneck sections. The presented systematic methodology mainly includes problem modelling, innovative solution algorithms, performance evaluation methods and system implementation architecture.

Firstly, to better understand the train rescheduling problems in junction areas, a formal mathematical model, Junction Rescheduling Model (JRM) is proposed in this thesis. The JRM is a MIP problem and is regarded as a typical NP-hard problem in mathematics.

An innovative algorithm DE_JRM is proposed for solving real time train rescheduling problems formulated with JRM. Based on general Differential Evolution algorithms, the algorithm DE_JRM is derived with the addition of an innovative operation named "Modification", which makes DE_JRM an efficient tool for solving real time train rescheduling problems in junction areas.

In addition, a stochastic performance evaluation method based on Monte-Carlo simulation methodology is introduced to evaluate the rescheduling algorithms. The performance of the algorithm DE_JRM is evaluated with the stochastic performance evaluation method for a flyover junction scenario and a flat junction scenario, which are from a case study of the Thameslink Route, and the results are compared with a commonly used junction control strategy, FCFS, and a conventional ARS strategy.

The evaluation results show that, for both flyover and flat junctions, under four proposed train delay distributions, the Weighted Average Delay (WAD) can be reduced significantly with the train rescheduling algorithm DE_JRM compared with FCFS and the ARS strategy. The average computation time of the algorithm DE_JRM is around 2-3 seconds, which is more than satisfactory for the real time applications of train rescheduling. The statistical WAD with the ARS strategy cannot be decreased significantly compared with FCFS. It indicates that the application of the ARS strategy cannot bring many benefits to decreasing the WAD in these scenarios.

It was also found that, with the application of algorithm DE_JRM, the WAD in flat junction scenarios is even lower than the WAD with FCFS and the ARS strategy in flyover junction scenarios. The application of advanced train traffic management and control systems has been shown to be an alternative approach, compared with infrastructure upgrading from flat junctions to flyover junctions in terms of decreasing the WAD. This could bring the infrastructure management companies more benefits in reducing cost.

The proposed methodology for solving real time train rescheduling problems in junction areas including JRM and the algorithm DE_JRM is also extended to model and solve real time train rescheduling problems for bottleneck sections of railway networks. The simulation experiment results show a good performance of the proposed methodology. As for all the four train delay distributions, the WAD is decreased significantly with the proposed train rescheduling algorithm DE_JRM compared with FCFS and the ARS strategy.

Finally, an integrated system architecture for traffic management and train control is introduced for system implementation of the proposed methodology of train rescheduling in junction areas and bottleneck sections on mainline railways. The system architecture is composed with the structure of Traffic Management system + DAS + Driver + Signalling system. An extension to the approach is proposed with application of ATO to replace DAS and Driver. The benefits and shortcomings with application of ATO have also been discussed.

7.2 Future Work

The author focused on modelling and solving the real time train rescheduling problems in junction areas and bottleneck sections on mainline railways. The further tasks are suggested to extend the work.

(1) The further evaluation of the algorithm DE_JRM for more scenarios will be worthy work before practical algorithm applications. This will refer to setup of standard benchmarks and comparison with other possible advanced algorithms for real time train rescheduling.

(2) It will be significant and worthy to extend the methodology to solve the train rescheduling problems in a large railway network. The performance of the algorithm DE_JRM, including goodness of the found optimised solutions and computation times, should be validated for large scale real time train rescheduling problems as one area of future work.

(3) The specific relationships between the impact factors in railway timetabling and train rescheduling such as margin times, recovery times, knock on delays, rescheduling algorithms etc. is very complicated, and hard to describe quantitatively. One area of

future work is to investigate the relationships between the impact factors based on the work that has been done in this thesis and try to describe these relationships formally and quantitatively. This will give a more clear understanding of the relationships of these impact factors, and it will help planners of railway timetabling, dispatchers in railway traffic control centres and other railway planning and operation people to improve railway services.

(4) The system methodology presented in this thesis is developed in a laboratory environment with computer simulation experiments. Because of the limitations of system modelling and simulation, there are still some aspects which have not been modelled and simulated in the thesis. A practical validation of the methodology could be necessary, and the algorithms and system architectures proposed in this thesis need to be developed and implemented in real railway traffic management systems in future.

The author is one of research members of a large cooperative EU FP-7 project, “ON-TIME”, the detailed information of the project can be found at (EU-Project-FP7 2011). This makes it possible to apply the methodology presented in this thesis into practice in the near future.

Appendix A. Description of a Conventional ARS Strategy

A conventional Automatic Route Setting (ARS) strategy is applied for performance comparison with the algorithm DE_JRM and FCFS in the simulation environment in this thesis. The principle of the ARS strategy is described in this section.

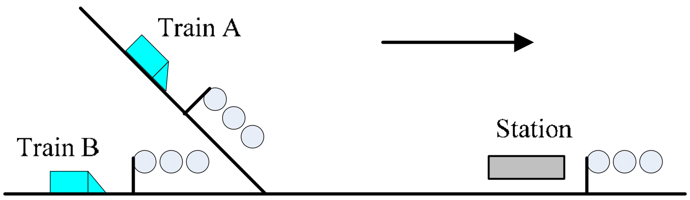

Figure A-1 Sketch map for a junction controlled by a conventional ARS

Figure A-1 shows a sketch map for a junction controlled by a conventional ARS. To explain the principle of the ARS strategy for the junction control, it is assumed that Train A and Train B are approaching the junction point and terminate at the station shown in Figure A-1. The decision of train sequence for passing the junction is made by the ARS system in terms of the current delay of trains, train class etc. The general decision making process is described as follows:

Train A and Train B are approaching the junction point from converging routes as shown in Figure A-1.

The ARS calculates the estimated delays due to the train passing sequence.

If Train A runs first, the delay to Train B is d_B;

If Train B runs first, the delay to Train A is d_A;

Given d_A, and d_B, the weighting of Train A ω_A, and the weighting of Train B ω_B, *If* $|d_A - d_B| \leq T$ (T is a configuration data, set to be 30s in this thesis)

Then run Train A and Train B in Timetabled order;

Otherwise *if* $\omega_A * d_A < \omega_B * d_B$

Then run Train B first;

Else run Train A first.

The logic shown above is the conventional ARS strategy applied in this thesis for performance comparison with the algorithm DE_JRM and FCFS.

Appendix B. Publications during PhD Research

---

[1] Chen, L., F. Schmid, M. Dasigi, B. Ning, C. Roberts, T. Tang. (2010). "Real-time train rescheduling in junction areas." Proceedings of the Institution of Mechanical Engineers Part F-Journal of Rail and Rapid Transit 224(F6): 547-557. (W. A. Agnew Meritorious/C. N. Goodall Award 2010, IMechE)

[2] Chen, L., F. Schmid, B. Ning, C. Roberts, T. Tang. (2010). “A Cooperative Strategy Framework of Train Rescheduling for Portal Junctions Leading Into Bottleneck Sections.” Computers in Railways XII. Computer System Design and Operation in Railways and Other Transit Systems: 935-944.

References

Abbass, H. A. (2002). "The self-adaptive Pareto Differential Evolution algorithm." Cec'02: Proceedings of the 2002 Congress on Evolutionary Computation, Vols 1 and 2: 831-836.

Adler, I., M. G. C. Resende, et al. (1989). "An Implementation of Karmarkar Algorithm for Linear-Programming." Mathematical Programming **44**(3): 297-335.

Brannlund, U., P. O. Lindberg, et al. (1998). "Railway timetabling using Lagrangian relaxation." Transportation Science **32**(4): 358-369.

Caimi, G., M. Fuchsberger, et al. (2011). "Periodic Railway Timetabling with Event Flexibility." Networks **57**(1): 3-18.

Caprara, A., M. Fischetti, et al. (2002). "Modeling and solving the train timetabling problem." Operations Research **50**(5): 851-861.

Caprara, A., M. Monaci, et al. (2006). "A Lagrangian heuristic algorithm for a real-world train timetabling problem." Discrete Applied Mathematics **154**(5): 738-753.

Carey, M. and A. Kwiecinski (1994). "Stochastic-Approximation to the Effects of Headways on Knock-on Delays of Trains." Transportation Research Part B-Methodological **28**(4): 251-267.

Chang, C. S. and B. S. Thia (1996). "Online rescheduling of mass rapid transit systems: Fuzzy expert system approach." Iee Proceedings-Electric Power Applications **143**(4): 307-316.

Charlwood, M., S. Turner, et al. (2004). A methodology for the assignment of safety integrity levels (SILs) to safety-related control functions implemented by safety-related electrical, electronic and programmable electronic control systems of machines. UK Health and Safety Executive Research Report 216.

Chen, L., F. Schmid, et al. (2010). "Real-time train rescheduling in junction areas." Proceedings of the Institution of Mechanical Engineers Part F-Journal of Rail and Rapid Transit **224**(F6): 547-557.

Cheng, Y. (1996). "A network-based train traffic simulation with changeable rescheduling strategies." Second International Symposium on Parallel Architectures, Algorithms, and Networks (I-Span '96), Proceedings: 214-220.

Cheng, Y. (1998). "Hybrid simulation for resolving resource conflicts in train traffic rescheduling." Computers in Industry **35**(3): 233-246.

Chiu, C. K., C. M. Chou, et al. (2002). "A Constraint-Based Interactive Train Rescheduling Tool." Kluwer Academic Publishers, Netherlands **7**(2): 167-198.

Chou, Y. H., P. F. Weston, et al. (2009). "Collaborative Rescheduling in a Distributed Railway Control System." Proceedings of 3rd International Seminar on Railway Operations Modelling and Analysis (CD-Rom): 1-17.

Corman, F., A. D'Ariano, et al. (2010). "A tabu search algorithm for rerouting trains during rail operations." Transportation Research Part B-Methodological **44**(1): 175-192.

D'Ariano, A. (2010). "Improving real-time train dispatching performance: optimization models and algorithms for re-timing, re-ordering and local re-routing." 4or-a Quarterly Journal of Operations Research **8**(4): 429-432.

D'Ariano, A. and T. Albrecht (2006). "Running time re-optimization during real-time timetable perturbations." Computers in Railways X: Computer System Design and Operation in the Railway and Other Transit Systems **88**: 531-540, 985.

D'Ariano, A., D. Pacclarelli, et al. (2007). "A branch and bound algorithm for scheduling trains in a railway network." European Journal of Operational Research **183**(2): 643-657.

Dantzig, G. B. (1948). "Programming in a Linear Structure." Bulletin of the American Mathematical Society **54**(11): 1074-1074.

Das, S., A. Abraham, et al. (2008). "Automatic clustering using an improved differential evolution algorithm." Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans **38**(1): 218-237.

Davis, P. K. and J. Bigelow (1998). "Introduction to multi-resolution modeling (MRM) with an example involving precision fires." Enabling Technology for Simulation Science Ii **3369**: 14-27.

Dorigo, M. (1992). Optimization, Learning and Natural Algorithms Italie, Politecnico di Milano.

EU-Project-FP7. (2011). "Optimal Networks for Train Integration Management across Europe." from http://www.ontime-project.eu/home.aspx.

Fay, A. (2000). "A fuzzy knowledge-based system for railway traffic control." Engineering Applications of Artificial Intelligence **13**(6): 719-729.

Ferreira, L. and A. Higgins (1996). "Modeling reliability of train arrival times." Journal of Transportation Engineering-Asce **122**(6): 414-420.

Fischetti, M., D. Salvagnin, et al. (2009). "Fast Approaches to Improve the Robustness of a Railway Timetable." Transportation Science **43**(3): 321-335.

Garey, M. R. and D. S. Johnson (1979). Computers and Intractability: A Guide to Theory of NP-Completeness. New York, Freeman.

Glover, F. (1977). "Heuristics for Integer programming Using Surrogate Constraints." Decision Sciences **8**(1): 156-166.

Glover, F. (1986). "Future Paths for Integer Programming and Links to Artificial-Intelligence." Computers & Operations Research **13**(5): 533-549.

Goverde, R. M. P. (2005). Punctuality of Railway Operations and Timetable Stability Analysis, Delft University of Technology. **PhD**.

Goverde, R. M. P. (2007). "Railway timetable stability analysis using max-plus system theory." Transportation Research Part B-Methodological **41**(2): 179-201.

Hampel, F. R. (1971). "General Qualitative Definition of Robustness." Annals of Mathematical Statistics **42**(6): 1887-&.

Hansen, I. A. (2004). "Optimisation of railway capacity use by stochastic modelling." Traffic and Transportation Studies, Proceedings: 1-10.

Hansen, I. A. (2006). "State-of-the-art of railway operations research." Computers in Railways X: Computer System Design and Operation in the Railway and Other Transit Systems **88**: 565-577.

Hansen, I. A. (2010). "Railway Network Timetabling and Dynamic Traffic Management." International Journal of Civil Engineering **8**(1): 19-32.

Higgins, A. and E. Kozan (1998). Modelling Train Delays in Urban Networks. Transportation Science. **32:** 346-357.

Hirai, C., N. Tomii, et al. (2007). "Train rescheduling algorithm using pattern description language and integrated algorithm framework." Procedings of the 11th Iasted International Conference on Artificial Intelligence and Soft Computing: 250-255, 310.

Ho, T. K., J. P. Norton, et al. (1997). "Optimal traffic control at railway junctions." Iee Proceedings-Electric Power Applications **144**(2): 140-148.

Hooker, C. A. (1995). "Adaptation in Natural and Artificial Systems - Holland,Jh." Philosophical Psychology **8**(3): 287-299.

Huisman, T. and R. J. Boucherie (2001). "Running times on railway sections with heterogeneous train traffic." Transportation Research Part B-Methodological **35**(3): 271-292.

Hyoudou, M., T. Seto, et al. (1997). "An expert system for train operation adjustment of Shinkansen." Transportation Systems 1997, Vols 1-3: 1145-1150.

Jacobs, J. (2004). "Reducing delays by means of computer-aided 'on-the-spot' rescheduling." Computers in Railway Six **15**: 603-612, 1015.

Karmarkar, N. (1984). "A New Polynomial-Time Algorithm for Linear-Programming." Combinatorica **4**(4): 373-395.

Karush, W. (1939). Minima of Functions of Several Variables with Inequalities as Side Constraints. Dept. of Mathematics. Chicago, Illinois, Univ. of Chicago. **M.Sc. Dissertation**.

Kennedy, J. and R. Eberhart (1995). "Particle swarm optimization." 1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6: 1942-1948.

Kirkpatrick, S., C. D. Gelatt, et al. (1983). "Optimization by Simulated Annealing." Science **220**(4598): 671-680.

Konno, H., N. Kawadai, et al. (2003). "Cutting plane algorithms for nonlinear semi-definite programming problems with applications." Journal of Global Optimization **25**(2): 141-155.

Krasemann, J. T. (2010). "Greedy algorithm for railway traffic re-scheduling during disturbances: a Swedish case." Iet Intelligent Transport Systems **4**(4): 375-386.

Kroon, L., G. Maroti, et al. (2008). "Stochastic improvement of cyclic railway timetables." Transportation Research Part B-Methodological **42**(6): 553-570.

Kuhn, H. W. T., A. W. (1951). Nonlinear programming. Proceedings of 2nd Berkeley Symposium. Berkeley, University of California Press**:** 481-492.

Kuhn, M. (1998). "Automatic route setting integrated in a modern traffic management system." International Conference on Developments in Mass Transit Systems: 140-145.

Land, A. H. and A. G. Doig (1960). "An Automatic Method of Solving Discrete Programming-Problems." Econometrica **28**(3): 497-520.

Liebchen, C. (2005). "A cut-based heuristic to produce almost feasible periodic railway timetables." Experimental and Efficient Algorithms, Proceedings **3503**: 354-366.

Liebchen, C. and L. Peeters (2009). "Integral cycle bases for cyclic timetabling." Discrete Optimization **6**(1): 98-109.

Liebchen, C., M. Schachtebeck, et al. (2010). "Computing delay resistant railway timetables." Computers & Operations Research **37**(5): 857-868.

Liu, J. and J. Lampinen (2005). "A fuzzy adaptive differential evolution algorithm." Soft Computing **9**(6): 448-462.

Luethi, M. (2008). "Evaluation of energy saving strategies in heavily used rail networks by implementing an integrated real-time rescheduling system." Computers in Railways Xi **103**: 349-358.

Luethi, M., G. Medeossi, et al. (2009). "Structure and Simulation Evaluation of an Integrated Real-Time Rescheduling System for Railway Networks." Networks & Spatial Economics **9**(1): 103-121.

Mannino, C. and A. Mascis (2009). "Optimal Real-Time Traffic Control in Metro Stations." Operations Research **57**(4): 1026-1039.

Martin, D. H. (1985). "The Essence of Invexity." Journal of Optimization Theory and Applications **47**(1): 65-76.

Mayer, D. G., B. P. Kinghorn, et al. (2005). "Differential evolution - an easy and efficient evolutionary algorithm for model optimisation." Agricultural Systems **83**(3): 315-328.

Mazzarello, M. and E. Ottaviani (2007). "A traffic management system for real-time traffic optimisation in railways." Transportation Research Part B-Methodological **41**(2): 246-274.

Mees, A. I. (1991). "Railway Scheduling by Network Optimization." Mathematical and Computer Modelling **15**(1): 33-42.

Mehta, F., C. Rößiger, et al. (2010). Latent energy savings due to the innovative use of advisory speeds to avoid occupation conflicts. Computers in railways VII. Beijing, China**:** 99-108.

Middelkoop, D. and M. Bouwman (2000). Train network simulator for support of network wide planning of infrastructure and timetables. Computers in Railways VII. Beijing, China. **7:** 267-276.

Middelkoop, D. and M. Bouwman (2001). "Simone: Large scale train network simulations." Wsc'01: Proceedings of the 2001 Winter Simulation Conference, Vols 1 and 2: 1042-1047.

Milchev, A. (2003). "Introduction to Monte Carlo methods." Computer Simulations of Surfaces and Interfaces **114**: 39-55.

Minoux, M. (1986). Mathematical Programming: Theory and Algorithms. Sussex, UK.

Nachtigall, K. and S. Voget (1997). "Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks." European Journal of Operational Research **103**(3): 610-627.

Nash, A. and D. Huerlimann (2004). "Railroad simulation using OpenTrack." Computers in Railway Six **15**: 45-54.

Neumaier, A. (1996). "Second-order sufficient optimality conditions for local and global nonlinear programming." Journal of Global Optimization **9**(2): 141-151.

Nobuyuki, O., M. Akatsu, et al. (1996). "Rescheduling method based on distributed cooperative problem solving model." Etfa '96 - 1996 Ieee Conference on Emerging Technologies and Factory Automation, Proceedings, Vols 1 and 2: 35-41.

Osman, I. H. and G. Laporte (1996). "Metaheuristics: A bibliography." Annals of Operations Research **63**: 513-623.

Pachl, J. (2002). Railway operation and control.

Powell, M. J. D. (1964). "Efficient Method for Finding Minimum of Function of Several-Variables without Calculating Derivatives." Computer Journal **7**(2): 155-&.

Price, K. V., R. M. Storn, et al. (2005). Differential Evolution - A Practical Approach to Global Optimization. Berlin, Springer-Verlag.

Qin, A. K. and P. N. Suganthan (2005). "Self-adaptive differential evolution algorithm for numerical optimization." 2005 Ieee Congress on Evolutionary Computation, Vols 1-3, Proceedings: 1785-1791.

Radtke, A. and D. Hauptmann (2004). "Automated planning of timetables in large railway networks using a microscopic data basis and railway simulation techniques." Computers in Railway Six **15**: 615-625.

Raeside, D. E. (1974). "Introduction to Monte-Carlo Methods." American Journal of Physics **42**(1): 20-26.

Raychaudhuri, S. (2008). "Introduction to Monte Carlo Simulation." 2008 Winter Simulation Conference, Vols 1-5: 91-100.

Rechenberg, I. (1971). Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution

Sayah, S. and K. Zehar (2008). "Modified differential evolution algorithm for optimal power flow with non-smooth cost functions." Energy Conversion and Management **49**(11): 3036-3042.

Schenk, H. (1998). "An elementary introduction to direct methods." Direct Methods for Solving Macromolecular Structures **507**: 19-26.

Schwanhauber, W. (1974). Die Bemessung der Pufferzeiten im Fahrplangefuge der Eisenbahn, RWTH Aachen. **Phd**.

Serafini, P. and W. Ukovich (1989). "A Mathematical-Model for the Fixed-Time Traffic Control Problem." European Journal of Operational Research **42**(2): 152-165.

Storn, R. and K. Price (1997). "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces." Journal of Global Optimization **11**(4): 341-359.

Tai, X. C. and M. Espedal (1998). "Rate of convergence of some space decomposition methods for linear and nonlinear problems." Siam Journal on Numerical Analysis **35**(4): 1558-1570.

Takagi, R., P. F. Weston, et al. (2006). "Optimal train control at a junction in the main line rail network using a new object-oriented signalling system model." Computers in Railways X: Computer System Design and Operation in the Railway and Other Transit Systems **88**: 479-488.

Takahama, T. and S. Sakai (2009). "Fast and Stable Constrained Optimization by the Epsilon Constrained Differential Evolution." Pacific Journal of Optimization **5**(2): 261-282.

Teo, J. (2006). "Exploring dynamic self-adaptive populations in differential evolution." Soft Computing **10**(8): 673-686.

Thameslink-Programme. from http://www.thameslinkprogramme.co.uk/cms/pages/home.

Tomii, N., Y. Tashiro, et al. (2005). "Train operation rescheduling algorithm based on passenger satisfaction." Quarterly Report of RTRI **46**(3): 167-172.

Tornquist, 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**(3): 249-266.

Tornquist, J. and J. A. Persson (2007). "N-tracked railway traffic re-scheduling during disturbances." Transportation Research Part B-Methodological **41**(3): 342-362.

Transport-for-London. "Journey Time Capability Calculation and Definitions ", from http://www.tfl.gov.uk/tfl/corporate/modesoftransport/tube/pppcontracts/6_2_1057_1.asp.

Vansteenwegen, P. and D. van Oudheusden (2006). "Developing railway timetables which guarantee a better service." European Journal of Operational Research **173**(1): 337-350.

Vromans, M., R. Dekker, et al. (2004). "Simulation and railway timetabling norms." Computers in Railway Six **15**: 685-694.

Watson, R. (2005). Using stochastic simulation to predict timetable performance - status and developments in the UK. 1st Int. Seminar on Railway Operations Modelling and Analysis. Delft.

Wen, S. H., Q. B. Lu, et al. (2008). "A modified differential evolution for constrained optimization." Proceedings of the International Conference Information Computing and Automation, Vols 1-3: 252-255.

Wong, R. C. W., T. W. Y. Yuen, et al. (2008). "Optimizing timetable synchronization for rail mass transit." Transportation Science **42**(1): 57-69.

Yang, G. Y., Z. Y. Dong, et al. (2008). "A modified differential evolution algorithm with fitness sharing for power system planning." Ieee Transactions on Power Systems **23**(2): 514-522.

Yuan, J., R. M. P. Goverde, et al. (2002). "Propagation of train delays in stations." Computers in Railways Viii **13**: 975-984.

Yuan, J., R. M. P. Goverde, et al. (2006). "Evaluating stochastic train process time distribution models on the basis of empirical detection data." Computers in Railways X: Computer System Design and Operation in the Railway and Other Transit Systems **88**: 631-640.

Yuan, J. X. and I. A. Hansen (2007). "Optimizing capacity utilization of stations by estimating knock-on train delays." Transportation Research Part B-Methodological **41**(2): 202-217.

Zhang, J. Q. and A. C. Sanderson (2007). "JADE: Self-adaptive Differential Evolution with fast and reliable convergence performance." 2007 Ieee Congress on Evolutionary Computation, Vols 1-10, Proceedings: 2251-2258.